# Uncertainty in Bayesian Neural Networks

Javier Antorán, Xiping Liu, Efstratios Markou, Xianrui Zheng; {ja666, xl445, em626, xz396}@cam.ac.uk

UNIVERSITY OF CAMBRIDGE

## Why be Bayesian?

- Weight uncertainty: knowing what we dont know.
- Balance modelling capacity and simplicity.

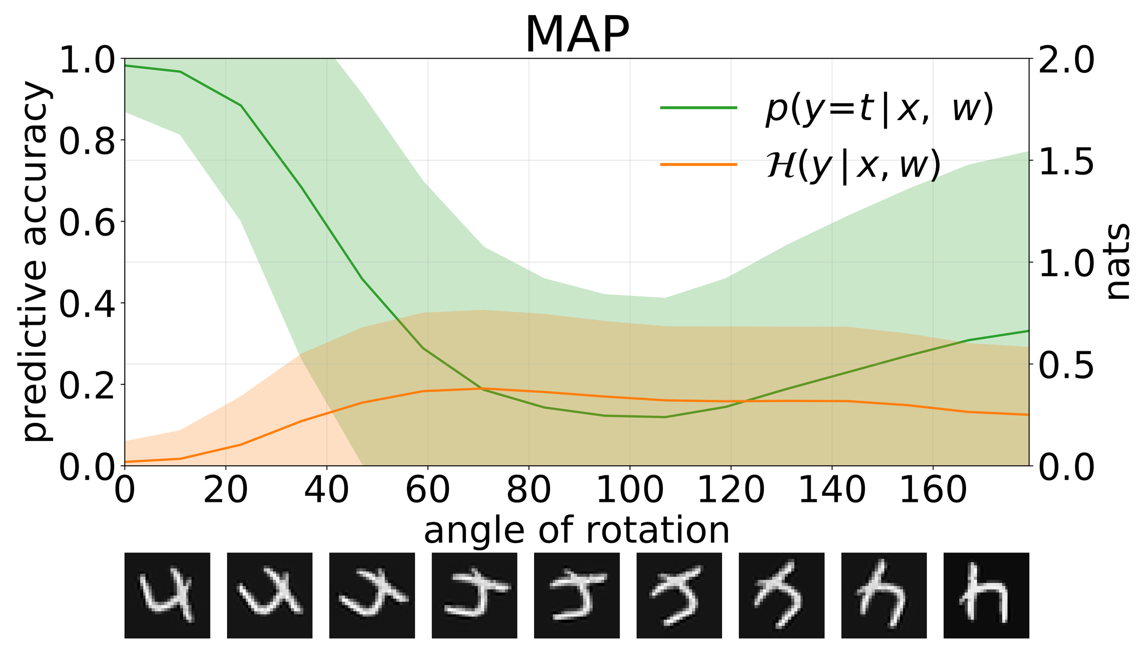


Figure: A NN trained with MAP inference presents low predictive entropy when misclassifying ood samples.

## Approximate Inference Methods

The posterior over $\mathbf{w}$ is intractable for neural nets. We consider the following approximations.

- **Bayes by Backprop** [1]

$$\text{ELBO} \approx \mathcal{L}_{BBP}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{N}\sum_{i=1}^{N}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{w}^{(i)}) - \log q(\mathbf{w}^{(i)}|\boldsymbol{\mu}, \boldsymbol{\sigma}) + \log p(\mathbf{w}^{(i)})\right]$$

$$\mathbf{w}^{(i)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(i)}; \quad \boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

- **MC Dropout** [3]

$$\text{ELBO} \approx \mathcal{L}_{drop}(\mathbf{m}) = \log p(\mathbf{y}|\mathbf{x}, \mathbf{w}) - \lambda \|\mathbf{m}\|_2^2$$

$$\mathbf{w} = \mathbf{m} \odot \mathbf{z}; \quad \mathbf{z} \sim \text{Bernoulli}(p_{drop})$$

