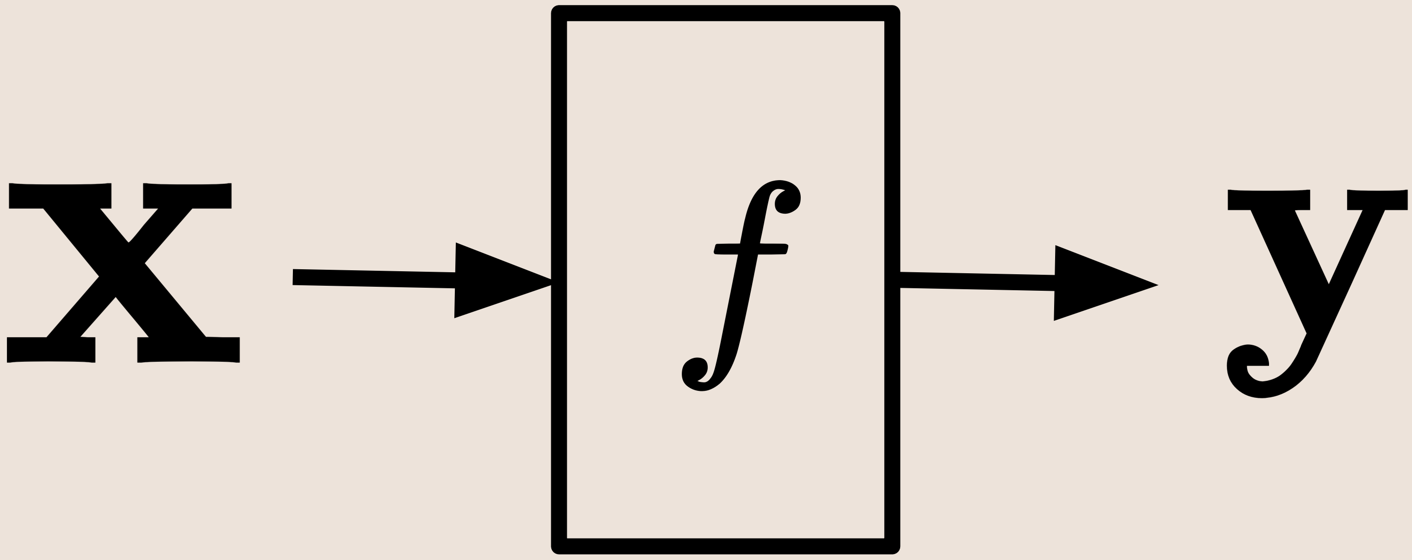
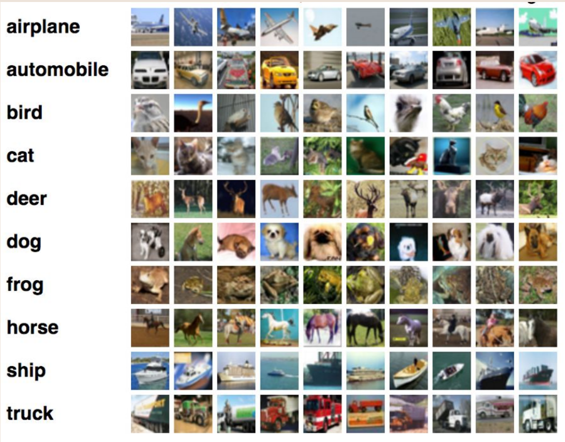
Towards better uncertainty in Bayesian Neural Networks

Wednesday 19th, December 2018

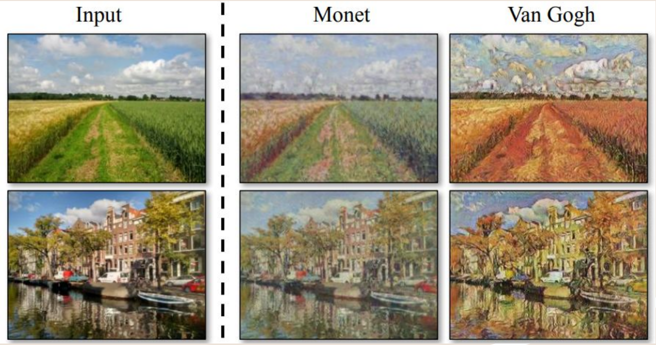
Melanie F. Pradier


x
f
y

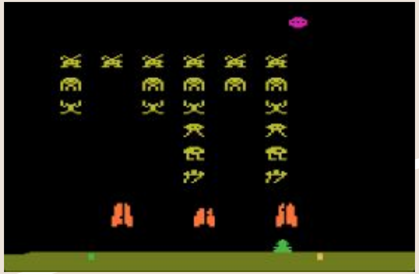


[Krizhevsky et.al, 2017]

[Ulatus post, 2016]

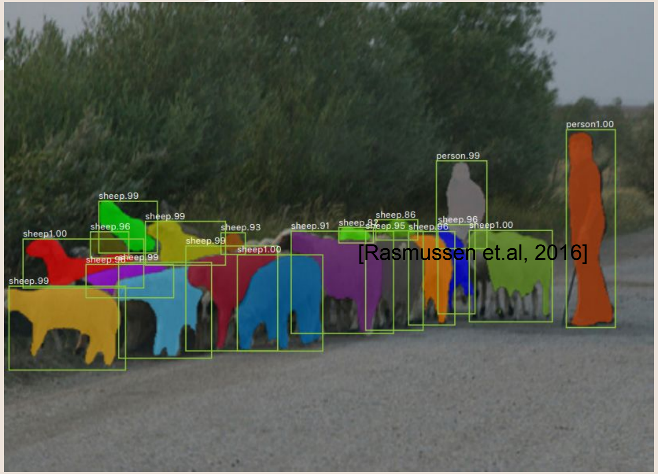


[Zhu et.al, 2018]



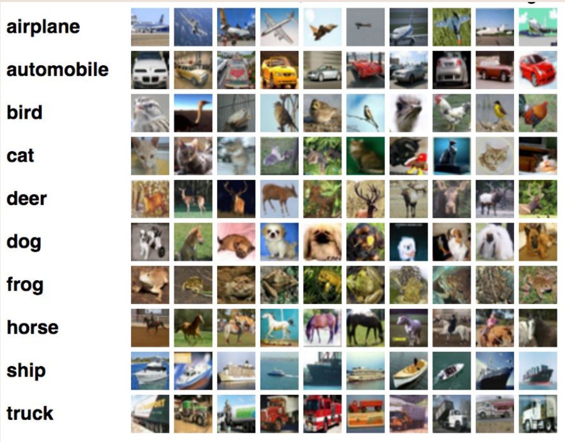
[Minh et.al, 2015]

[He et.al, 2018]



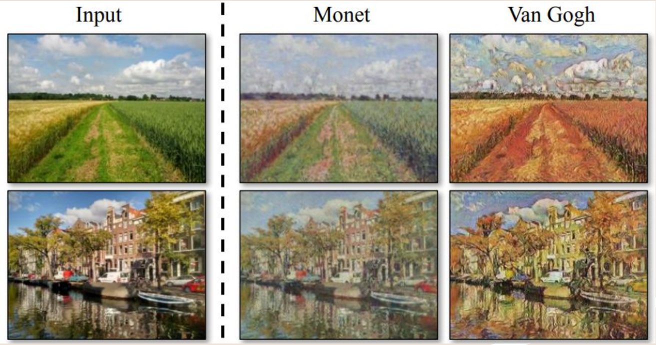
[Rasmussen et.al, 2016]

[Silver et.al, 2017]

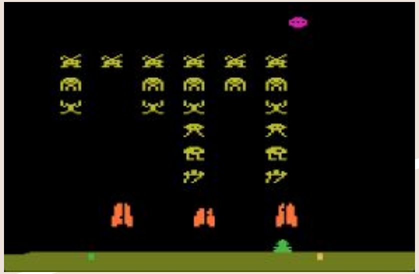


[Krizhevsky et.al, 2017]

[Ulatus post, 2016]



[Zhu et.al, 2018]

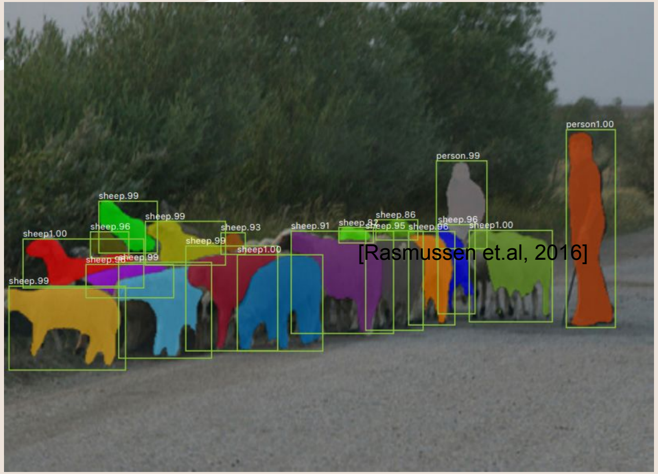


[Minh et.al, 2015]

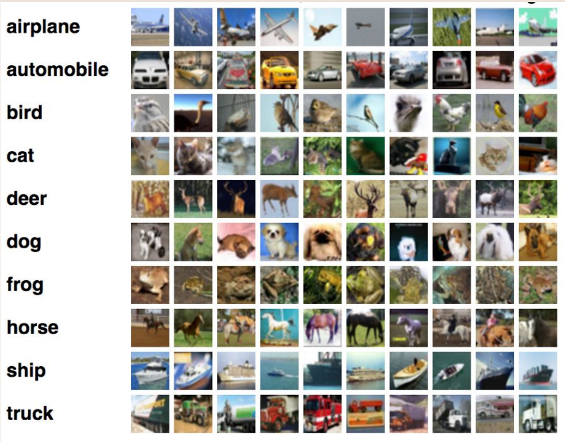
[Silver et.al, 2017]

- Highly-predictive
- Scalable

[He et.al, 2018]

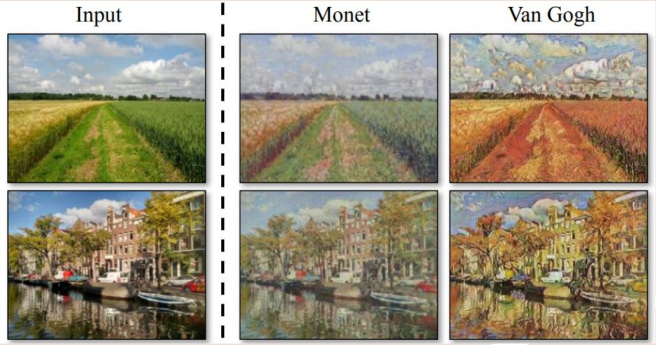


[Rasmussen et.al, 2016]