- **Stochastic Gradient Langevin Dynamics** [4]

$$\Delta \mathbf{w}^{(i)} = \frac{\epsilon^{(i)}}{2} M\Big[\nabla \log p(\mathbf{w}^{(i)}) + \frac{N_D}{N_{batch}}\sum_{n=1}^{N_{batch}} \nabla \log p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w}^{(i)})\Big] + \boldsymbol{\eta}^{(i)}$$

$$\boldsymbol{\eta}^{(i)} \sim \mathcal{N}(\mathbf{0}, \epsilon^{(i)} M)$$

## Uncertainty Decomposition

Uncertainty caused by noise, or **Aleatoric uncertainty**, can be quantified as $\mathbb{E}_{q(\mathbf{w})}[\sigma^2_{pred}]$ or $\mathcal{H}_a = \mathbb{E}_{q(\mathbf{w})}[\mathcal{H}(\mathbf{y}'\,|\,\mathbf{x}', \mathbf{w})]$. Model or **Epistemic uncertainty** can be measured as $\text{Var}_{q(\mathbf{w})}(\mu_{pred})$ or $\mathcal{H}_e = \mathcal{H}(\mathbf{y}'\,|\,\mathbf{x}') - \mathcal{H}_a$, [2].

## Homoscedastic Regression

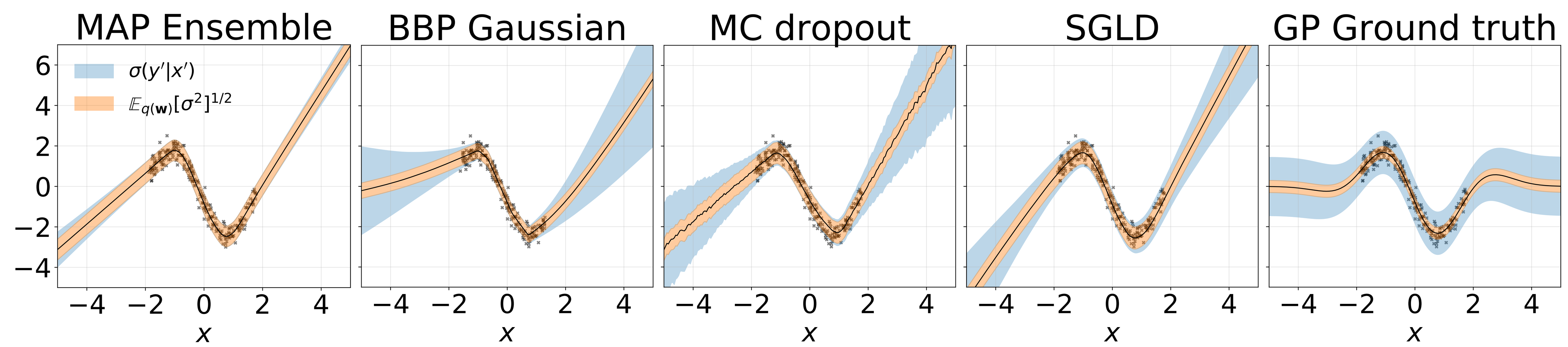


Figure: Toy homoscedastic regression task. Data is generated by a GP with a RBF kernel ($\ell = 1$, $\sigma_n = 0.3$). We use a single-output FC network with one hidden layer of 200 ReLU units to predict the regression mean $\mu(x)$. A fixed $\log\sigma$ is learnt separately.

## Heteroscedastic Regression

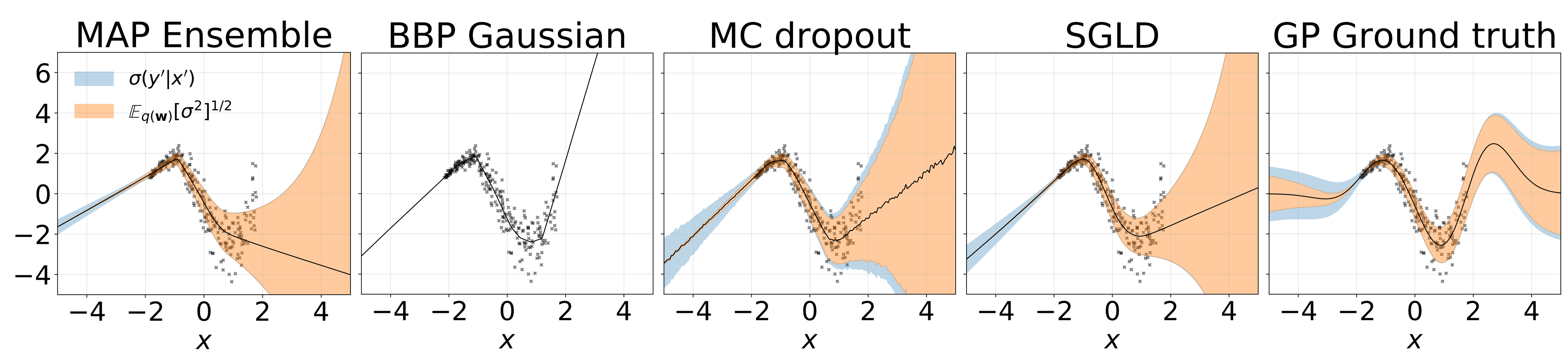


Figure: Toy heteroscedastic regression task. Data is generated by a GP with a RBF kernel ($\ell = 1$ $\sigma_n = 0.3 \cdot |x + 2|$). We use a two-head network with 200 ReLU units to predict the regression mean $\mu(x)$ and log-standard deviation $\log\sigma(x)$.

## MNIST Classification

| MNIST | MAP | MAP Ensemble | BBP Gaussian | BBP GMM | BBP Laplace | BBP Local Reparam | MC Dropout | SGLD | P-SGLD |
|---|---|---|---|---|---|---|---|---|---|
| Log Likelihood | -572.90 | -496.54 | -1100.29 | -1008.28 | -892.85 | -1086.43 | -435.458 | -828.29 | -661.25 |
| Error % | 1.58 | 1.53 | 2.60 | 2.38 | 2.28 | 2.61 | 1.37 | 1.76 | 1.76 |

Table: MNIST test results for methods under consideration. We approximate $\mathbb{E}_{q(w)}[p(\mathbf{y}'|\mathbf{x}', \mathbf{w})]$ with 100 MC samples. We use a FC network with two 1200 unit ReLU layers. If unspecified, the prior is Gaussian. P-SGLD uses RMSprop preconditioning.

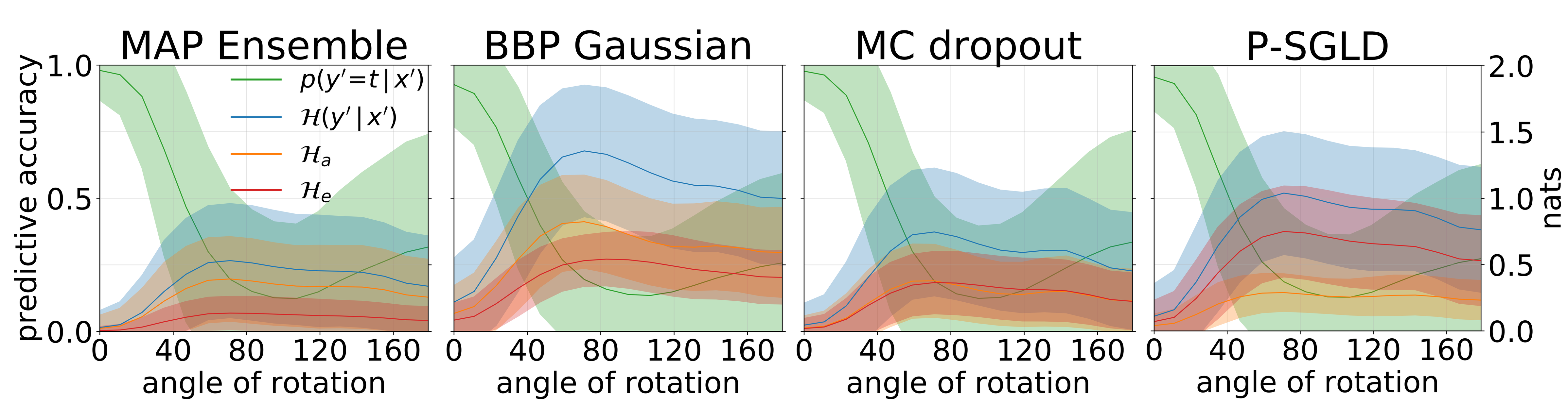


Figure: Comparison of each method's decomposed predictive entropy on ood samples: rotated MNIST digits. $t$ is the correct class.