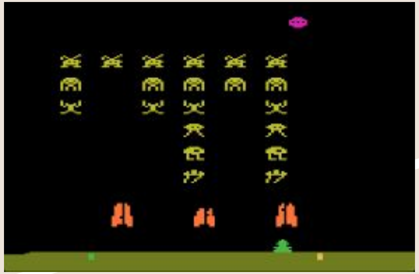


[Krizhevsky et.al, 2017]

[Ulatus post, 2016]



[Zhu et.al, 2018]



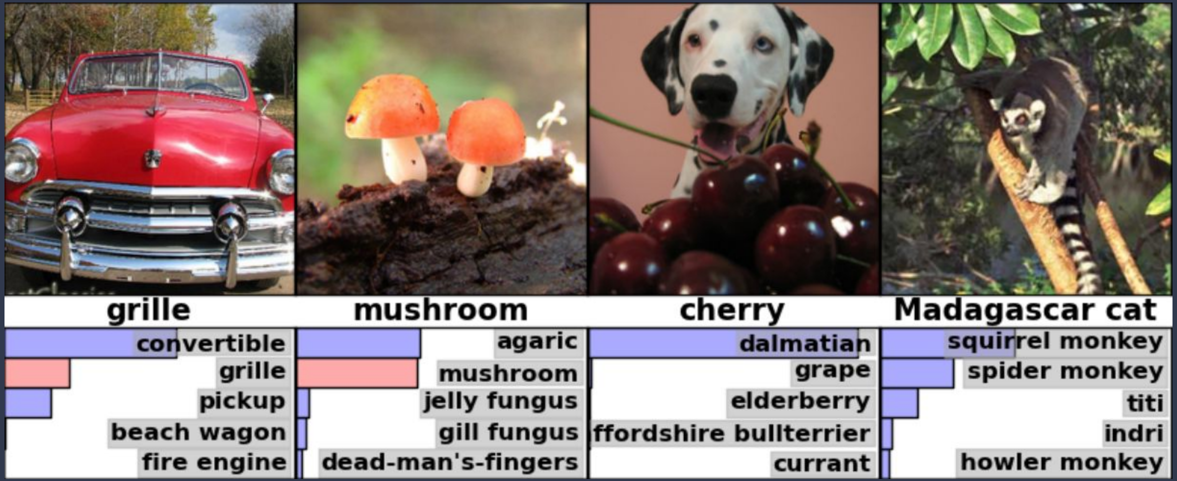
[Minh et.al, 2015]

[He et.al, 2018]

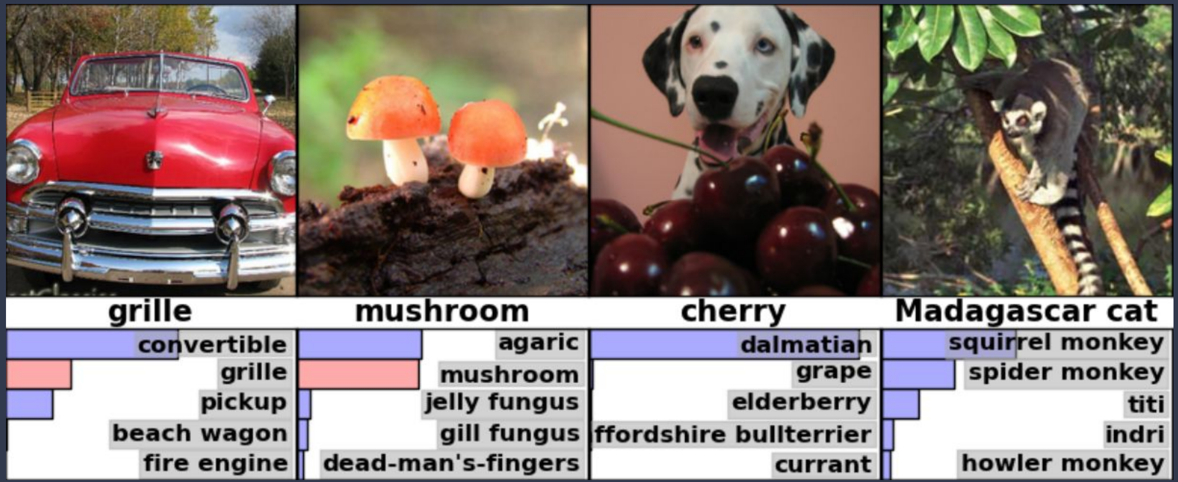
[Rasmussen et.al, 2016]

[Silver et.al, 2017]

- Highly-predictive
- Scalable

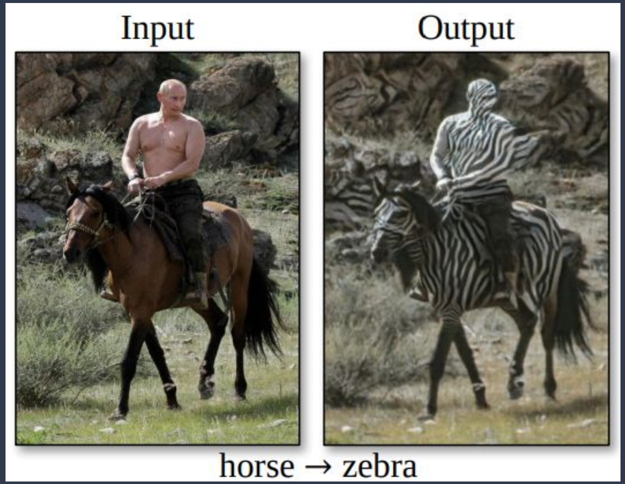


[Krizhevsky et.al, 2017]

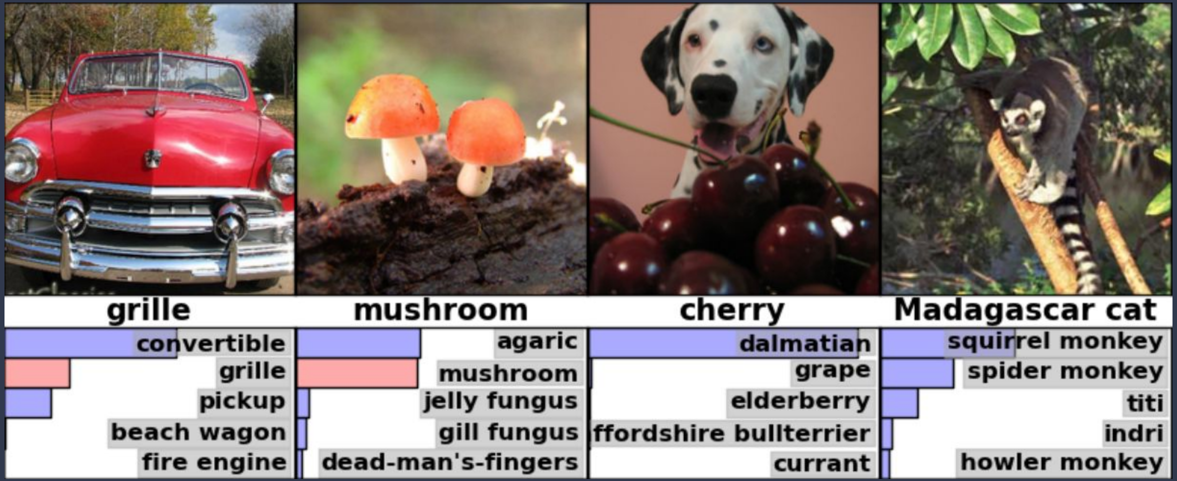


[Krizhevsky et.al, 2017]

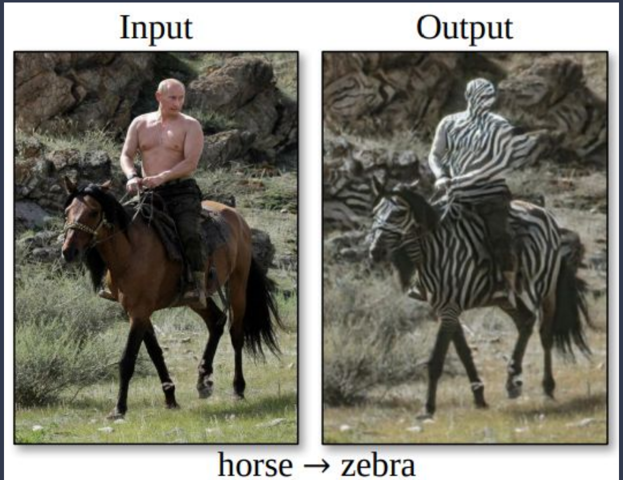
[Zhu et.al, 2018]



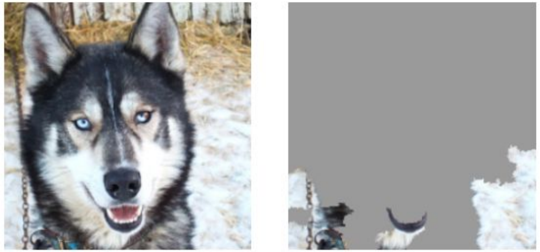
[Krizhevsky et.al, 2017]



[Zhu et.al, 2018]



[Ribero et.al, 2016]



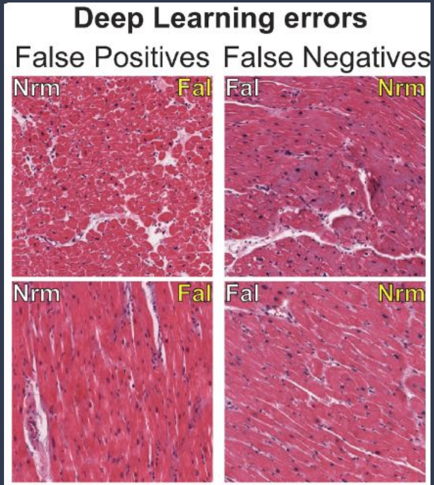
(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

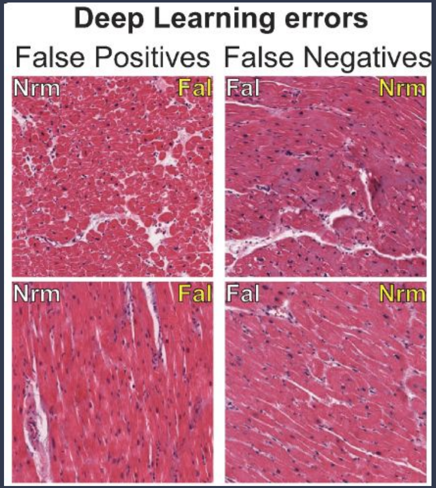
[Eykholt et.al, 2018]

[Nirschi et.al, 2018]

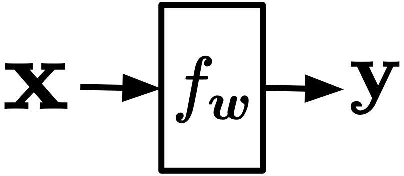


[Eykholt et.al, 2018]

[Nirschi et.al, 2018]



Our Goal:



$$\mathbf{y} = f_w(\mathbf{x}) + \boldsymbol{\epsilon}$$

Quantify Uncertainty

With such uncertainty, we can:

- Alert humans in unclear situations
- Diagnose ML systems (when and how does it fail)
- Get better predictive accuracy

Overview

Two sources of uncertainty

epistemic

$p(\mathbf{w}|\mathcal{D})$

aleatoric

$p(\mathbf{y}|\mathbf{x}, \boldsymbol{w})$

$$\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \boldsymbol{\epsilon}$$

[Depeweg et.al, 2017]

Overview

Two sources of uncertainty

epistemic

$p(\mathbf{w}|\mathcal{D})$

aleatoric

$p(\mathbf{y}|\mathbf{x}, \boldsymbol{w})$

$$\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \boldsymbol{\epsilon}$$

[Depeweg et.al, 2017]

Overview

In this talk...

Two sources of uncertainty

epistemic

$p(\mathbf{w}|\mathcal{D})$

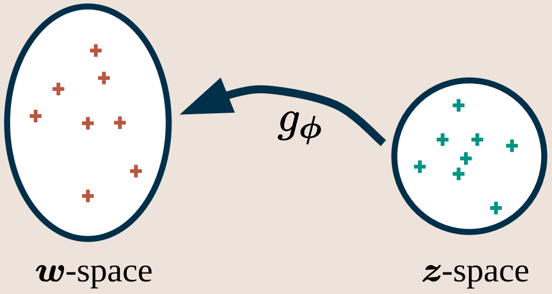
aleatoric

$p(\mathbf{y}|\mathbf{x}, \boldsymbol{w})$

$$\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \boldsymbol{\epsilon}$$

[Depeweg et.al, 2017]

- Approximate $f_{\boldsymbol{w}}$ with a Bayesian Neural Network



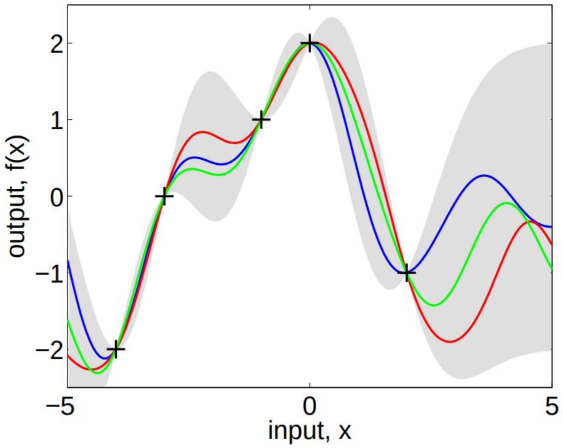
- Modeling + inference contributions

How to estimate function uncertainty?

How to estimate function uncertainty?

Gaussian Process (GP)

[Rasmussen et.al, 2016]

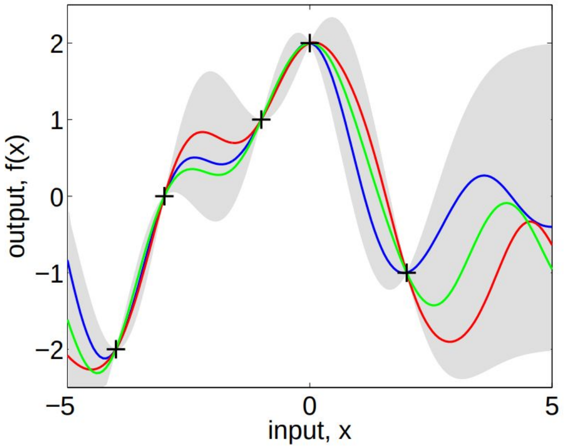


$$f(x) \sim \mathrm{GP}\left(m(x), k(x, x')\right)$$

How to estimate function uncertainty?

Gaussian Process (GP)

[Rasmussen et.al, 2016]



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

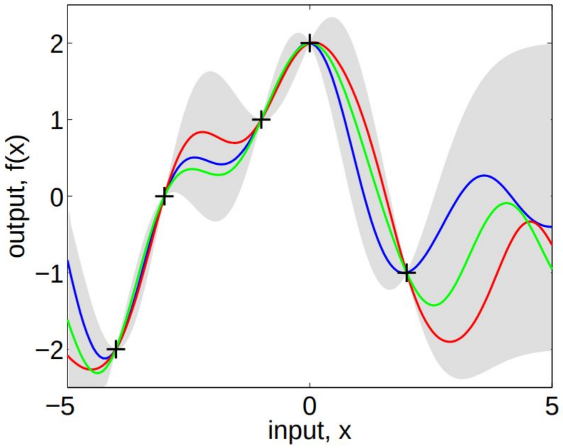
Drawbacks of GPs

- Scalability
- Kernel learning is not trivial

How to estimate function uncertainty?

Gaussian Process (GP)

[Rasmussen et.al, 2016]



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

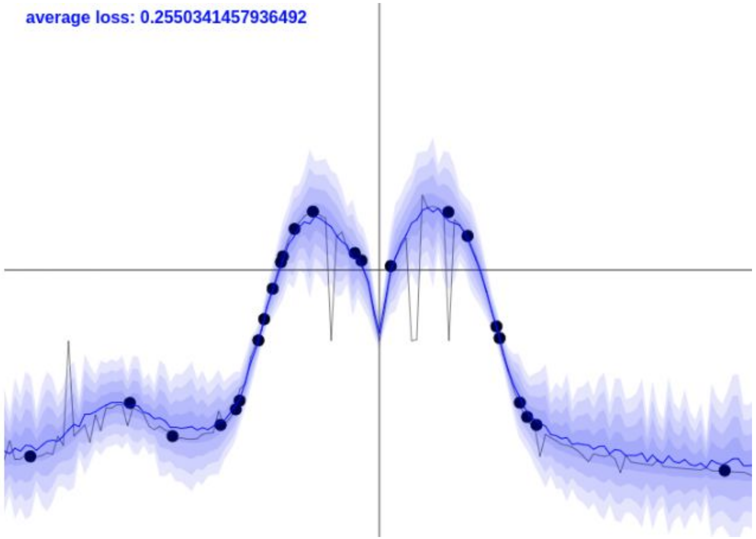
Drawbacks of GPs

- Scalability
- Kernel learning is not trivial

Alternative: Neural Networks with uncertainty

- Ensemble of Neural Networks
  [Lakshminarayanan et al., 2017; Pearce et.al, 2018]
- Bayesian Neural Networks
  [Buntine et al., 1991; MacKay, 1992; Neal, 1993]

Bayesian Neural Network (BNN)



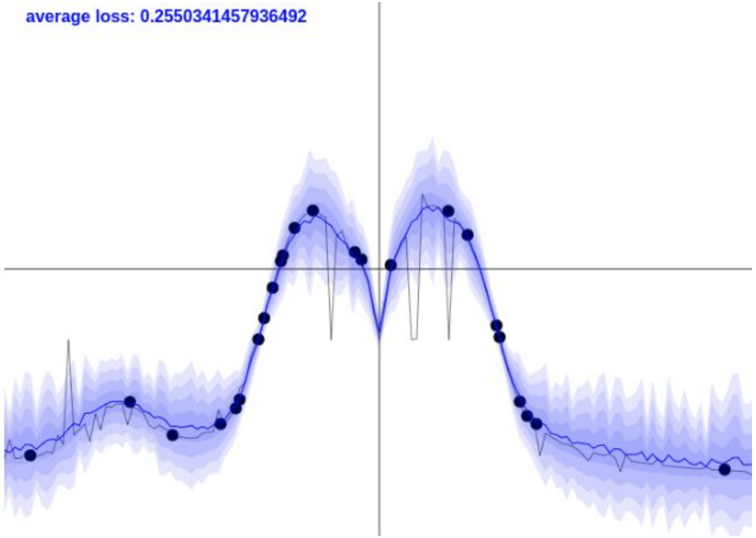
[What my deep model does not know, post of Yarin Gal, 2015]

$$\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$$

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

$$\boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

Bayesian Neural Network (BNN)



[What my deep model does not know, post of Yarin Gal, 2015]

$$\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

$$\boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

Quantities of interest:

- Posterior of the weights $p(\boldsymbol{w}|\mathcal{D})$
- Predictive distribution

$$p(\mathbf{y}^\star|\mathbf{x}^\star, \mathcal{D}) = \int p(\mathbf{y}^\star|\mathbf{x}^\star, \boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}$$

$$p(\boldsymbol{w}|\mathcal{D})$$

is intractable!

$$p(\boldsymbol{w}|\mathcal{D})$$

is intractable!

Inference options:

- Markov Chain Monte Carlo
  Hamiltonian Monte Carlo [Neal, 1993]
- Variational Inference
  [Graves, 1993] [Blundell et.al, 2015]

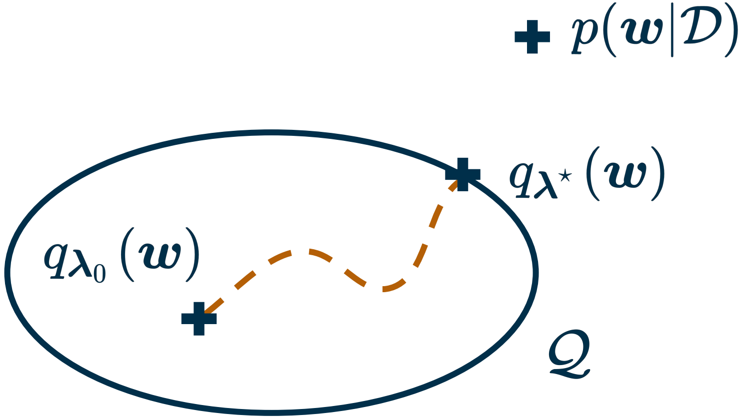
Variational Inference for BNNs

[Blundell et.al, 2015]

Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmin}}\; D_{\text{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$



Variational Inference for BNNs

[Blundell et.al, 2015]

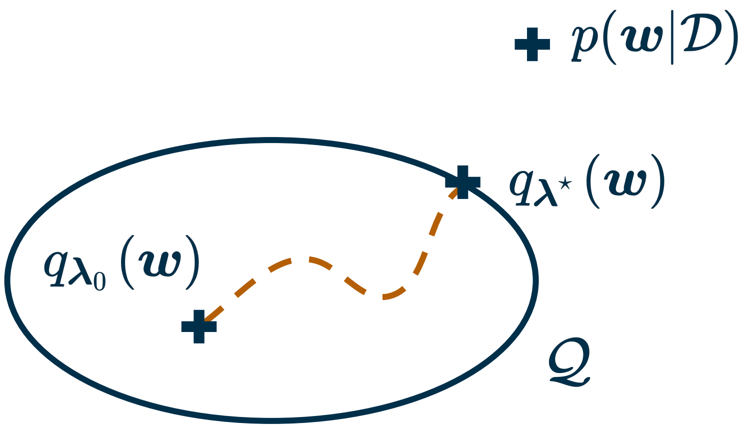
Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmin}}\, D_{\text{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\right)$$

$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmax}}\, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\left[\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})\right] - D_{\text{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\right)$$



Variational Inference for BNNs

[Blundell et.al, 2015]

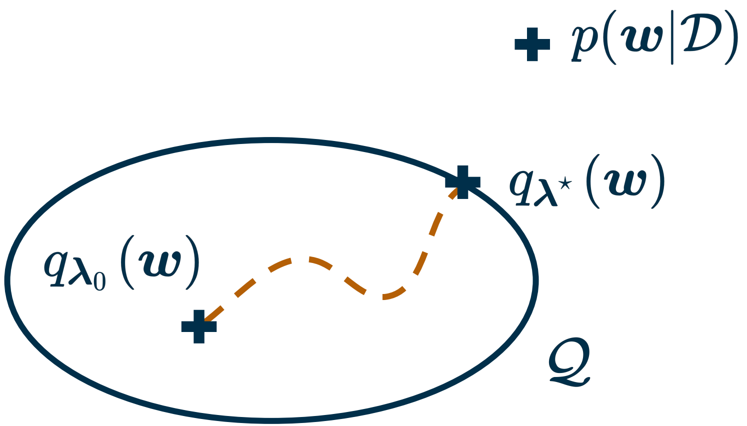
Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmin}}\ D_{\text{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$

$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmax}}\ \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p\big(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}\big)\Big] - D_{\text{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

Variational Inference for BNNs

[Blundell et.al, 2015]

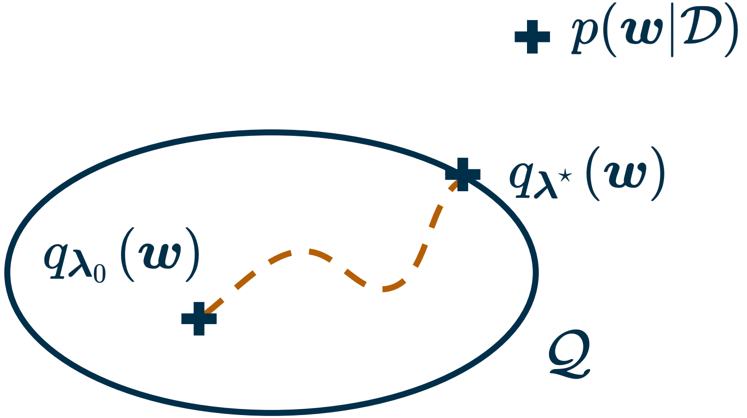
Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmin}}\; D_{\text{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$

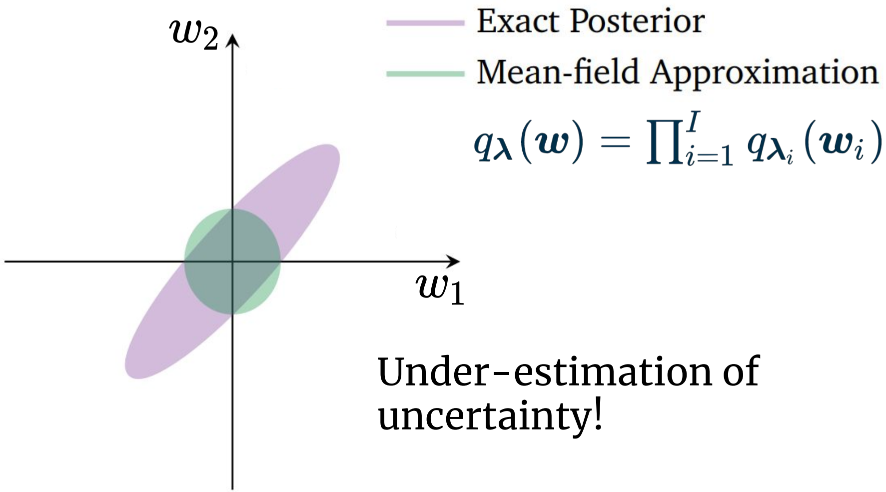
$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\text{argmax}}\; \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p\big(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}\big)\Big] - D_{\text{KL}}\big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\big)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

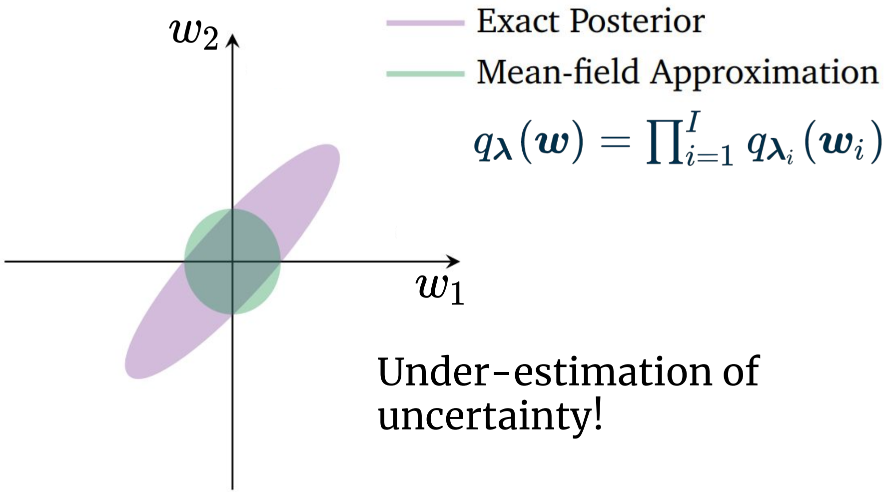
Is mean-field VI good enough?



$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) = \prod_{i=1}^{I} q_{\boldsymbol{\lambda}_i}(\boldsymbol{w}_i)$$

Under-estimation of uncertainty!

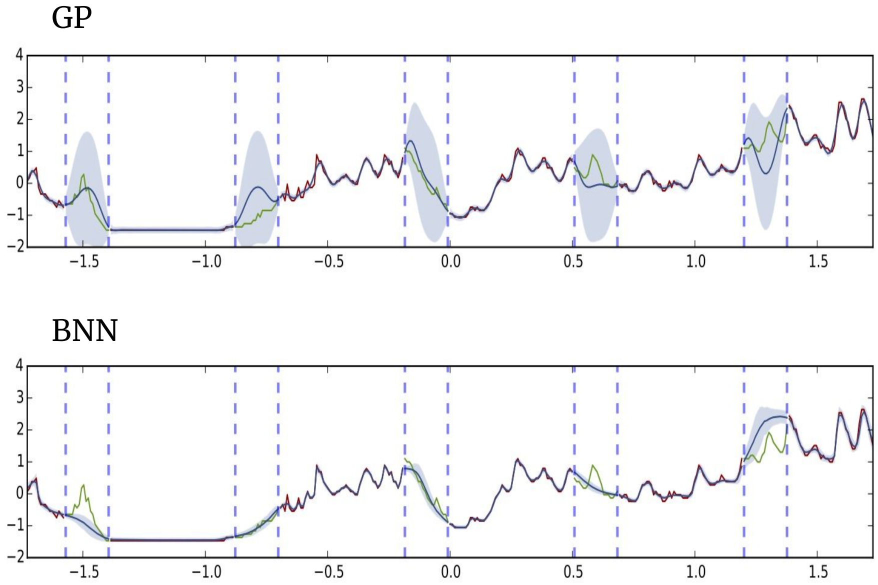
Is mean-field VI good enough?



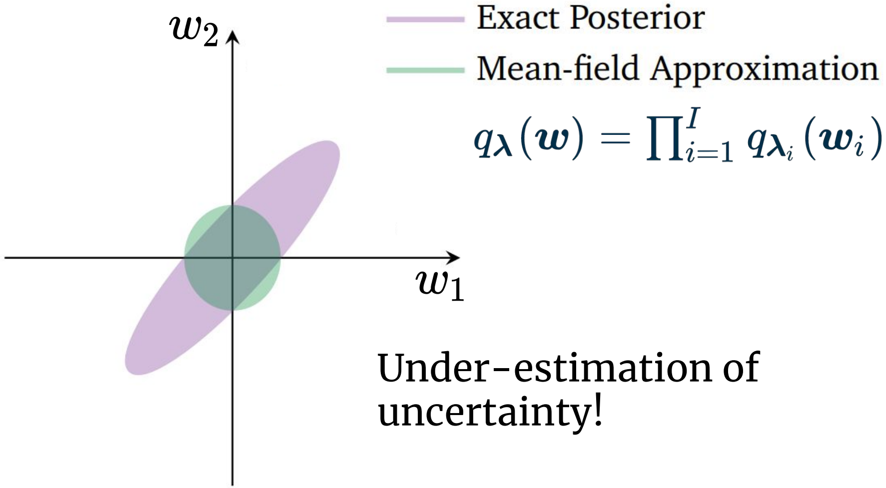
$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) = \prod_{i=1}^{I} q_{\boldsymbol{\lambda}_i}(\boldsymbol{w}_i)$

Under-estimation of uncertainty!

Example on solar irradiance dataset [Gal et.al, 2015]



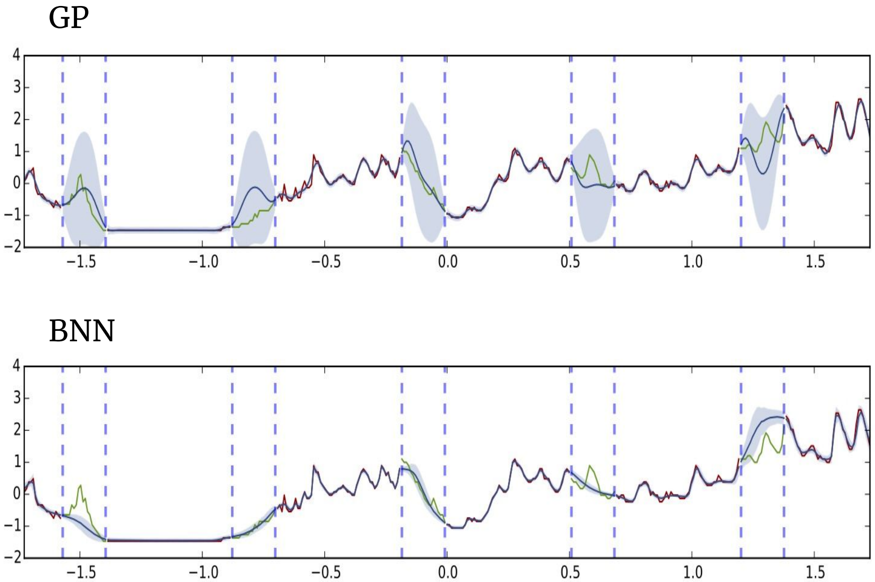
Is mean-field VI good enough?



$$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) = \prod_{i=1}^{I} q_{\boldsymbol{\lambda}_i}(\boldsymbol{w}_i)$$

Under-estimation of uncertainty!

Example on solar irradiance dataset [Gal et.al, 2015]



- **Better priors**, e.g., multivariate Gaussians [Louizos et al, 2016]
- More **flexible variational** approx. in weight Space [Louizos et.al, 2017]

Standard BNN

Modeling

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \quad \boldsymbol{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}),$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$

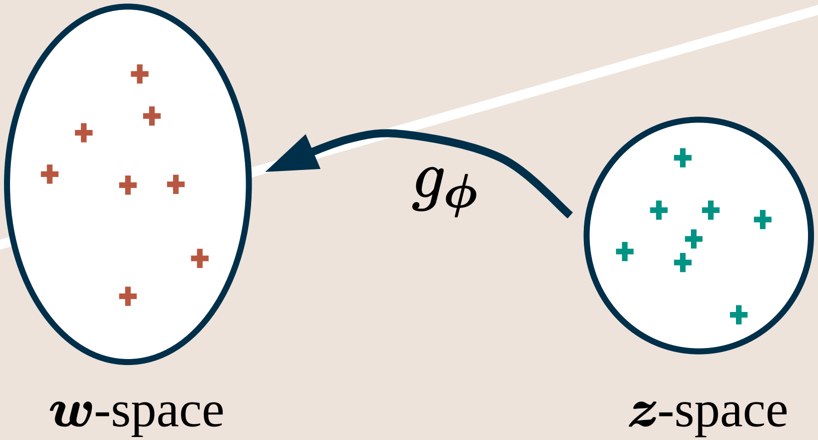
Weight redundancy
[Denil et.al, 2013]

Latent-Projection BNN

Modeling

$$\boldsymbol{y} = f_{\boldsymbol{w}}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \quad \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z}), \quad \boldsymbol{\phi} \sim p(\boldsymbol{\phi}),$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$$



$$D_w \gg D_z$$

Weight redundancy
[Denil et.al, 2013]

How about inference?

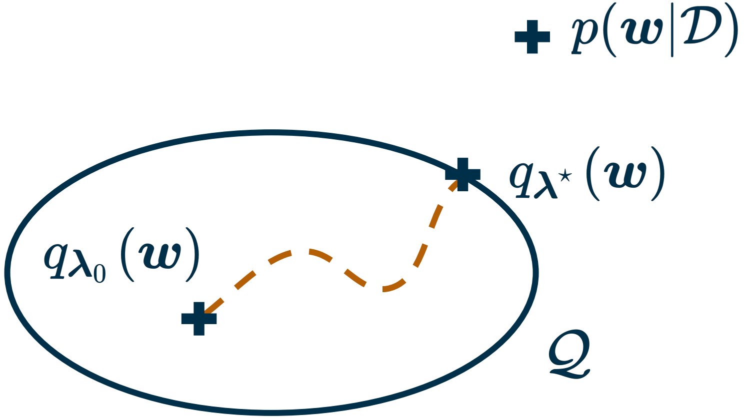
Objective: approximate $p(\boldsymbol{w}|\mathcal{D})$

$q_{\boldsymbol{\lambda}}(\boldsymbol{w}) \in \mathcal{Q}$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}}\, D_{\mathrm{KL}}\Big(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})\Big)$$

$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p\big(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}\big)\Big] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{w})||p(\boldsymbol{w})\right)$$



How about inference?

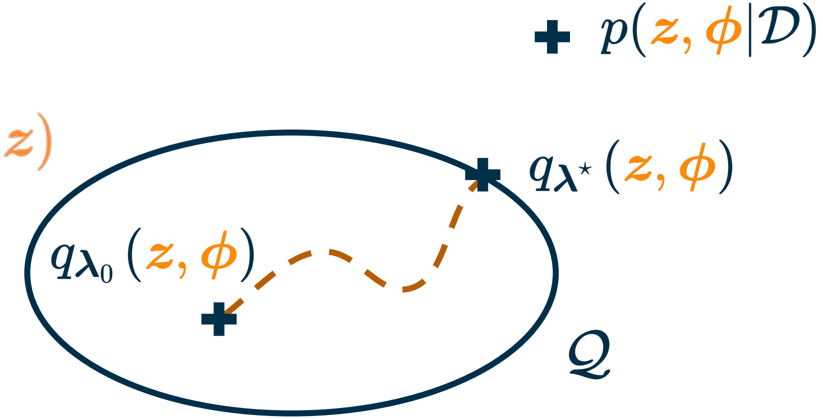
Objective: approximate $p(\boldsymbol{z}, \boldsymbol{\phi} | \mathcal{D})$

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

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}} \, D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) || p(\boldsymbol{z}, \boldsymbol{\phi} | \mathcal{D})\right)$$

$$\Updownarrow$$

$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmax}} \, \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\left[\log p\left(\boldsymbol{y} | \boldsymbol{x}, g_{\boldsymbol{\phi}}(\boldsymbol{z})\right)\right] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) || p(\boldsymbol{z})\right) - D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}) || p(\boldsymbol{\phi})\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

How about inference?

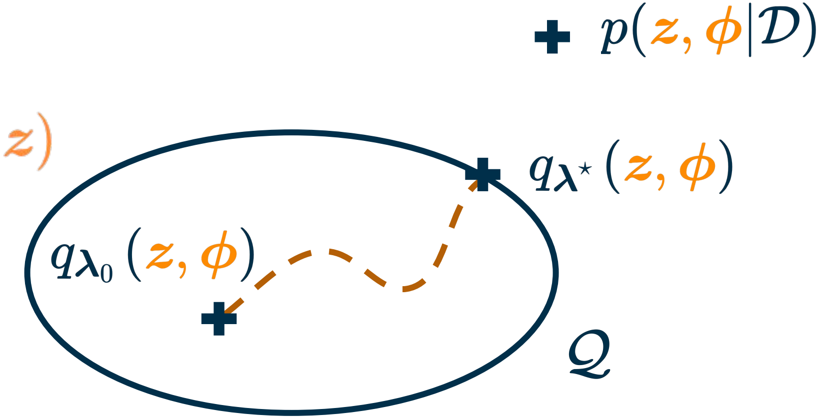
Objective: approximate $p(z, \phi | \mathcal{D})$

$$z \sim q_{\lambda_z}(z), \quad \phi \sim q_{\lambda_\phi}(\phi), \quad w = g_\phi(z)$$

$$\underset{\lambda^\star}{\operatorname{argmin}}\ D_{\mathrm{KL}}\left(q_{\lambda}(z, \phi) || p(z, \phi | \mathcal{D})\right)$$

$$\Updownarrow$$

$$\underset{\lambda^\star}{\operatorname{argmax}}\ \mathcal{L}(\lambda) = \mathbb{E}_q\left[\log p(y | x, g_\phi(z))\right] - D_{\mathrm{KL}}\left(q_{\lambda_z}(z) || p(z)\right) - D_{\mathrm{KL}}\left(q_{\lambda_\phi}(\phi) || p(\phi)\right)$$



Black-box VI [Ranganath et.al, 2013] + reparametrization trick [Kingma et.al, 2014; Rezende et.al, 2015]

$$\underset{\boldsymbol{\lambda}^{\star}}{\operatorname{argmin}}\; D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z},\boldsymbol{\phi})\,||\,p(\boldsymbol{z},\boldsymbol{\phi}|\mathcal{D})\right)$$

jointly does not work!

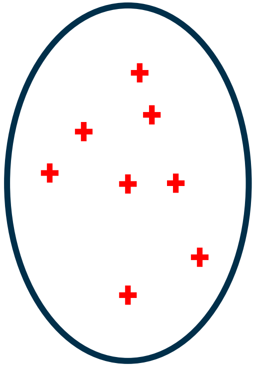
$$\underset{\boldsymbol{\lambda}^\star}{\operatorname{argmin}}\; D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z},\boldsymbol{\phi})\,||\,p(\boldsymbol{z},\boldsymbol{\phi}|\mathcal{D})\right)$$

jointly does not work!

Our solution: find smart initialization

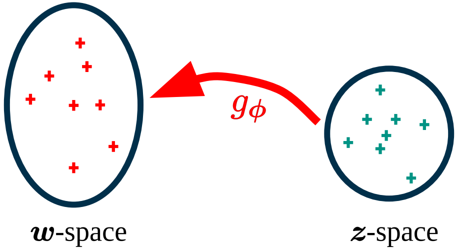
Solution: 3-stage Inference Framework

1. Characterize weight space



Train ensemble of neural networks

2. Find point estimate g_ϕ



Train an autoencoder

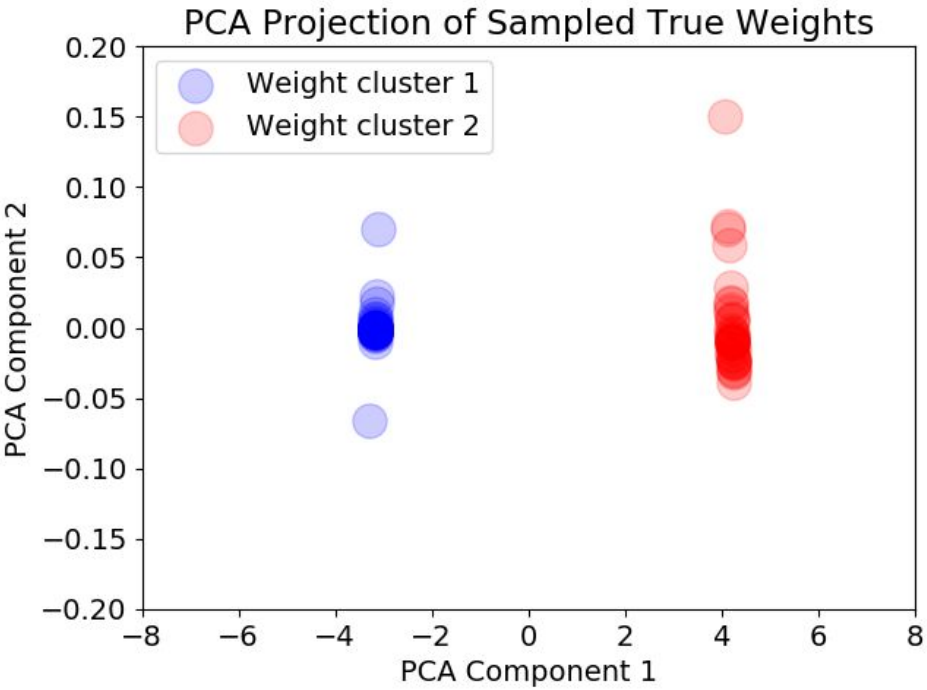
3. Black-box VI (BBVI)

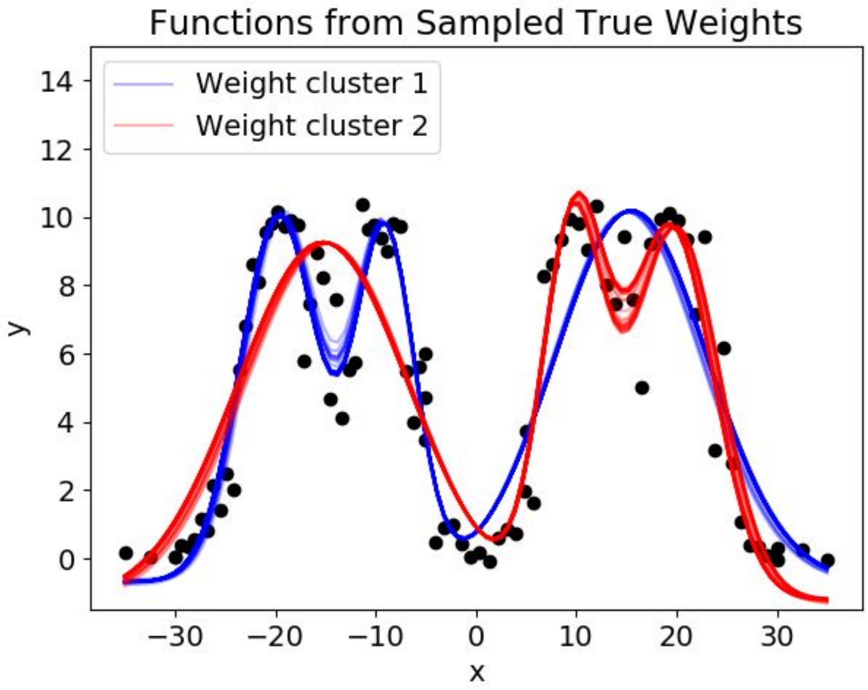
$$D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) || p(\boldsymbol{z}, \boldsymbol{\phi} | \mathcal{D})\right)$$

Principled BBVI with smart initialization

Results

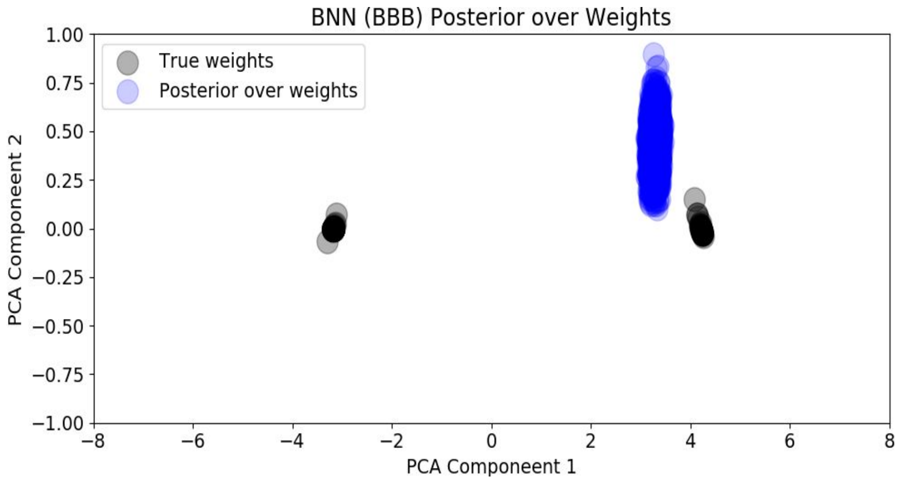
Illustrative Toy Example

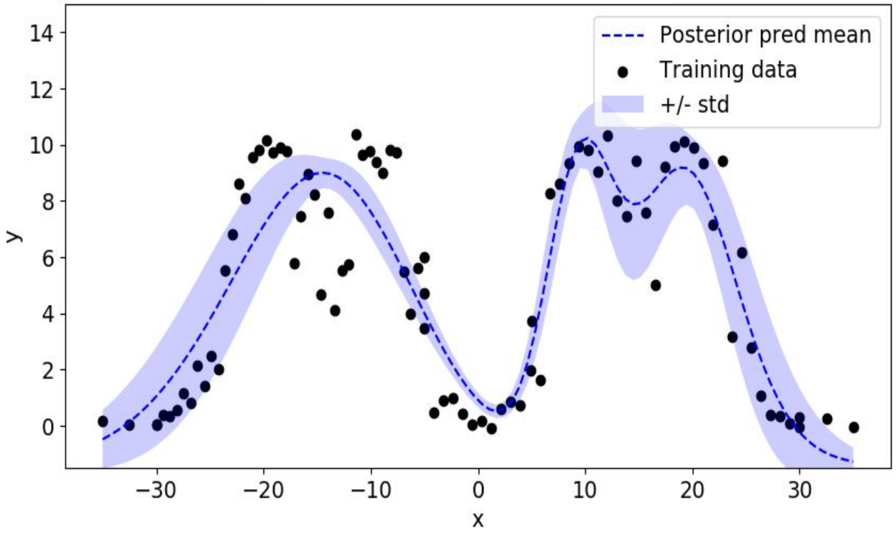




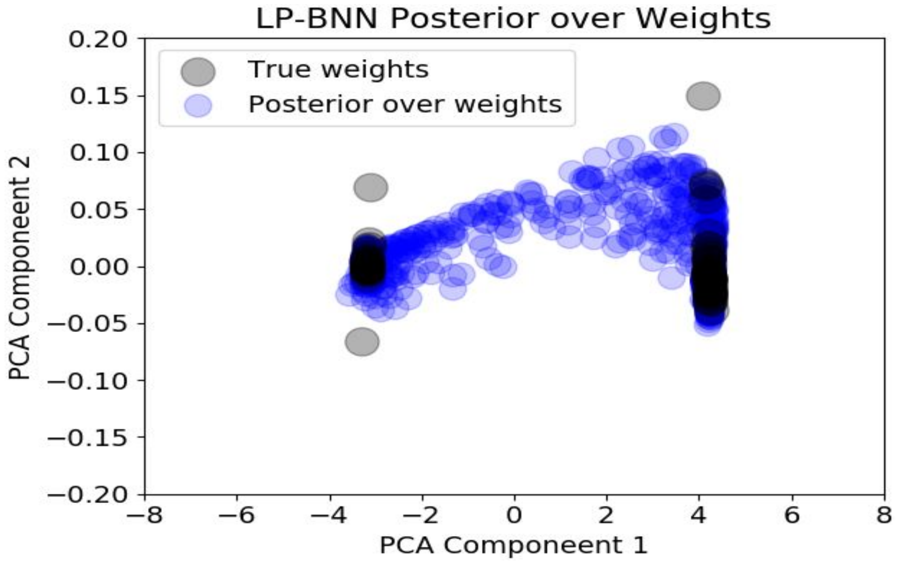
Standard BNN

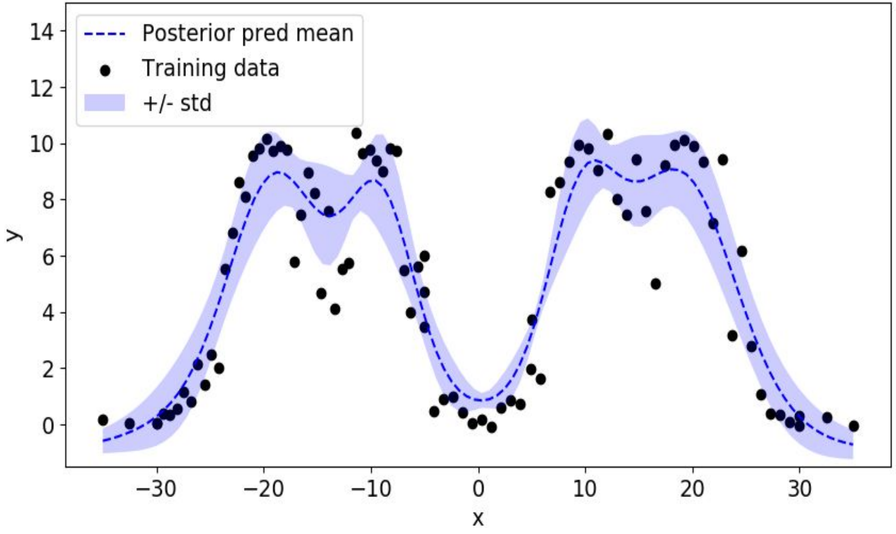
Inference with Bayes By Back Prop (BBB) [Blundell et.al, 2015]



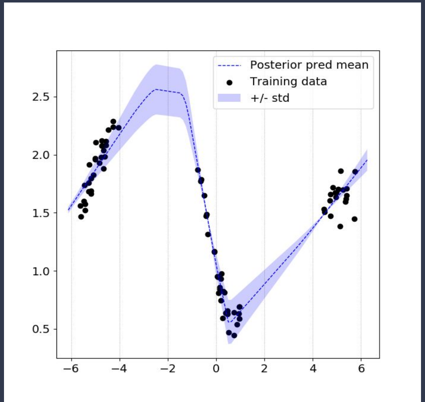


Latent Projection BNN

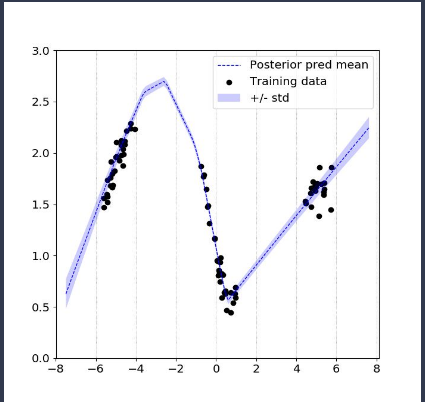




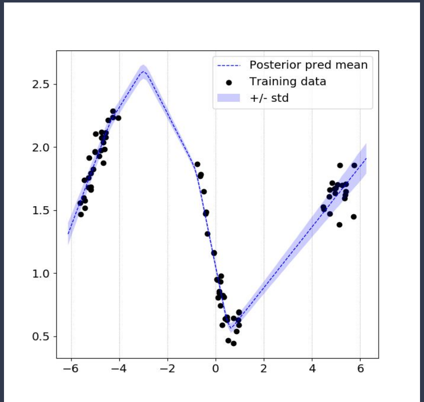
Results: Uncertainty estimation



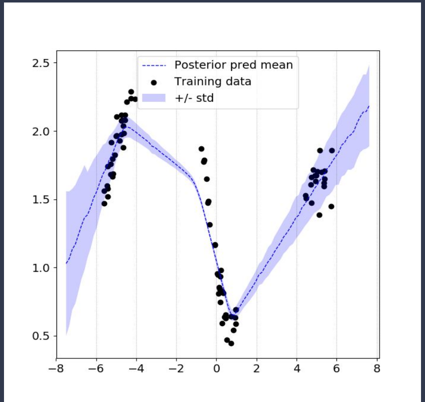
LP-BNN



BBB



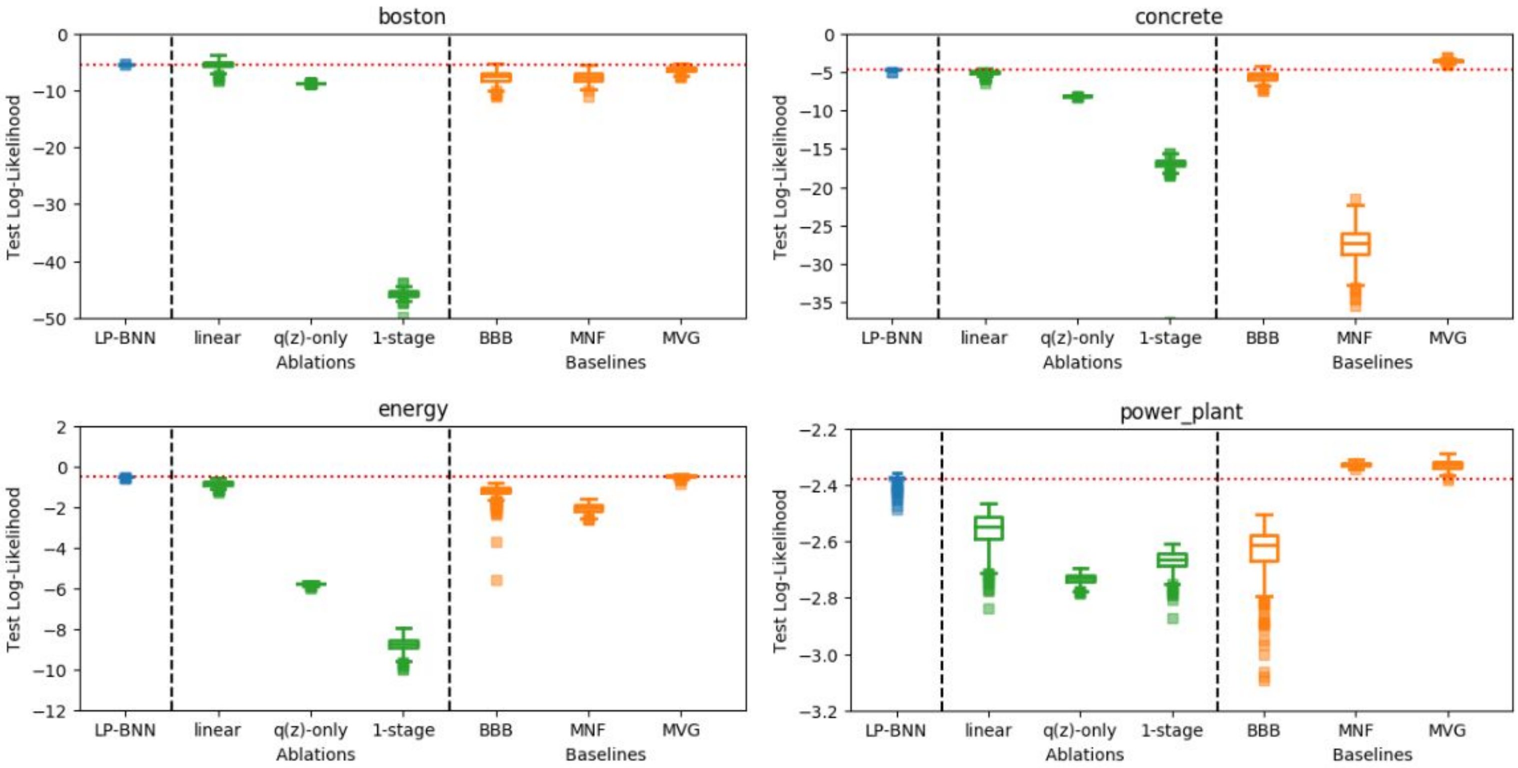
MVG



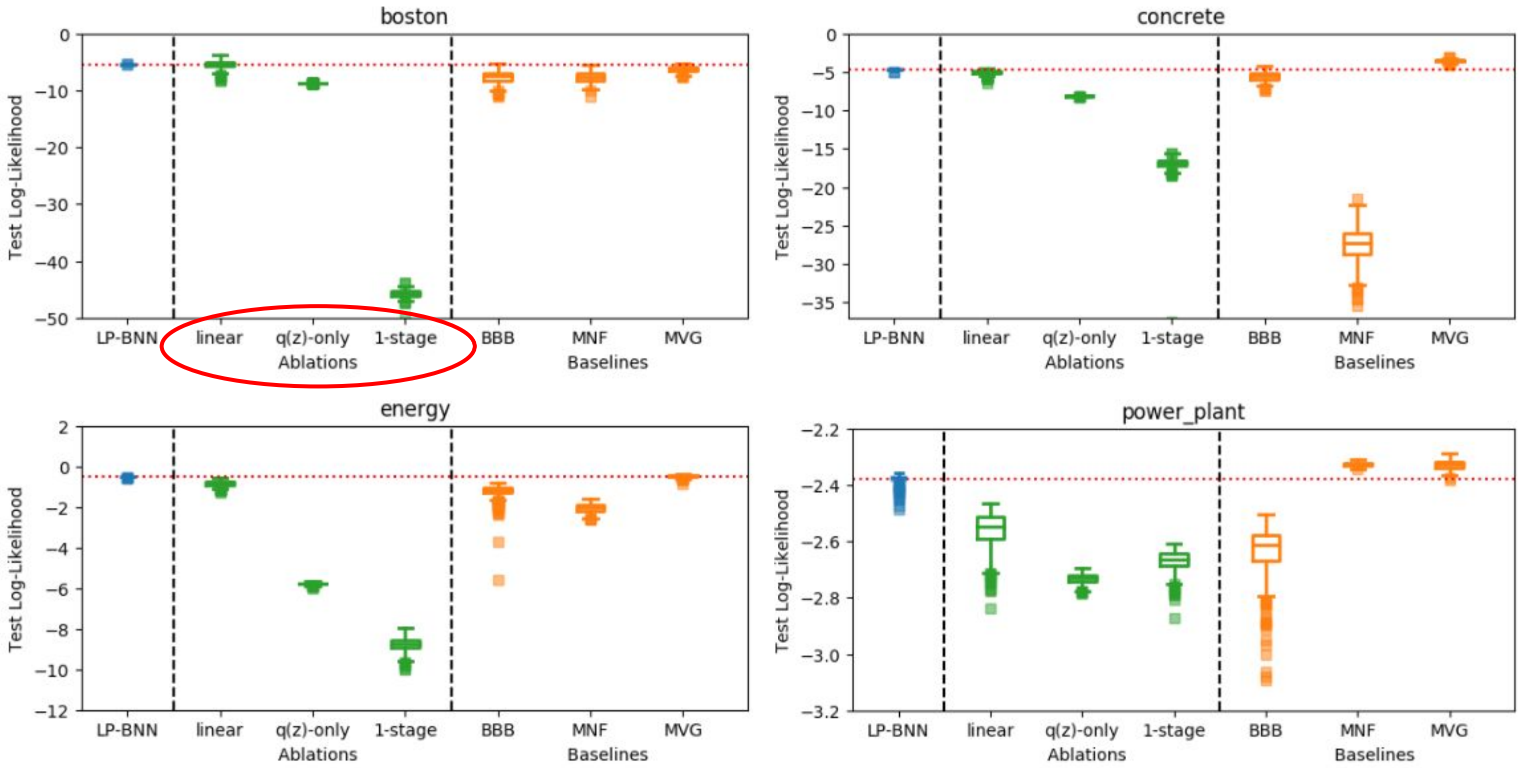
MNF

- BBB: Bayes by Back Prop [Blundell et.al, 2015]
- MVG: Multivariate Gaussians [Louizos et.al, 2016]
- MNF: Multiplicative Normalizing Flow [Louizos et. al, 2017]

Results: Generalization

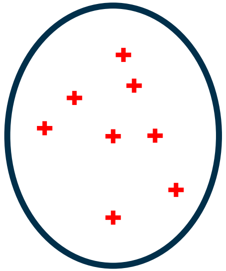


Results: Generalization

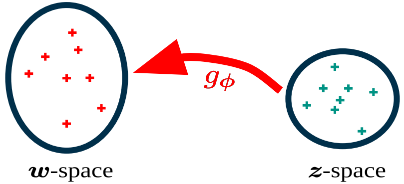


Results: Generalization (Ablations)

1. Characterize w-space



2. Find point estimate g_ϕ

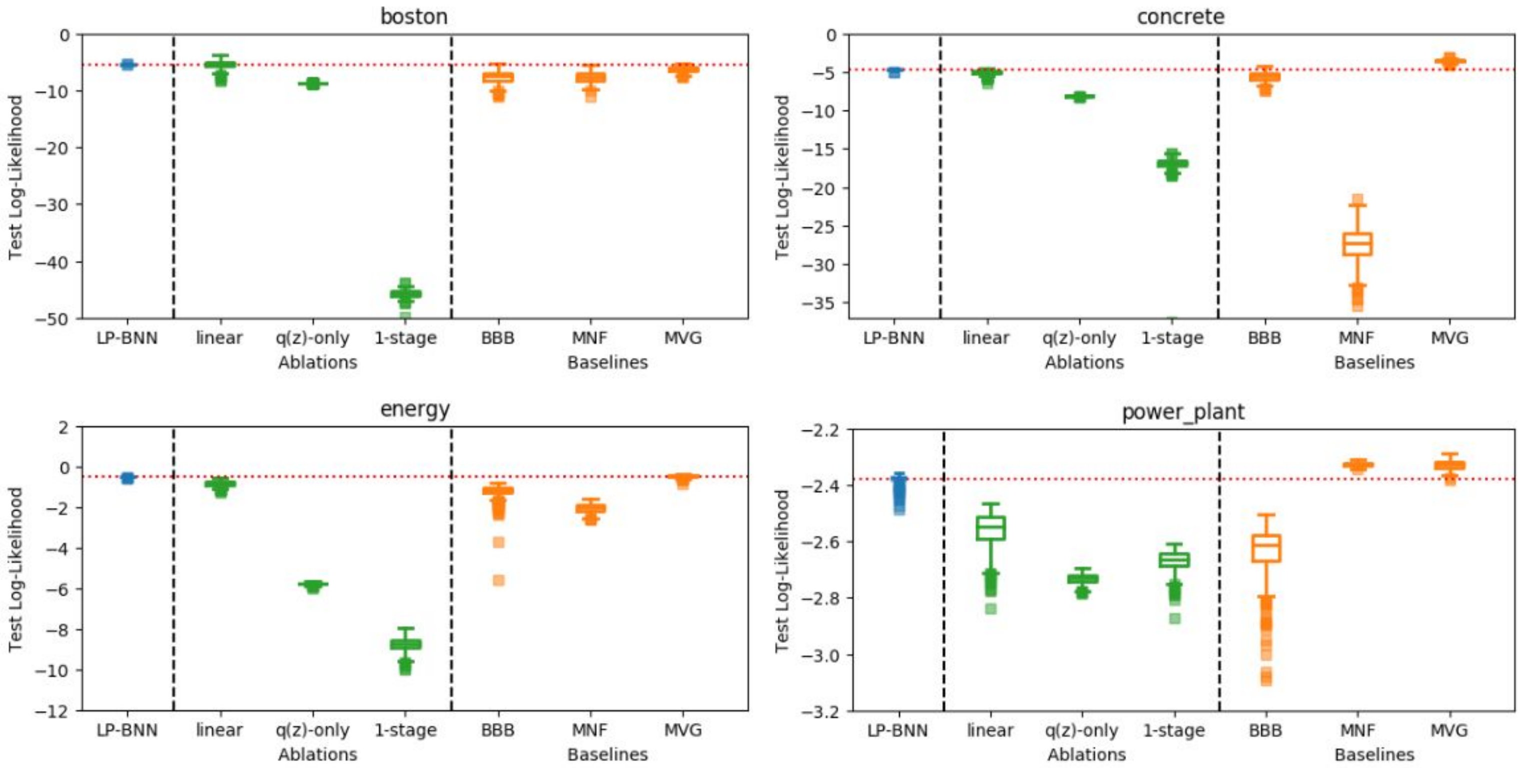


3. Black-box VI (BBVI)

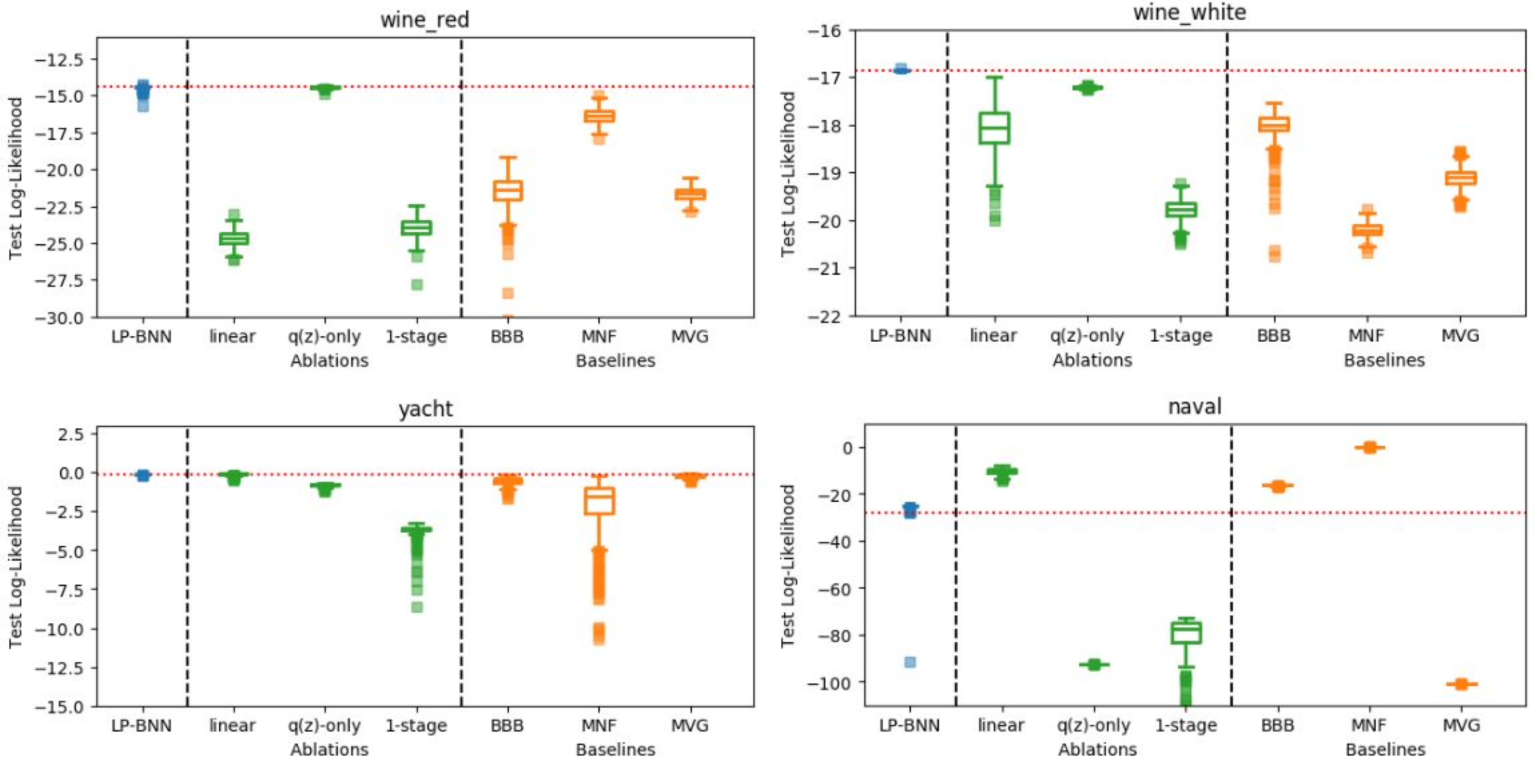
$$D_{\mathrm{KL}}\left(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) || p(\boldsymbol{z}, \boldsymbol{\phi} | \mathcal{D})\right)$$

1-stage	✗	✗	✓
linear	✓	linear	✓
q(z) only	✓	✓	$q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$

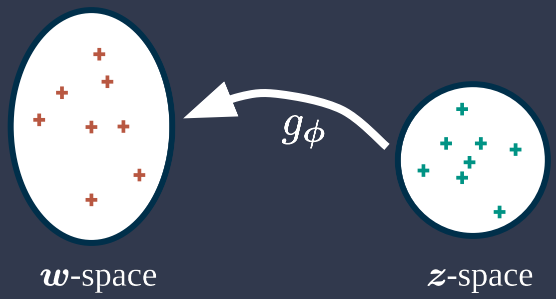
Results: Generalization



Results: Generalization



Conclusions



https://arxiv.org/abs/1811.07006

In this talk...

- Alternative modeling for BNNs
- Better approximate inference

Future improvements:

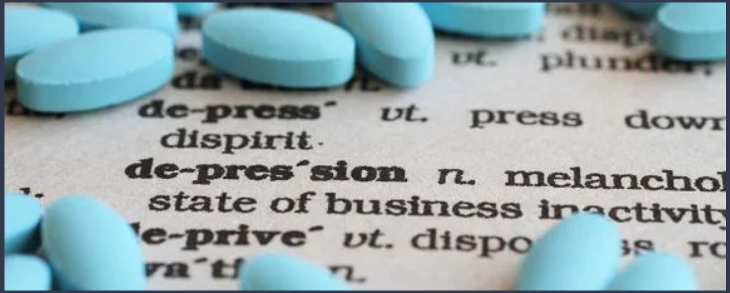
- Scalability
- Flexibility of variational distribution in latent space

Medical Applications (ongoing)

- HIV simulator
- Intensive Care Unit
- Depression Data

"Predicting treatment discontinuation after antidepressant initiation"

[Pradier et.al, 2018: submitted to JAMA]



Thank you!

Weiwei Pan

Jiayu Yao

Soumya Ghosh

Finale Doshi-Velez

VE RI TAS CRCS Center for Research on Computation and Society at Harvard John A. Paulson School of Engineering and Applied Sciences

HDSI Harvard Data Science Initiative

https://melaniefp.github.io/

Prediction-constrained Autoencoder

$$\{\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star\} = \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min_{\boldsymbol{\theta},\boldsymbol{\phi}} \Bigg\{ \frac{1}{R} \sum_{r=1}^{R} \left(\mathbf{w_c}^{(r)} - g_{\boldsymbol{\phi}} \left(f_{\boldsymbol{\theta}} \left(\mathbf{w_c}^{(r)} \right) \right) + \gamma^{(r)} \right)^2$$

$$+ \beta\, \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\frac{1}{R} \sum_{r=1}^{R} \log p\big(y | x, g_{\boldsymbol{\phi}} \left(f_{\boldsymbol{\theta}} \left(\mathbf{w_c}^{(r)} \right) \right)\big) \right] \Bigg\},$$