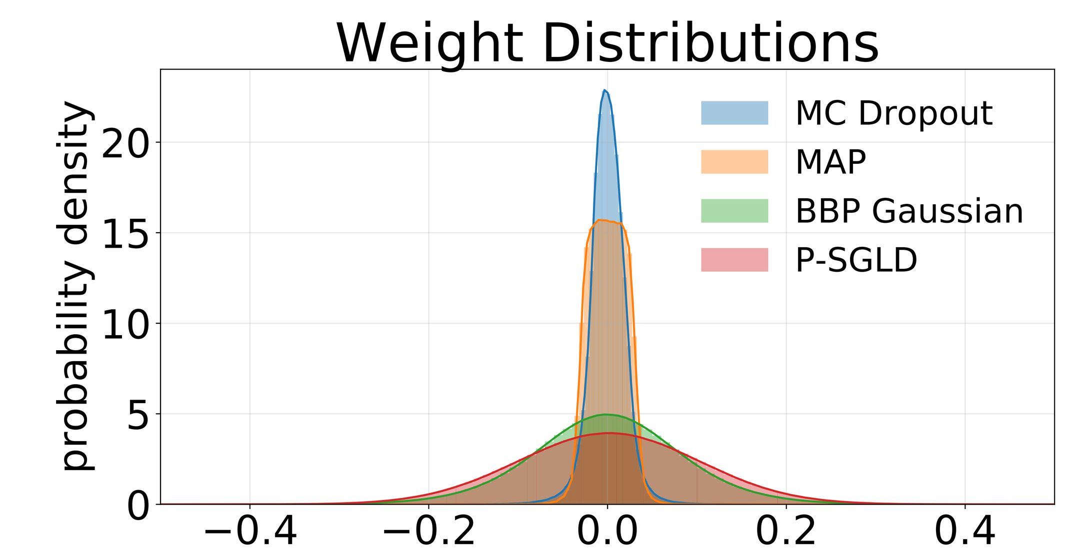


Figure: Histograms of weights sampled from each model trained on MNIST. We draw 10 samples of $\mathbf{w}$ for each model.

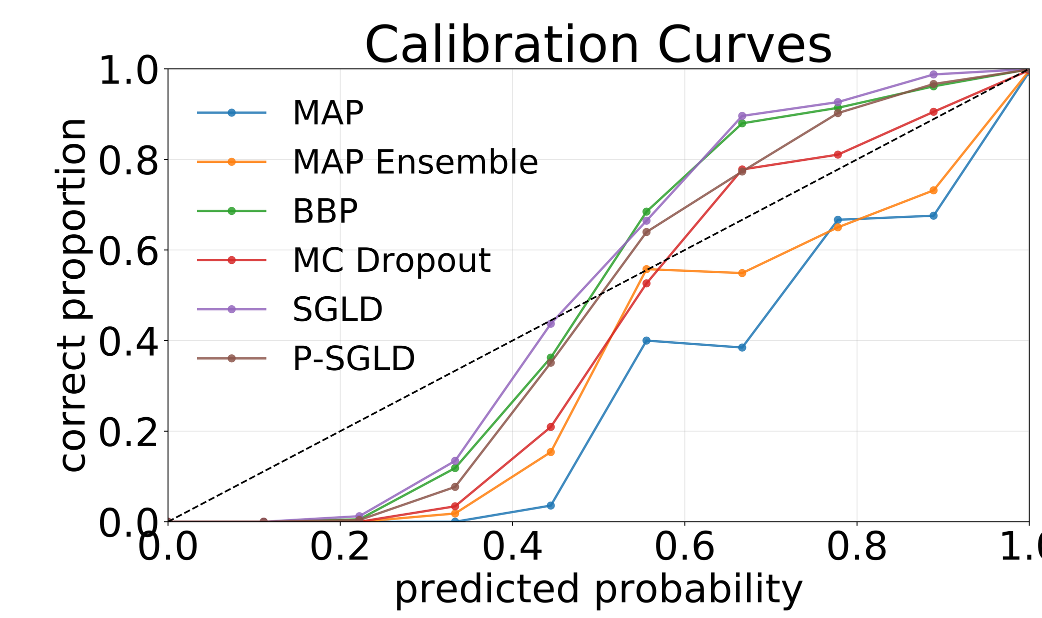


Figure: MAP results in overconfidence on MNIST-test. Approximate inference methods are underconfident for high p.

## Discussion

Bayesian methods produced plausible uncertainties on the homoscedastic task. They underestimate epistemic uncertainty on the heteroscedastic task. Additional experiments on real datasets are needed.

BBP underfits MNIST, resulting in a large aleatoric uncertainty. SGLD methods provide better epistemic uncertainty on ood samples through a less localised posterior approximation; the samples of $\mathbf{w}$ explain the data in diverse ways. Weight distributions reflect this.

Method performance varies across tasks and metric being evaluated. There is no clear best method. Optimising BBP hyperparameters is difficult.

## References

[1] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks.

[2] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning.

[3] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

[4] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics.