Master's thesis
June 2024

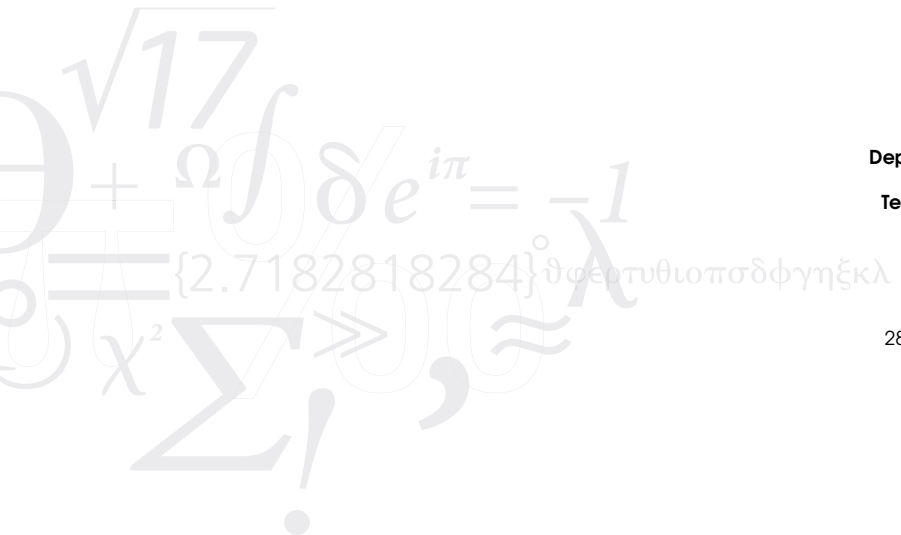
DTU Compute
Department of Applied Mathematics and Computer Science

Uncertainty Quantification in Subnetwork Ensemble Methods for Neural Networks

Authors: Yucheng Fu & Malthe Dohm Andersen

DTU

Supervisor: Mikkel N. Schmidt



DTU Compute

Department of Cognitive Systems

Technical University of Denmark

Richard Petersens Plads

Building 321

2800 Kongens Lyngby, Denmark

Abstract

Modern deep neural networks tend to be overconfident and provide poorly calibrated uncertainty estimates. To improve the reliability of deep learning models, it is necessary to develop models that accurately quantify uncertainty and correct the overconfidence of neural networks. Bayesian neural networks (BNN) and subnetwork ensembles, such as MIMO, have been shown to improve the calibration and robustness of uncertainty estimates for deep neural networks. We introduce the MIMBO neural network, which is a combination of the two aforementioned methods that combines the learnable weight posteriors of BNN with the subnetwork ensemble of MIMO. We compare the performance of MIMBO with the aforementioned methods on regression and classification tasks and evaluate them based on predictive performance and uncertainty estimation. MIMBO produces well-calibrated uncertainty estimates that outperform BNN and are comparable to MIMO on CIFAR-10 and CIFAR-100, while achieving the best results on CIFAR-10-C of the tested models. Additionally, we show that MIMBO is more uncertain when predicting on in-distribution data than MIMO and BNN due to its increased subnetwork diversity. However, this increased diversity leads to a decrease in accuracy.

All code is available in our Github repository:
`https://github.com/Malthe57/MastersThesis`

Acknowledgements

We would first and foremost like to express our gratitude to our supervisor Mikkel N. Schmidt for his supervision, encouragement and insights. His help was important for us, both in terms of guiding the direction of the project, by suggesting many angles that we had not considered, as well as his help with understanding the theory behind the methods.

We would also like to thank Andreas Lau Hansen and Lukas Wanzeck, whose solidarity and company was appreciated during the time we worked on this thesis.

Contents

CHAPTER 1

Introduction

Neural networks are the foundation of deep learning [1], which is a discipline within the field of machine learning. Deep learning has in recent years contributed to major improvements in areas such as natural language processing [2], image classification [3], object detection [4], and medical image analysis [5]. It is already powering many common consumer products, such as cameras and smartphones [1], and is also utilised in decision-making systems in fields such autonomous driving [6] and AI-assisted medical decision making [7].

For these kinds of applications, it is crucial that the machine learning model is accurate and provides reliable estimates of its uncertainty alongside the predictions, such that predictions with high uncertainty can be ignored or checked by a human expert [8, 9]. For classification tasks, this entails outputting a confidence score along with its predicted label [10].

Modern deep neural networks with high accuracy tend to be poorly calibrated, as the predicted confidences do not reflect the true correctness likelihood of the network [11]. They often suffer from overconfidence issues, which in the context of classification means that predictions are assigned too high of a confidence score [12]. This is especially a problem when evaluating out-of-distribution examples whose underlying data distribution is different from the data that the model is trained on, as it often results in high confidence for incorrect predictions [13]. To improve the trustworthiness of neural networks, it is therefore important to study and develop uncertainty quantification methods that provide well-calibrated uncertainty estimates and are robust to dataset shifts.

Two popular techniques for quantifying uncertainty in deep neural networks are Bayesian neural networks (BNNs), which learn a distribution over its weights, and ensemble methods, which involves training multiple independent models and aggregating their predictions. Ensemble methods are primarily designed with the purpose of improving predictive performance [8], but they have been shown to also provide reliable uncertainty estimates for predictive tasks [14]. This is in contrast to Bayesian neural networks that were introduced with uncertainty quantification as one of its main advantages [15].

Blundell et al. [16] introduce an efficient and backpropagation-compatible algorithm for learning the distribution of weights in neural networks called Bayes by backprop.

Rather than assigning a value to each weight in the neural network, this approach learns a posterior distribution over possible weight values. The posterior distribution of weights cannot be estimated using exact Bayesian inference due to the number of weights in modern neural networks. Instead, a Gaussian distribution is used to approximate the Bayesian posterior using variational inference. Rather than updating each weight during backpropagation, the variational parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta)$ of the Gaussian distribution for the weights are updated instead. This doubles the number of weights compared to a deterministic neural network, but the Bayesian neural network is able to quantify uncertainty in the model weights.

During training and inference, weights are sampled from the variational posterior distribution. By sampling weights from the distribution, each weight sample is essentially one realisation of an independently trained neural network. By aggregating the prediction of multiple models whose weights are sampled from the posterior, one can reduce the weight uncertainty. Blundell et al. find that the Bayesian neural network achieves predictive performance comparable to models trained with dropout.

Bayesian neural networks are theoretically well-founded and motivated by Bayesian principles, but in practice ensemble methods tend to perform better in both accuracy and uncertainty estimation [17]. One of the main disadvantages of ensemble models is the computational costs of having to train and run inference on multiple models [18].

Consequently, new ensemble methods have been developed to improve the parameter efficiency of ensemble models. Havasi et al. [19] present the multi-input multi-output (MIMO) configuration for neural networks that enables training multiple subnetworks within a neural network. This allows a neural network to function as an ensemble without the additional costs that ordinary ensembles have during training and inference. By giving each subnetwork different input data during training, the subnetworks learn to independently solve a predictive task. At inference, each subnetwork is given the same data and are evaluated as an ensemble by averaging over the outputs of each subnetwork.

Training subnetworks within a network is theoretically motivated by overparameterisation in deep neural networks, which allows large parts of a network to be pruned after training without impacting the performance [20, 21].

1.1 Aim of the thesis

In this thesis, we propose a novel subnetwork ensemble method that combines the ideas of Bayes by backprop BNNs and MIMO neural networks. We call the combination of these two methods multi-input multi-Bayesian output (MIMBO) neural networks. Both methods lead to more robust uncertainty estimates, albeit in different manners. Bayes-by-backprop affects how the weights in neural networks are learned, while MIMO exploits the overparameterisation of neural networks to actualise multiple subnetworks within a network. As the two methods improve on different parts of a single neural network, we hypothesise that they will act complementarily and that combining the two methods can yield a model with more well-calibrated uncertainty estimates and comparable predictive performance. The goal of this work is to implement and test the aforementioned methods in order to answer the following research question:

How do MIMBO neural networks compare to Bayes by backprop Bayesian neural networks and MIMO neural networks in terms of providing accurate predictions and well-calibrated uncertainty estimates?

1.2 Scope

In this thesis, we primarily study Bayesian neural networks and the subnetwork ensembles MIMO and MIMBO. For comparison, we also study the multi-headed Naive neural network, a MIMO-configured model with a different training scheme, and a deterministic neural network that we use as a baseline. We do not consider other ensembles, such as deep ensembles, in order to focus on models that are contained in a single neural network.

We evaluate the models on tabular regression tasks, both toy data and real-life data (Communities and Crime), and image classification tasks, CIFAR-10 and CIFAR-100. We study their performances on both in-distribution and out-of-distribution data.

All models are evaluated in terms of their predictive performance and uncertainty estimates. Predictive performance is measured as root mean squared error (RMSE) for the regression tasks and as accuracy for the classification tasks. Uncertainty estimates are evaluated by observing model calibration on reliability diagrams and using summary statistics, such as the expected calibration error (ECE) and the negative log likelihood (NLL). Summary statistics provide an indication of the overall calibration, while reliability diagrams offer a more detailed view of a model's calibration, which we will use to evaluate how well-calibrated a model is at various levels of confidence.

1.3 Thesis outline

The thesis is structured as follows:

Literature study: This section provides an overview of relevant model calibration and uncertainty quantification methods, and provide the theoretical motivation for subnetwork ensemble methods.

Theoretical Concepts: In this section we introduce the necessary theory on machine learning and Bayesian probability theory to understand the methods in the rest of the thesis.

Methods: This section contains a thorough description of our models: the baseline, BNN, MIMO, Naive and MIMBO, and how we implement and train them. Moreover, we present the relevant evaluation metrics and visualisation methods and explain how they are read and interpreted. Finally, we introduce the datasets we use and explain the experiments we conduct.

Results: In this section, we present the experimental results and analyse the performances of BNN, MIMO and MIMBO in terms of predictive performance and uncertainty quantification.

Discussion and Conclusion: We discuss the most important results and takeaways from our experiments and interpret them in the context of other similar studies. We discuss the strengths and limitations of the proposed MIMBO model, and conclude on the viability of MIMBO for uncertainty quantification.

CHAPTER 2

Literature Study

As the aim of the thesis is to obtain robust and well-calibrated uncertainty estimates, it is necessary to understand what methods already exist for this purpose, as well as understand why these methods work. In this chapter, we aim to provide an overview of the problem that is calibration of neural networks and relevant methods for achieving well-calibrated uncertainty estimates.

Well-calibrated uncertainty estimates can be obtained using post-hoc calibration methods or obtained directly from models using Bayesian neural networks or neural network ensembles [8]. Post-hoc calibration methods adjust uncertainty estimates after training using hold-out data, whereas Bayesian neural networks and neural network ensembles both use model averaging to obtain robust uncertainty estimates. This thesis focuses on Bayesian neural networks using variational inference and subnetwork ensembles, but understanding what other relevant methods exist in the literature is of great importance.

The literature review begins with explaining **Calibration of neural networks** and sources of miscalibration. This section explains what calibration is and why it is a relevant problem for neural networks. Some post-hoc methods for improving calibration are also presented to provide context for the model averaging methods that we focus on in this work.

We proceed to **Approximate Bayesian inference methods for neural networks** and briefly explain various methods used for approximate Bayesian inference in neural networks. Our aim is to provide perspective on the variational inference method that we use and give an overview of other common methods in the literature.

Then follows an introduction to neural network **Ensemble Methods**. The benefits of ensembles of neural networks are explained, along with how ensembles benefit from model diversity. A few ensembling methods relevant to uncertainty quantification are presented along with their advantages and disadvantages.

Finally, we focus on **MIMO neural networks** and the theoretical motivation for subnetwork methods. As MIMO models are central to this thesis, we explain the argument for why multiple subnetworks can be parameterised within one network. Lastly, we briefly go over some other methods that build upon MIMO neural networks to

give perspective to future work.

2.1 Calibration of neural networks

Calibration in machine learning refers to the proportionality between a model's prediction accuracy and its predictive uncertainty. For instance, in a classification setting a model is well-calibrated if the predicted confidence matches the likelihood of it being correct [11]. Well-calibrated models are therefore essential in uncertainty quantification, as uncertainty estimates from a poorly calibrated model can be misleading [22].

Guo et al. [11] show that modern deep neural networks are more poorly calibrated than older neural networks, despite achieving better predictive performance. These models are overconfident in most cases, meaning that they are more confident in their predictions than the predictions are accurate. They find that there are several factors that contribute to this issue. Some miscalibration is attributed to the increased capacity of neural networks, which improves the ability to generalise, but also results in worse calibration. Guo et al. note that the introduction of batch normalisation, which reduces the need for additional regularisation, also causes models to be more miscalibrated. Finally, miscalibration may be caused by the neural network overfitting the negative log likelihood (NLL). Guo et al. explain that since NLL becomes minimised by having higher confidence in correct predictions, the model learns to predict with high confidence on the training set. Overfitting the NLL does not negatively impact the prediction accuracy at inference time, but does negatively impact the test NLL.

Methods to improve calibration are mainly centred on improving the uncertainty estimates, as prediction accuracy is usually already maximised in a machine learning model. For instance, calibrating an underconfident model by simply lowering the accuracy is undesirable for obvious reasons. There are two approaches to improving the uncertainty estimates of a given machine learning model. The first approach is to recalibrate the models after training by adjusting the uncertainty estimates. The second approach is to alter the machine learning method to improve the initial uncertainty estimates. In this chapter we will discuss both approaches, starting with post-training calibration.

2.1.1 Post-hoc calibration methods

Various methods for calibrating the uncertainty estimates of a trained model exist. A major benefit of calibrating a model after training is that the predicted class probabilities, i.e. the uncertainty estimates, can be improved without changes to the model itself. Calibration methods for classification tasks take the predictive probabilities of a model and improves them post-hoc. This makes calibration methods a good

solution for overconfident out-of-the-box uncertainty estimates. Calibration methods typically require a calibration dataset that is a hold-out set of the training data. If a validation set is used for hyperparameter tuning it can also be used for calibration [11] to avoid reducing the training set size further.

Since the topic of calibration often concerns classification, many calibration methods are made for that task. Histogram binning [23] is a calibration method for binary classifiers, with further extensions for multi-class classification [11]. It is a simple to implement method, but since it scales prediction probabilities individually, it can also change the prediction label, and thus affect the accuracy of the calibrated model.

Platt scaling [24] is a parametric calibration method that calibrates probabilities by fitting the output logits with logistic regression. Compared to a method like histogram binning it has the advantage of preserving prediction accuracy as the scaling is monotonic. The main disadvantage of Platt scaling is that it assumes that the distibution of probabilities prior to scaling are similar to a sigmoid function, which is not the case for all models. Its extension, temperature scaling [11], works for multi-class classification by scaling the output logits of a network before applying softmax. Temperature scaling requires choosing a suitable value for the temperature parameter, which can be done manually or learned through fitting to the calibration data. In return, it offers monotonic scaling of multiclass prediction logits while being simple to implement. Beta calibration [25] offers an alternative parametric calibration method to Platt scaling that maps the prediction class scores using a beta distribution rather than a logistic distribution. This method is as simple to implement as Platt scaling while being better suited for some cases, e.g. if data has heteroscedastic noise.

2.2 Approximate Bayesian inference methods for neural networks

Bayesian neural networks learn a posterior distribution over its weight parameters and demonstrate potential to quantify reliable uncertainty estimates [15]. By sampling multiple times from the posterior, one can obtain multiple different instances of the same neural network and ensemble the predictions made by the different networks. Exact Bayesian inference for learning the posterior is intractable for large neural networks, so instead approximate Bayesian methods are used.

In this section, we provide an overview of approximate Bayesian inference methods that are commonly used for Bayesian neural networks: variational inference, Markov chain Monte Carlo methods, and Laplace approximation. We describe the intuition behind the methods and give an overview of some relevant papers that apply them to Bayesian neural networks.

2.2.1 Variational inference

Variational inference is an optimisation-based method for approximate Bayesian inference. The approximate variational posterior $q(\boldsymbol{w}|\boldsymbol{\theta})$ belonging to a variational family $\mathcal{Q}$ is optimised by updating its variational parameters $\boldsymbol{\theta}$ to minimise a "distance" measure, the Kullback-Leibler divergence, between the approximate variational posterior $q(\boldsymbol{w}|\boldsymbol{\theta})$ and the true posterior $p(\mathbf{w}|\mathcal{D})$ [26].

Hinton & van Camp [27] first introduced the idea of estimating a simple, analytically tractable approximation to the true posterior weight distribution of a neural network using variational inference [28]. Graves [29] later introduced a method that does not require analytical solutions, making it scalable to larger neural networks. The method uses gradient descent for updating the variational posterior and derives unbiased gradient estimates using the Gaussian characteristic function [30].

Building upon their work, Blundell et al [16] introduce the Bayes by Backprop algorithm, which seeks to learn a Gaussian approximation to the true weight posterior using variational inference. Unlike the method by Graves, the Bayes by Backprop algorithm obtains low-variance gradient estimates using the reparameterisation trick. They demonstrate that this subtle change makes Bayes by Backprop comparable to dropout in terms of predictive performance.

One drawback of these variational inference methods is that they introduce extra variational parameters for each weight in the neural network. For instance, a mean-field Gaussian variational posterior requires doubling the number of parameters in the neural network, as the Gaussian distribution is parameterised using a μ and σ value. Parameter-efficient variational inference methods for neural networks have been proposed for this problem.

Dusenberry et al. [15] present Rank-1 BNN, a parameter-efficient method for variational inference in neural networks inspired by BatchEnsemble [18], where the weights in each layer is parameterised using a shared weight matrix W and vectors r and s that are sampled from the variational posteriors $q(r)$ and $q(s)$ respectively. Posterior weight samples are obtained through element-wise multiplication of W with the resulting rank-1 matrix that is the outer product of r and s^T. Rank-1 BNNs outperform BatchEnsemble and mean-field variational inference BNNs (like Bayes by Backprop) on CIFAR10 and CIFAR100 in terms of accuracy, NLL and ECE.

2.2.2 Markov chain Monte Carlo (MCMC) methods

Markov chain Monte Carlo (MCMC) sampling has been a dominant paradigm in the field of approximate Bayesian inference for decades, because they can generate samples from a wide range of distributions [26, 31]. From an initial sample, more samples are generated by iteratively applying a Markov transition, which can be thought of

as a conditional probability density $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x})$ for the next sample $\mathbf{z}_t$ conditioned on the current sample $\mathbf{z}_{t-1}$ [32]. The aim is to construct an appropriate Markov transition such that the limiting distribution of the Markov chain at convergence is the posterior distribution of interest that we want to sample from [33].

One drawback of MCMC methods is that they may take a very long time to converge to the desired distribution can take a very long time [34]. Furthermore, the samples are autocorrelated (and not i.i.d) due to the Markov transition [32].

There exists several algorithms for constructing Markov chains that converge to the target distribution. The most important ones are the Metropolis-Hastings (MH) algorithm [35, 36

2.3 Ensemble Methods

Another group of methods for estimating uncertainties are so-called ensemble methods where multiple neural networks are trained independently and an ensemble prediction is obtained by combining their predictions.

Neural network ensembles were first proposed by Hansen and Salamon [44], introducing the idea of training multiple neural networks on the same dataset to ensemble over their predictions. Hansen and Salamon use a voting ensemble for a classification problem, letting the prediction be the label that most ensemble members predict. The main advantage of using an ensemble of neural networks is that the ensemble generalises better and has better predictive performance than a single neural network.

Recent work suggests that ensembles of deep neural networks offer additional benefits other than improvement improvement in predictive performance, namely uncertainty quantification and robustness to dataset shift [14, 45]. In this section, we give an overview of some (non-Bayesian) ensemble methods for uncertainty quantification in neural networks.

2.3.1 Monte Carlo (MC) dropout

Dropout is a regularisation technique for neural networks where elements in the input to a layer are randomly set to 0 during training with dropout rate p. [46]. Gal et al. [45] present a framework using dropout to estimate predictive uncertainty for neural networks, called MC-dropout. The main idea behind MC-dropout is to perform T stochastic forward passes with dropout at inference time and then average the T outputs. Intuitively, this is equivalent to creating an ensemble with T different models [14]. In a later paper, Gal et al. [47] extend MC-dropout to convolutional neural networks by adding dropout layers after convolutional layers.

MC-dropout is relatively easy to implement [10], however, it is computationally expensive at inference, as it may require many forward passes per batch to get reasonable uncertainty estimates for classification tasks [45].

2.3.2 Deep ensembles

Lakshminarayanan et al. [14] propose a framework for estimating predictive uncertainty using an ensemble of deep neural networks. The ensemble is treated as a uniformly-weighted mixture model that averages the predictions of each ensemble member. They find that ensembles with just $M = 5$ members improve uncertainty estimates and express higher uncertainty on out-of-distribution samples compared to baseline models and MC-dropout.

6Q`i 2i HX (;m2 i? i M BKTQ`i Mi 72 im`2 Q7 2Mb2K#H2b Bb i?
i?2B` 2Mb2K#H2 K2K#2`bX 6Q` BMbi M+2- B7 HH i?2 M2m` H
QmiTmi i?2 b K2 T`2/B+iBQMb- i?2`2 Bb MQ ; BM BM 2Mb2K#H2
6m`i?2`KQ`2- 6Q`i 2i HX b?Qr i? i 2Mb2K#H2b rBi? K2K#2`b i
7`QK b+` i+? T`QpB/2 ;`2 i2` /Bp2`bBiv +QKT `2/ iQ bm#bT +2
b ` M/QK T2`im`# iBQM Q7 i?2 r2B;?ib M/ J* /`QTQmiX h?2 BM
KQ/2Hb K M ;2 iQ 2tTHQ`2 /Bz2`2Mi KQ/2b BM i?2 7mM+iBQM
b KTHBM; K2i?Q/b QMHv 2tTHQ`2 bBM;H2 KQ/2X //BiBQM H
K2i?Q/b rQ`F +QKTH2K2Mi `v iQ i?2 2Mb2K#H2b- rBi? +QK#B
Q/b T`QpBM; #2ii2` BM i2`Kb Q7 T`2/B+iBQM ++m` +v- 2bT2+
K2K#2`bX

1BSBNFUFS FRDJFOU FOTFNCMF NPEFMT

HBKBi iBQM Q7 i?2 7` K2rQ`F T`QTQb2/ #v G Fb?KBM ` v M
i iBQM H #QiiH2M2+F 7Q` i?2 BMi KQ/2HbX S ` K2i2`@2{+B2Mi
2tTHQBi r2B;?i b? `BM; BM M2m` H M2irQ`Fb iQ `2/m+2 i?2 Mm

q2M 2i HX (Mi`Q/m+2 " i+?1Mb2K#H2- K2i?Q/ 7Q` +QMbi`m+iB
rBi? HQr2` K2KQ`v M/ +QKTmi iBQM H +QbibX AM 2 +? H v2`-
b? `2/ r2B;?i K i`Bt HH 2Mb2K#H2bX 1 +? 2Mb2K#H2 r_i? b irQ H2
M s_i- M/ 2 +? 2Mb2K#H2 K2K#2` Q#i BMb mMB[m2 r2B;?i K i`
rBb2 KmHiBTHB+ iBQM #2ir22M i?2 b? `2/ r2B;?i K i`Bt M/ Q
r_i M s_i^TX G2 `MBM; i?2 pB2r $r_i s_i^T$ b? `2/ Q` i? i? 2B` T`Q/m+i `2
H2bb r2B;?ibX " i+?1Mb2K#H2 KQ/2Hb `2 K `;BM HHv 7 bi2` i
2Mb2K#H2 KQ/2Hb- #mi rBi? +QKT ` #H2 ++m` +v M/ mM+2`i B
#BHBiv 2biBK iBQM (Mx2H 2i HX- H/ mTQM i?2 rQ`F Q7 q2M 2i HX M
>vT2`@# i+?2Mb2K#H2b- r?2`2 2 +? Q7 i?2 2Mb2K#H2 K2K#2`
?vT2`T ` K2i2`bX h?2v }M/ i? i +QK#BMBM; KQ/2Hb rBi? /Bz2`
K F2b >vT2`@# i+? 2Mb2K#H2b QmiT2`7Q`K " i+?1Mb2K#H2b
1*1X

.VMUJ JOQVU .VMUJ PVUQVU OFVSBM O

JmHiB@BMTmi JmHiB@QmiTmi UJAJPV M2m` H M2irQ`Fb Bb
2i HX (N(iQ i` BM T ` K2i2`@2{+B2Mi M2m` H M2irQ`F 2Mb2K#H
BM; KmHiBTH2 BM/2T2M/2Mi M2m` H M2irQ`Fb- i?2 2Mb2K#H
bm#M2irQ`Fb rBi?BM bBM;H2 M2m` H M2irQ`FX h?2 mi?Q`b
};m` iBQM Bb bBKTH2 iQ BKTH2K2Mi 7Q` M 2tBbiBM; M2m` H
QMHv `2[mB`2b KQ/B7vBM; i?2 BMTmi M/ QmiTmi H v2`b Q7 i?2

T`2/B+iBQMbX >Qr2p2`-rmvi?2bbkM/HBHM+? M;2 T ` K2i2`Bb2b
/2Mi bm#M2irQ`Fb rBi?BM M2m` H M2irQ`F K v MQi #2 2MiB`2

AM i?Bb b2+iBQM- r2 T`QpB/2 i?2 i?2Q`2iB+ H KQiBp iBQM #
/2T2M/2Mi bm#M2irQ`Fb rBi?BM QM2 M2irQ`F M/ Tmi JAJP M
+QMi2ti Q7 2tBbiBM; HBi2` im`2X JQ`2Qp2`- r2 ;Bp2 M Qp2`p
i? i #mBH/ mTQM Q` r2`2 BMbTB`2/ #v JAJPX

/FVSBM OFUXPSL DPNQSFTTJPO BOE PWFSQBSB

L2m` H M2irQ`F +QKT`2bbBQM K2i?Q/b `2 KQiBp i2/ #v p `B
T`QpBM; T`2/B+iBQM M+m2/mv+BM; i?2 bBx2 Q7 i?2 M2irQ`F M
iBQM H BMi2MbBivkBXBMi?2 b2M+2M(b?QrM i? i M2m` H M2irQ`F
i2+?MB[m2b + M `2/m+2 i?2 MmK#2` Q7 T ` K2i2`b #v mT iQ N
BKT +iBM; i?2 T`2/B+iBp2 T2`7Q`K M+2kBX7 i?2 M2m` H M2irQ`

q2HH@bim/B2/ +QKT`2bbBQM i2+?MB[m2b BM+Hm/2 FMQrH2/;
7`QK M 2Mb2K#H2 Q7 M2m` H M2irQ`Fb Bb /BbiBHH2/ BMiQ
rQ`F #v i` BMBM; i?2 bK HH2` M2m` H M2irQ`F iQ K i+? i?2 T`
2Mb2K#H2 kQ8/yXH (MQi?2` i2+?MB[m2 + HH2/ M2m` H M2irQ`F
`2KQpBM; mMBKTQ`i Mi M2irQ`F +QMM2+iBQMb # b2/ QM ?2r
i?2 `2K BMBM; bmk#R8XQhF2(b2 +QKT`2bbBQM K2i?Q/b /2KQM
KQ/2`M M2m` H M2irQ`Fb rBi? KBHHBQMb- Q` 2p2M #BHHBQ
Qp2`T ` K2i2`Bb2/- M/ i?2 7mM+iBQM H2 `M2/ #v H `;2 M2m`
`2T`2b2Mi2/ rBi? 72r2` KQ/2H T ` K2i2`bX

5IF -PUUFSZ 5JDLFU)ZQPUIFTJT

:Bp2M i?2 H `;2 MmK#2` Q7 b22KBM;Hv bmT2`~mQmb T ` K2i2
M im` H [m2biBQM i?2M `Bb2b, q?v + M r2 MQi i` BM i?2b2 b
7`QK b+` i+?\ 6` MFH2 kM/i?2`Q#BMb2(i? i T`mMBM; i2+?MB[m2
bm#M2irQ`Fb i? ir2`2 BMBiB HBb2/ iQ #2 i` BM2/ 2{+B2MiHvX
hB+F2i >vTQi?2bBb 7i2` r?B+? i?2 T T2` Bb M K2/,

ó ` M/QKHv@BMBiB HBx2/- /2Mb2 M2m` H M2irQ`F +QMi BMb
bm+? i? ir?2M i` BM2/ BM BbQH iBQM Bi + M K i+? i?2 i2bi
M2irQ`F 7i2` i` BMBM; 7Q` iKQbi i?2 b k2kMmK#2` Q7 Bi2` iB

6` MFH2 M/ * `#BM b?Qri? i T`mMBM; /2Mb2 M2irQ`F M/ `2i
BM bT `b2 bm#M2irQ`F rBi? 2[m H T2`7Q`K M+2 iQ i?2 /2Mb2
MBM; iB+F2iôX h?2B` K2i?Q/ T`Qp2b i? i Bi Bb TQbbB#H2 iQ i
T2`7Q`K b r2HH b H `;2` M2irQ`F ;Bp2M i?2 `B;?i BMBiB HBb
>vTQi?2bBb Qz2`b M 2tTH M iBQM 7Q` r?v i?2 H `;2 b? `2 Q7

#2 T`mM2/ r v 7i2` i` BMBM; rBi?Qmi BKT +iBM; T2`7Q`K M+
MmK#2` Q7 T ` K2i2`b Q7 i?2 M2irQ`F #27Q`2 i` BMBM; H2 /b i
KQ/2H ++m` +vX 6`QK i?2 ?vTQi?2bBb Bi 7QHHQrb- i? i bBM+
bm#M2irQ`F rBHH H2 `M #2ii2` i? M Qi?2`b- BM+Hm/BM; KQ`2
rBHH BM+`2 b2 i?2 HBF2HB?QQ/ Q7 }M/BM; órBMMBM; iB+F2

JAJP M2m` H M2irQ`Fb 2tTHQBi i?2 Qp2`T ` K2i2`Bb iBQM Q7
iQ `2 HBb2 KmHiBTH2- BM/2T2M/2Mi órBMMBM; iB+F2ibô rBi
T`mMBM; i?2 `2/mM/ Mi r2B;?ib- i?2v `2 mb2/ iQ +im HBb2 Km

%FSJWBUJWF NFUIPET PG .*.0

h?2 Q`B;BM H JAJP T T2` #vR>Np7Q+Bm2b2bHXQM BK ;2 +H bbB}+ i
#mi i?2 JAJP +QM};m` iBQM ? b #22M TTHB2/ iQ KmHiBTH2
}2H/ Q7 /22T H2 `MBM;X "8 KMiMQ2m+H2X (JAJP l@L2i 7Q` KQM
/2Ti? 2biBK iBQMX *v8;j2`i/ 2Tii iH?X2 (JAJP +QM};m` iBQM iQ i?2 6
_@*LL KQ/2H 7Q` `Q#mbi Q#D2+i /2i2+iB8QMm-iBMH/BSbB2ii?Q2TJQAJpJ2Pi
+QM};m` iBQM 7Q` j. Q#D2+i /2i2+iBQM 7Q` GB. _ TQBMi +HQ

JmHiBTH2 KQ/B}+ iBQMb iQ i?2 JAJP +QM};m` iBQM ? p2 Hb
2i H8X8)(BMi`Q/m+2 JBtJQ i? i +QK#BM2b JAJP rBi? KBt2/b KT
i iBQM- r?2`2 `iB}+B H b KTH2b `2 +`2 i2/ #v KBtBM; M/ +QK
62`B M+ 28)eH/X b( 2 `Hv@2tBib iQ JAJP M2m` H M2irQ`Fb iQ T`
T`2/B+iBQMb 7Q` i?2 b K2 BMTmi i/Bz2`2Mi /2Ti?b Q7 i?2 M2i
/Bp2`b2 2Mb2K#H2X h?2b2 K2i?Q/b `2TQ`i bmT2`BQ` ++m` +v
+QKT `2/ iQ JAJPX

4VNNBSZ

bM2m` H M2irQ`Fb ? p2 #2+QK2 H `;2` M/ KQ`2 ++m` i2-i?2B
? p2 #2+QK2 rQ`b2X SQbi@?Q+ K2i?Q/b HBF2 >BbiQ;` K "BM
b+ HBM; `2b+ H2 i?2 mM+2`i BMiv 2biBK i2b Q7 i` BM2/ M2m`
+? M;2 i?2 M2m` H M2irQ`F #27Q`2 i` BMBM;- bm+? b 2Mb2K#
iQ BKT`Qp2 + HB#` iBQMX J2i?Q/b bm+? b i?2 p `B Mib Q7 TT
M2irQ`Fb M/ p `BQmb 2Mb2K#H2 K2i?Q/b BKT`Qp2 + HB#` iBQ
bBx2 M/ +QKTmi iBQM H BMi2MbBivX hQ +QmMi2` +i i?Bb /`
+QmMi2`T `ib Q7 i?2b2 K2i?Q/b ? p2 #22M /2p2HQT2/- bm+? b
r?B+? T`2b2`p2b i?2 #2M2}ib Q7 KQ/2H p2` ;BM; rBi?Qmi BM
M2irQ`F T ` K2i2`bX

$)"15&3

5IFPSFUJDBM DP

"MFBUPSJD BOE FQJTUFNJD VODFSUBJO

AM i?Bb rQ`F- r2 /BbiBM;mBb? #2ir22M irQ ivT2b Q7 mM+2`i B
H2 iQ`B+ M/ 2TBbi2KB+ mM+2`i BMivX

h?2 H2 iQ`B+ mM+2`i BMiv Bb HbQ + HH2/ / i mM+2`i BMiv
BM?2`2Mi ` M/QKM2bb U2X;K2 bm82/K2hM?i2M2QTBbi2VKBM+/miM(+2`i
Bb /m2 iQ i?2 B;MQ` M+2 Q7 i?2 KQ/2H- BX2X T ii2`Mb M/ 7
i?2 KQ/2H ? b MQi H2 `M2/ #2+ mb2 Bi ? b MQi #22M i` BM2/
mM+2`i BMiv Bb B``2/m+B#H2 r?BH2 2TBbi2KB+ mM+2`i BMiv

#BZFTJBO NBDIJOF MFBSOJOH

J +?BM2 H2 `MBM; Bb /Bb+BTHBM2 i? i BMpQHp2b i` BMBM;
T`2/B+iBQMb QM M2r- mM b2 2M / i X /G2 i Qi2 i` BMBM; b2i Q7
b KTH2b- x?B`2 i?2 BMTmi M/ i?2 i `;2iX M2m` H M2irQ`F B
KQ/2H p(y|w;x) i? i i F2b M BMTmi M/ +iQKTmi2b T`2/B+iBQMb r2B;?i
T ` K2i2`b h?2 r2B;?i7 i?2 M2m` H M2irQ`F `2 i` BM2/ mbBM; /
i` BMBM; (R9R)X

6Q` K +?BM2 H2 `MBM; KQ/2H r2 i` BM?2 r2B;?ib i` BMBM;2b2i
K v 2tBbi b2p2` H b2ib Q7 KQ/2H T ` K2i2`b i? i `2 +QMbBbi2Mi BM
BHHmbi` i2/ QM 6B;m`2 bbB+ H K +?BM2 H2 `MBM;- i?2 BK
Q7 r2B;?ib i? i `2bmHib BM i?2 #2bi }i ;Bp2M i?2 / i X AM " v2
b2iiBM;- i?2 Q#D2+iBp2 Bb BMbi2 / iQ H2 `M /Bbi`B#miBQM Q
w 7Q` 2 +? r2B;?i- # b2/ QM i?2 / i X

6B;m`2 jXSR`X2/B+iBQ50M/bB2Q27`2MiHv BMBiB HBb2/ M2m` H M2irQ`Fb i` B
/ i X 1 +? M2m` H M2irQ`F H2 `Mb /Bz2`2Mi b2iQ7 T ` K2i2`b-`2bmI
T`2/B+iBQMb BM i?2 `2;BQMb rBi? MQ i` BMBM;/ i p BH #H2X

1PJOU FTUJNBUFT PG NPEFM QBSBNFUFST

h?2 r2B;?iBM M2m` H M2irQ`F `2 2biBK i2/ mbBM;K tBKmK H
iBQM UJG1V,

$$w^{JG1} = \arg\max_{w} \log p(Djw)$$

h?2`2 2tBbib MQ M HviB+ H bQHmiBQM /m2 iQ i?2 b?22` Mm
M2irQ`FbX AMbi2 /-i?2 K tBKmK wBF2HbQ#iQB/b2QHmbiBQM;` /B
/2b+2MeX(

_2;mH `Bb iBQM+ M #2 BMi`Q/m+2/ #v TH +wBM?B+T?B2Qb`mHTiQM
BM i?2 K tBKmK TQbi2`BQ`B UJ SV bQHmiBQM,

$$w^{JS} = \arg\max_{w} \log p(wjD)$$
$$= \arg\max_{w} \log p(Djw) + \log p(w)$$

+QKKQM +?QB+2 Q7 T`BQ` Bb i?2 x2`Q@K2 M: mbbB M T`BQ
H `Bb iBeQXM (

#BZFTJBO JOGFSFODF

G2 `MBM; T`Q# #BHBiv/Bbi`B#miBQM Qp2`i?2 r2B;?iT ` K2i
2biBK i2 Bb mb27mH 7Q`[m MiB7vBM;i?2 mM+2`i BMiv bbQ+
i2`bX

h?2 TQbi2`BQ` /Bbi`B#miBQM Q7 i?2 r2B;?ib #2 BM72``2/ mbBM
" v2bö `mH2- r?B+? KQ/2Hb i?2 `2H iBQM #2ir22M i?2 T`BQ` /Bb
/Bbi`B#miBQM 7i2` Q#b2`pBM; / i - + HH2/ i?2 TQbi2`BQ` /Bb

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

r?2`2 $p(w|D)$ Bb i?2 TQbi2`BQ` i? i 2tT`2bb2b i?2 /Bbi`B#miBQM Q
Q#b2`p2/ $p(D|w)$ Bb i?2 HBF2HB?QQ/ i? i 2tT`2bb2b ?Qr T`Q# #
/ i Bb 7Q` b2i Q7 T $p(w)$ K2i2`b2-T`BQ` i? i BM+Q`TQ` i2b T`BQ`
i?2 T ` K2i2`b #27Q`2 Q#b2`pBM; / i 2 M/ $p(D)$ Bb i?2 2pB/2M+2 M/ +ib b b+
7 +iQ` iQ 2Mbm`2 i? i i?2 TQbi2`BQ` /Bbi`B#miBQM Bb p HB/ /Bbi`B#miBQM-
iQ BM72` i?2 TQbi2`BQ` /Bbi`B#miBQM Bb + HH2/ 2t +i " v2bB

6Q` K +?BM2 H2 `MBM; TTHB+ iBQMb- Bi Bb Q7i2M KQ`2 +QMp
`mH2 M/ BMbi2 / KQ/2H i?2 HQ;@TQbi2`BQ` b,

$$\log p(w|D) = \frac{\log p(D|w) \log p(w)}{\log p(D)}$$

$$= \log p(D|w) + \log p(w) \quad \log p(D)$$

h?2 2pB/2M+2 + M #2 2p Hm i2/ #v K `;BM HBbBM; Qp2` HH T

$$p(D) = \int p(D|w)p(w)\,dw$$

>Qr2p2`- 7Q` KQ/2Hb rBi? K Mv T ` K2i2`b i?2`2 2tBbib MQ
i?2 BMi2;` H iQ 2p Hm i2 i?2 TQbi2`BQ`- r2 i?2`27Q`2 M22/ iQ
K2i?Q/bX

1PTUFSJPS BQQSPYJNBUJPO

h?2 irQ KQbi TQTmH ` K2i?Q/b 7Q` TT`QtBK i2 " v2bB M BM7
BM72`2M+2 M/ J `FQp +? BM JQMi2 * `HQ UJ*J*VX AM i?Bb i?
p `B iBQM H BM72`2M+2- b Bi ? b i?2 /p Mi ;2 Q7 QTiBKBbBM
iBQM M/ #2BM; 7 bi2` ` M;BMX KQbi + b2b (

7BSJBUJPOBM JOGFSFODF

o `B iBQM H BM72`2M+2 Bb M BM72`2M+2 K2i?Q/ 7Q` TT`Qt
#BHBiv /Bbi`B#miBQM $p(w|D)$ mbBM; MQi?2` /Bbi`B#miBQM $q(w)$ iQ i?2
p `B iBQM H TQ X KBHv

h?2 p`BiBQMQHB7bKBHv bh2iv Q7 TQbbB#H2 TT`QtBK i2 /Bbi`B#mi #v i?2 p`BiBQM H T, ` K2i2`b

$$Q = \{q(w|\ ) : \ \ 2 \ \ g$$

r?2`2 Bb i?2 b2i Q7 TQbbB#H2 p`BiBQM H T `K2i2`bX +QKKQM +?QB+2 Q p`BiBQM H 7 KBHv Bb i?2 K2 M@}2H/ : mbbB M 7 KBHv r?2 r2B;?ib `2 BM/2T2M/2Mi (

$$q(w|\ ) = \prod_j N(w_j | \ _j; \ _j^2)$$

rBi? p`BiBQM H T=`(K2i2`b$_j$; $_j$)X

o`BiBQM H BM72`2M+2 Bb MQTiBKBb iBQM@# b2/ K2i?Q/ r BKBb2 i?2 /Bp2`;2M+2 #2ir22M i?2 $p(w|D)$ Q M/ $q(w|\ )$ B#mi BQMb. mTiBM; i?2 QTiBK H p Hm2b Q7 i?2 p`BiBQM (8N@e N)X H T` K2i2`b

,VMMCBDL -FJCMFS EJWFSHFODF

+QKKQMHv mb2/ /Bp2`;2M+2 K2 bm`2 Bb i?2 EmHH# +F@G2B+ M #2 /2}M2/ b M BMi2;` H Q` M 2tT2+i iBQM X 6Q` $q(w|\ )$ M/ i?2 /Bb $p(w|D)$ - i?2 EG /Bp2`;2M+2 BbkQM q i?2 7Q`K (

$$KL[q(w|\ )||p(w|D)] = \int q(w|\ )\log\frac{q(w|\ )}{p(w|D)}\,dw = E_q\left[\log\frac{q(w|\ )}{p(w|D)}\right] \qquad \text{UjXRV}$$

h?2 EG /Bp2`;2M+2 #2 BMi2`T`2i2/ b K2 bm`2 Q7 /BbbBKBH #miBQMb- bQ i?2 QTiBK H p`BiBQM H T` K2i2`b + M #2 2biB 2tT`2bbBQM BM jXRm iBQM U

$$= \arg\min \ KL[q(w|\ )||p(w|D)]$$

6m`i?2`KQ`2- Bi TQbb2bb2b i?`22 KBKTQ`i Mi T`QT2`iB2b (

RX h?2 EG /Bp2`;2M+2 Bb MQM@bvKK2i`B+- K2 MBM;,

$$KL[q(w|\ )||p(w|D)] \neq KL[p(w|D)||q(w|\ )]$$

kX h?2 EG /Bp2`;2M+2 b iBb}2b i?2 T`QT2`iv Q7 MQM@M2; iB

$$KL[q(w|\ )||p(w|D)] \geq 0$$

jX6Q` i?2 EG /Bp2`;2M+2 $KL[q(w|\ )||p(w|D)] = 0$ B7 M/ QMHv B7,

$$q(w|\ ) = p(w|D)$$

q2 + M 2tT M/ QM i?2 2tT2+i iBQXN R BQM Q2#fm iBM QM U

$$E \, G[q(w j \ ) jj p(w jD)] = E \ \log \frac{q(w j \ )}{p(w jD)}$$

$$= E_q[\log q(w j \ )] \quad E_q[\log p(w jD)] \qquad \text{UjXkV}$$

$$= E_q[\log q(w j \ )] \quad E_q \ \log \frac{p(D; w)}{p(D)}$$

$$= E_q[\log q(w j \ )] \quad E_q[\log p(D; w)] + E_q[\log p(D)]$$

$$= E_q[\log q(w j \ )] \quad E_q[\log p(D; w)] + \log p(D)$$

AM i?2 H bi log p(D) T Bb BM/2T2MM/2M i -Q7 Q i?2 2tT2 $E_q[\log p(D)]$ M bBKTHv # 2+QK2b log p(D) Xb _2 `` M; BM; i?2 2tT`2bbBQM `2bmHib BM

$$\log p(D) = E_q[\log p(D; w)] \quad E_q[\log q(w j \ )] + E \, G[q(w j \ ) jj p(w jD)]$$

$$= L[q] + E \, G[q(w j \ ) jj p(w jD)] \qquad \text{UjXjV}$$

$$L[q]$$

r?2`2 $L[q] = E_q[\log p(D; w)] \quad E_q[\log q(w j \ )]$ Bb + HH2/ i?2 2pB/2M+2 HQr2` U1G"PV- #2+ mb2 Bi +ib b HQr2` # QmM/ iQ` log p(D) XQ` i?2 2pB/2M+2

aBM+2 log p(D) Bb +QMbi Mi [q(w j)] E G[p(w jD)] 0- K tBKBbBM; i?2 1G"P BI 2[mBp H2Mi iQ i?2 Q`B;BM H Q#D2+iBp2 Q7 KBMBKBbBM; i?2 E 1G"P Bb T`272``2/- bBM+2 Bi QMHv `2[mB`2b 2p p(D; w) iM M; i?2 D(p `B iBQM H q(w j ) i2`KbQ MQi i?2 2p p(D) X M+2

$)"15&3

.FUIPET

#BZFT CZ #BDLQSPQ #BZFTJBO OFVSBM

h?2" v2b #v" +FT`QT H;Q`Bi?K T`QTQb2/ #v"HmM/2HH 2i H
+QKT iB#H2 H;Q`Bi?K 7Q` i` BMBM;" v2bB M M2m` H M2irQ
r2B;?iBMi?2 M2irQ`F Bb `2T`2b2Mi2/ #v T`Q# #BHBiv/Bbi`B#
p Hm2b- b BHHmbi`9iXRQM2 T;mQ`# #BHBiv/Bbi`B#miBQM Qp2`
BM72``2/ mbBM; p `B iBQM H BM72`2M+2- bBM+2 2t +i" v2b
Q7 M2m` H M2irQ`F Bb BM72 bB#H2 rBi?i?2 KQmMiQ7 T`
M2irQ`RyX(

6B;m`2 9XRMXi?2 H27iBb 6mHHv@+QMM2+i2/MM2m``QHMbBMQ?2 nBMTm
M/?B//2M H v2`X 1 +? r2B;?iBMi?2 M2irQ`F Bb H2 `M2/-}t2/ p H
" v2bB M 7mHHv@+QMM2+i2/mM2m``QMbBQM F?2BMTmi M/?B//2M H v
r2B;?iBb `2T`2b2Mi2/ #v H2 `M2/T`Q# #BHBiv/Bbi`B#miBQMX

h?2" v2b #v# +FT`QT H;Q`Bi?K Bb HbQ TTHB+ #H2iQ +QMp
U*LLbV- r?B+? `2 TQTmH ` 7Q` T`Q9+d2XbbBML, BTKTH2B/2b (}Hi2`
iQ MBK;2-+`2 iBM; 72 im`2K TQ7i?2 BK;2X h?Bb HHQr
BK;2 72 im`2b Qp2`i?2eRXQH2 BK;2(

_2;mH ` *LLb `2 T`QM2iQ Qp2`}iiBM;9Q-Hb2K/BMM;/iQbQpb2`(+QM
/2Mi T`2/B+iBQMb M/mM`2HB #H2emXM+2v2bBMMv2mB`KHi2Mb2(rQ

`2 BM?2`2MiHv `Q#mbi ; BMbi Qp2`}iiBM;- r?B+? KQiBp i2b
+QMpQHmiBQM H M2m` H M2irQ`FbX h?2 }Hi2` p Hm2b BM "
`2 `2T`2b2Mi2/ mbBM; T`Q# #BHBiv /Bbi`B#miBQMb Qp2` TQb
}t2/ p Hm2bX h?Bb Bb BHHmbi` i2/ QM }; m`2 9XRXR

6B;m`2 9XRXRX i?2 H27i Bb *QMpQHmiBQM }Hi2` rBi? }t2/ p Hm2b BM
H v2`X PM i?2 `B;?i Bb *QMpQHmiBQM }Hi2` r?2`2 i?2 }Hi2` p Hm2
T`Q# #BHBiv /Bbi`B#miBQMX AM " v2bB M +QMpQHmiBQM H H v2` 2
iBM72`2M+2X

-PTT GVODUJPO

h?2 HQbb 7mM+iBQM Q7 i?2 " v2b #v # +FT`QT "LL + M #2 /2`
EG@/Bp2`;2M+2 BjMX V2 pmH BB;?Q MvU,

$$
\begin{aligned}
\mathrm{EG}[q(w|\ )\|p(w|D)] &= E_q[\log q(w|\ )] - E_q[\log p(w|D)] \\
&= E_q[\log q(w|\ )] - E_q[\log p(D|w)] - E_q[\log p(w)] + E_q[\log p(D)] \\
&= \mathrm{EG}[q(w|\ )\|p(w)] - E_q[\log p(D|w)] + \log p(D)
\end{aligned}
$$

r?2`2 EG[q(w|\ )||p(w)] Bb i?2 EG@/Bp2`;2M+2 #2ir22M i?2 TT`QtBK
$q(w|\ )$ M/ i?2 T`BQ` p(w) r?B+? +ib b `2;mH `Bb2` i? i T`2p2Mib i?2
TQbi2`BQ` 7`QK /2pB iBM; iQQ K m+?X 7`QK i?2 T`BQ` (

h?2 QTiBK H p `B iBQM H T ` K2i2`b KBMBKBb2 i?2 EG@/Bp2`;2M+2
$q(w|\ )$ M/ $p(w|D)$ `2 i?2M,

$$
\begin{aligned}
&= \arg\min \mathrm{EG}[q(w|\ )\|p(w|D)] \\
&= \arg\min \mathrm{EG}[q(w|\ )\|p(w)] - E_q[\log p(D|w)] + \log p(D) \\
&= \arg\min \mathrm{EG}[q(w|\ )\|p(w)] - E_q[\log p(D|w)] \qquad \text{U9XRV}
\end{aligned}
$$

h?2 2tT`2bbBQM BM 2[m iBQM U i?2 K `;BM H HBF2HB?QQ/ p(D) BM i?2 QTiBKBb iBQM
M22/ iQ #2 2p Hm i2/ /m`BM; i?2 QTiBKBb iBQM T`Q+2bb- #2+

i?2 p `B iBQM-Q rM? BH+? H2 /b iQ i?2 7QHHQrBM; HQbb 7mM+iBQM,

$$\begin{aligned} L(D; \;) &= \; \mathrm{E\,G}[q(w|\;)||p(w)] \quad E_q[\log p(D|w)] \\ &= E_q[\log q(w|\;)] \quad E_q[\log p(w)] \quad E_q[\log p(D|w)] \\ &= \quad L[q] \end{aligned}$$

LQiB+2 i? i i?2 HQbb 7mM+iBQM Bb +im HHv i?2 M2; iBp2 1G
b2+iBQM jXjXk KBMBKBbBM; i?2 EG@/Bp2`;2M+2 Bb 2[mBp H2Mi iQ
Q` BM i?Bb + b2- KBMBKBbBM; i?2 M2; iBp2 1G"PX

bbmKBM; i?2 i` BMBM; /`i BMiQ KBMB@#i+?2b- i?2 HQbb 7Q` KB
$b = 1, \dots, B$ + M #2 7Q`KmH i2/ b,

$$L(D_b; \;) = \;_b(E_q[\log q(w|\;)] \quad E_q[\log p(w)]) \quad E_q[\log p(D_b|w)] \qquad \text{U9XkV}$$

r?2`2 $_b$ 2 [0; 1] Bb r2B;?i 7 +iQ` 7Q` i?2 EG@/Bp2`;2M+2 i2`K 7Q` i?
r2 rBHH /Bb+mbb BM /2iBX 9XkXk BM b2+iBQM

h?2 2tT2+i iBQMb BM i?2 HQbb 9XkXk + M #2 [mT TiBQM BK i2/ MmK2`B+ H
JQMi2 * `HQ b KTHBM; (

$$L(D_b; \;) \approx \frac{1}{S}\sum_{s=1}^{S} \;_b \; \log q(w^{(s)}|\;) \quad \log p(w^{(s)}) \quad \log p(D_b|w^{(s)}) \qquad \text{U9XjV}$$

r?2`2 $w^{(s)}$ Bb i?2 $s$i? JQMi2 * `HQ b KTH2 /` rM 7`QK i?2 p `B iBQM
TQbi2`BQ` $q(w|\;)$X .m`BM; i` BMBM; M/ p HB/ iBQM- bBM;H2 JQM
/` rMX .` rBM; K Mv b KTH2b Bb +QKTmi iBQM HHv 2tT2MbBp2
b KTH2 r b bm{+B2Mi 7Q` i?2 "LLöb iQ H2 `MX

AM i?2 7QHHQrBM; b2+iBQMb- i?2 T`BQ` $\log p(w)$ M/ HBF2HB?QQ/ $\log p(D|w)$ i2`Kb- ;BpBM; 7Q` p `B iBQM H T`BQ` M/ p `B iBQM H T Q 2`BQ` $q(w|\;)$ BMi2`KbQ7 i?2 [m MiBiB2b 2tTHB+BiHv

M/ p `B iBQM H HQg $q(w|\;)$ bi2 `/BQ +mbb2/ KQ`2 BM@/2Ti?X

-PH QSJPS EJTUSJCVUJPO

h?2 T`BQ` i? i Bb T`QTQb2/ #v R2H/?HM/ 2i KX (bH `2 T`BQ` Q
: mbbB M /Bbi`B#miBQMb rBi? x2`Q K2 MX >Qr2p2`- r2 Dmbi
: mbbB M T`BQ`X

$$p(w) = \prod_j N(w_j | 0; \;^2) \qquad \text{U9X9V}$$

r?2`2 $w_j$ `2 i?2 r2B;?ibjB?MH?v2` M/ $^2$ Bb i?2 p `B M+2 Q7 i?2 : mb
/Bbi`B#miBQMX h?2 : mbbB M T`BQ` QMHv `2[mB`2b imMBM;
imMBM; irQ p `B M+2b- THmb i?2 KBtim`2 r2B;?i BM i?2 b+ H2 K
+?QQbBM; : mbbB M T`BQ` K2 Mb i? i i?2 `2;mH `Bb iBQM Q7
Gk `2;mH `Bb iBQM- b /2bj+`B#2/ BM b2+iBQM jXkXk

h?2 HQ;@T`BQ` Bb Q#i BM2/ #v HQ;@i` Mb 7Q`K, BM; i?2 T`BQ`

$$\log p(w) = \log \left(\prod_j N(w_j | 0, \sigma_1^2) \right) = \sum_j \log N(w_j | 0, \sigma_1^2) \qquad \text{U9X8V}$$

-PH MJLFMJIPPE GVODUJPOT

AM i?Bb b2+iBQM- r2 T`2b2Mi i?2 HBF2HB?QQ/ 7mM+iBQM 7Q` ;Bp2 M BMimBiBp2 mM/2`bi M/BM; Q7 2 +? HBF2HB?QQ/ 7mM+ bB M HQ;@HBF2HB?QQ/ 7Q` `2;`2bbBQM i bFb- 7QHHQr2/ #v i i bFbX

h?2 HBF2HB?QQ/ Q7 / i ;Bp2 M $p(D|w)$; +i T #2K2i2 `bii2M b +QM/ iBQM H Q7 i?2 i `;2i ;Bp2M BMTmib M/ r2B;?ib (

$$p(D|w) = \prod_{i=1}^{N} p(y_i | x_i; w)$$

: mbbB M HQ;@HBF2HB?QQ/ 7Q` `2;`2bbBQM

6Q` `2;`2bbBQM T`Q#H2Kb- i?2 MQBb2 Bb bbmK2/ iQ #2 : mbb : mbbB M HBF2HB?QQ/ 7mM+iBQM,

$$p(D|w) = \prod_{i=1}^{N} p(y_i | x_i; w) = \prod_i^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2\sigma_i^2} (y_i - \mu_i)^2 \right)$$

r?2`2 $\mu_i = \mu_i(x_i | w)$ M/ $\sigma_i = \sigma_i(x_i | w)$ `2 M2m` H M2irQ`F T`2/B+iBQMb Q7 M/ bi M/ `/ /2pB iBQM 7Q` BMTmi $x_i$ M/ iBKm?b;2 ?Mv$b_i$ 2R Bb i?2 i `;2iX h?2 K2 M Bb i?2 T`2/B+i2/ pyHm2 7Q? 2i b`;2i M/ `/ /2pB iBQM Bb mM+2`i BMiv Q7 Bib T`2/B+iBQMX

h?2 HQ;@HBF2HB?QQ/ Bb QM i?2 7Q`K,

$$\log p(Djw) = \log \prod_{i}^{N} \frac{1}{\sqrt{2\ \ _i^2}} \exp\left(\ \frac{1}{2\ _i^2}(y_i \ \ _i)^2 \right)$$

$$= \sum_{i}^{N} \log \frac{1}{\sqrt{2\ \ _i^2}} \exp\left(\ \frac{1}{2\ _i^2}(y_i \ \ _i)^2 \right)$$

$$= \sum_{i}^{N} \log(1) \ \frac{1}{2}\log(2\ ) \ \frac{1}{2}\log(\ _i^2) \ \frac{1}{2\ _i^2}(y_i \ \ _i)^2$$

$$= \sum_{i}^{N} \ \frac{1}{2}\left(\log(\ _i^2) + \frac{(y_i \ \ _i)^2}{\ _i^2} \right) + C_1$$

h?2 M2m` H M2irQ`F T`2/B+ib i?2 K2 M $_i$ M/ i?2 p `B M+2 $_i^2$ 7Q` 2 +? BMTmi $x_i$ Bb K tBKBb2/ b i?2 M2m` H M2irQ`F H #miBQM XW QA`Mbi2 / Q7 K tBKBbBM; i?2 HQ;@HBF2HB?QQ/- r2 HQ;@HBF2HB?QQ/,

$$\log p(Djw) = \ \frac{1}{2}\sum_{i=1}^{N} \left(\log(\ _i^2) + \frac{(y_i \ \ _i)^2}{\ _i^2} \right) \qquad U9XeV$$

h?2 : mbbB M HQ;@HBF2HB?QQ/ +QMbBbib Q7 irQ i2`Kb i? i 2 T`2/B+iBQMb rBi? M TT`QT`B i2 mM+2`i BMivX h?2 b2+QM/ 2``Q` $(y_i \ \ _i)^2$ ; BMbi i?2 T`2/B+i2/ mM+2`i BMiv $_i^2$X h?2 }`bi i2`K ?+M2 #2 KBMBKBb2/ #v KBMBKBbBM; i?2 b[m `2/ 2``Q` - Q` K tBKBbBM; i?2 }`bi i2`K TmMBb?2b i?2 : mbbB M HQ;@HBF2HB?QQ/ B7 i? H `;2X

GQ;@HBF2HB?QQ/ 7Q` +H bbB}+ iBQM

6Q` +H bbB}+ iBQM T`Q#H2Kb r?2`2 i?2 Hi#2H7Q` 2 +? BMTmi Bb $y_i \in f1;:::;Kg$- r2 mb2 + i2;Q`B+ H /Bbi`B#miBQM b HBF2HB?QQ/

$$p(Djw) = \prod_{i=1}^{N} p(y_i jx_i;w)$$

$$= \prod_{i}^{N} \ast\ i(y_i j\ \ (z(x_i jw)))$$

$$= \prod_{i}^{N} \ast\ i(y_i jp(x_i))$$

$$= \prod_{i}^{N} \prod_{c}^{C} p_c(x_i)^{I[y_i = c]}$$

r?2`2 $p(x_i) = [p_1(x_i)\; p_2(x_i)\; \dots\; p_C(x_i)] = \sigma(z(x_i|w))$ Bb p2+iQ` Q7 bQ7iK T`Q# #BHBiB2b bbB;M2/ iQ 2 + ? +H bb Q7 i?2 BMTmi x_i X h?2 bQ7iK t 7mM+iBQM

$$\sigma(z) = \left[\frac{\exp\{z_1\}}{\sum_c^C \exp\{z_c\}} \; \dots \; \frac{\exp\{z_C\}}{\sum_c^C \exp\{z_c\}} \right]$$

i F2b i?2 QmiTmi 7`QK i?2 H bi H v2` Q7 i?2 M2m` H M2irQ`F Bi BMiQ T`Q# #BHBiv /Bbi`B#miBQMX q2 HBF2rBb2 + M 2tT`2bb i?2 i`m2 H #2H y_i Bb QM2 ; iQ *H bb i?2 7Q`K,

$$\log p(D|w) = \log \prod_i^N \prod_c^C p_c(x_i)^{\mathbb{I}[y_i = c]}$$

$$= \sum_i^N \sum_c^C \mathbb{I}[y_i = c] \log p_c(x_i)$$

AiBb BKTHB+Bi i? i i?2 T`2/B+i2/ HQ;@bQ7iK t T`Q# #BHBiB2b w- bQ i?2 HQ;@HBF2HB?QQ/ Bb K tBKBb2/ b i?2 M2m` H M2irQ`F /Bbi`B#mi2b $q(w|\theta)$ i? i i?2 ;`QmM/ i`mi? +H bb Bb bbB;M2/ bQ7iK t T`Q# #BHBivX _ i?2` i? M K tBKBbBM; i?2 HQ;@HBF2HB?QQ/ KBMBKBb2 i?2 M2; iBp2 HQ;@HBF2HB?QQ/,

$$\log p(D|w) = \sum_i^N \sum_c^C \mathbb{I}[y_i = c] \log p_c(x_i) \qquad \text{U9Xd V}$$

h?Bb 2tT`2bbBQM Bb HbQ + HH2/ i?2 +`Qbb@2Mi`QTv HQbb 7mM+iBQM BM 9Xd[m iBQM i?2 HQ;@bQ7iK t T`Q# #BHBiv Q7 i?2 +Q``2+i +H bb Bb K tBKBb2/X

7BSJBUJPOBM QPTUFSJPS EJTUSJCVUJPO

6Q` i?2 p `B iBQM H TQbi2`BQ` /Bbi`B#miBQM $q(w|\theta)$ p `B iBQM H 7 KBHv rBi? p `B iBQM H T `K2i2`b $\theta$ : mbbB M /Bbi`B#miBQM + MMQi #2 M2; iBp2X r2 T `K2i2`Bb2

$$\sigma = \log(1 + \exp(\rho))$$

h?2 p `B iBQM H TQbi2`BQ` /Bbi`B#miBQM Bb 7mHHv 7+iQ`Bx2/ Qp2` i?2 p `B i
T ` K2i2`b (;) Bb ;Bp2M b,

$$q(w_j) = \prod_j N(w_j; \mu_j, \sigma_j) \quad \text{; aQ7iTHmb} \qquad \text{U9X3V}$$

h?2 p `B iBQM H HQ;@TQbi2`BQ` Bb Q#i BM2/ #v HQ;@i` Mb7Q

$$\log q(w_j) = \log \left(\prod_j \frac{1}{\sqrt{2 \pi \sigma_j^2}} \exp\left(-\frac{(w_j - \mu_j)^2}{2 \sigma_j^2} \right) \right) \qquad \text{U9XNV}$$

8FJHIUJOH PG ,- EJWFSHFODF UFSN

h?2`2 2tBbib b2p2` H r vb Q7 +?QQbBM; i?2 [m MiBiv, ... i?2 HQ;@HBF2HB?QQ/ i2`K Bb +QKTmi2/ 7Q` KBMB# i+? Q7 i 7Q` r2B;?iBM; i?2 EG@/Bp2`;2M+2 i2`K Bb rBi? i?2 MmK#2` Q7 bBM:` p2b 2N, HX (

$$\pi_b = \frac{n_b}{N} \qquad \text{U9XRyV}$$

r?2`2 $n_b$ Bb i?2 # i+? bBx2 Q7 KBMB# i+? i?2 iQi H b KTH2b BM i?2 i` b2DX

>Qr2p2`- r2 7QmM/ i? i i?2 r2B;?iBM; b+?2K2 Q7 :` p2b 2i HX T`QM2 iQ Qp2`}iiBM;- b pBbm HBb2/ BM TT2M/Bt X AMbi2 /- b+?2K2 T`QTQb2/ #v "HmM/2HH 2i HX 7Q` bHB?KQMM 2#Ti+?2K2 i?2 r2B;?iBM; π_b Bb +QKTmi2/ b 7QHHQrb,

$$\pi_b = \frac{2^{B-b}}{2^B - 1} \qquad \text{U9XRRV}$$

r?2`2 B Bb i?2 MmK#2` Q7 # i+?2b ... M /2[m i2Hv H `;2 MmK#2` Q7 # i+?2bX- "HmM/2HH 2i HX `; i? i i?2 bi `i Q7 2 +? 2TQ+? r?2M i?2 KQ/2H ? b MQi Q#b2`p EG@/Bp2`;2M+2 i2`K /QKBM i2b i?2 HQbb- BM T`2p2Mib i?2 bi` v iQQ 7 ` r v 7`QK i?2 T`BQ`X >Qr2p2`- 7Q` i?2 H ii2` # i+ Q#b2`p2/ KQ`2 /0 -M/ i?2 r2B;?iBM; H2ib i?2 HBF2HB?QQ/ i2`K HQbb 7mM+iBQMX

0QUJNJTBUJPO BMHPSJUIN

.m`BM; i` BMBM;- i?2 p `B iBQMH T `KQ2i2`b ... T`QT ; iBQM iQ K F2 i?2 p `B iBQM H TQbi2`BQ` q(w) i?2 i`m2 TQbi2`

p(wjD) X h?2 p `B iBQM H T ` K2i2`b `2 mT/ i2/ #v QTiBKBbBM;
2[m iBQXMbBM; # +FT`QT ; iBQM M/R eXB2Mi/2b+2Mi (

h?2 r2B;?i T ` K2i2`b 7Q`i?2 "LL `2Q#i BM2/ #v i FBM; b KTH
TQbi2`BQ` mbBM; i?2 `2T ` K2i2`b 9XBQMbi2+FQ7 b KTHBM;
r2B;?ib/B`2+iHv 7`QK i?2 p `B iBQM H TQbi2`BQ`

$$w \quad q(wj) = N(\ ; aQ7iTH^2)mb$$

r2+ M `2T ` K2i2`Bb2 i?2 r2B;?ib b,

$$w = t(\ ;) = \quad + aQ7iTH)mb$$

r?2`2Bb MQBb2 pN2(0;I) X EBM;K 2ei9 HXQ ri? i#v `2T ` K2i2`BbB
w = t(;) mbBM; M mtBHB `v M-QT2b2 pM/BK#H2bb Bb b2T ` i2/ 7`C
p `B iBQM H T ` K2i2`b i? ir2 QTiBKBb2 r?B+? H2 /b iQ H2bb
h?2 ;` /B2Mi Q7 i?2 HQbb 7mM9XBQMBQK2T[mi2BQMBUM; i?2 T
/2`Bp iBp2b rBi? `2bT2+iiQ 2 +? Q7 i?2 p `B/iBQbM/2b+`B#K2/i2`b
BM T`QTQbBiBQM R BM TT2M/Bt *X

h?2 " v2b #v # +FT`QT QTiBKBb iBQM H;Q`Bi?K 7Q` i` BMBM;

H;Q`Bi?KPRiBKBb iBQM H;Q`Bi?K 7Q` " v2b #v " RBT`QT- / T

_2[mB`2; ; 2R
L(D_B;) 0
7Q2 +? # i/+Q?
O a KTH2 r2B;?i
N (0;I)
aQ7iTH)mb
w +
(;)

O *QKTmi2 HQbb
$L(D_B;) \quad _b(\log q(wj) \quad \log P(w)) \quad P(Djw)$

O *QKTmi2 ;` /B2Mib M/ mT/ i2
$= \frac{@L(D_B;)}{@v} + \frac{@L(D_B;)}{@}$. *QKTmi2 ;` /B2Mib r`iX
$= \frac{@L(D_B;)}{@v} \frac{}{1+\exp(\quad)} + \frac{@L(D_B;)}{@}$. *QKTmi2 ;` /B2Mib r`iX
. lT/ i2
. lT/ i2

2M/ 7Q`

*OJUJBMJTBUJPO PG #BZFTJBO MBZFST

AMQm` 2tT2`B2M+2- +Q``2(+i)Hv) BTM`BK2iH2`bBbBQM`i2B;?ib M/ #B BM i?2 " v2bB M HBM2 ` M/ +QMpQHmiBQM H H v2`b Bb BKTQ M2irQ`Föb #BHBiv iQ H2 `MX AM i?Bb b2+iBQM- r2 /2b+`B#2 ?Q T ` K2i2`b BM i?2 " v2bB M HBM2 ` M/ +QMpQHmiBQM H H v2

*OJUJBMJTBUJPO PG

i }`bi- r2 i`B2/ iQ BMBiB HBb2 i?2 HBM2 ` M/ +QMpQHmiBQM /Bbi`B#miBQM,

$$U (6; 5)$$

>Qr2p2`- 2p2M 7i2` 2tT2`BK2MiBM; rBi? KmHiBTH2 +QK#BM iB #QmM/ 7Q` i?2 mMB7Q`K /Bbi`B#miBQM- i?2 " v2bB M M2m` H H v2`b r2`2 mM #H2 iQ H2 `M Mvi?BM;X

q2 }t2/ i?Bb T`Q#H2K #v BMBiB HBbBM; i?2 " v2bB M HBM2 +QMpQHmiBQM H H v2`b BM i?2 b K2 r v b i?2 ``2 M;2K2Mi Q7 Hv2`b BM SvhQ`+?- mbBM; E BKBM; mMB7Q`K BMBiB HBb iBQM (

$$U \quad p\frac{1}{k}; p\frac{1}{k}$$

r?2`2 k Bb b+ H ` p Hm2 i? i p `B2b 7Q` HBM2 ` M/ +QMpQHmiB H v2` rBi? BMTmi 72 im`2b +QMpQHmiBQM H H v2` rBi? BMTmi - M/ F2`M2H bBx2 p Hm2b 7Q`

$$k_{HBM2} = \hat{f}_{BMTmi}$$
$$k_{+QMp} = f_{BMTmi} s^2$$

*OJUJBMJTBUJPO PG

h?2 T ` K2i2`b BM i?2 " v2bB M HBM2 ` M/ +QMpQHmiBQM iB HBb2/ mbBM; i?2 mMB7Q`K /Bbi`B#miBQM,

$$U (6; 5)$$

q2 2tT2`BK2Mi2/ rBi? b2p2` H ` M;2b M/ 7QmM/ i? i i?2 BMBiB K ii2` b Km+?- mMHBF2 i?2 BMBiB HBb iBQM Q7 2bmHiBM; bi /2pB iBQM Bb bm{+B2MiHv bK HH- i?2 M2irQ`F Bb #H2 iQ H2 `

#// PVUQVU

" v2bB M M2m` H M2irQ`Fb H2 `M TQbi2`BQ` /Bbi`B#miBQM M/ r2 + M i?2 2`27Q`2 Q#i BM KmHiBTH2 /Bz2`2Mi KQ/2Hb #v b `BQ`X b /2b+`B#29 BM X2+iBQM 2 QMHv QM+2 /m`BM; i` BMBM; r2 + M b KTH2ib Q7 KQ/2H r2B;?ib M/ ;;`2; i2 i?2B` T`2/B+iBQMb im`2 T`2/B+iBQM Q7 2 +QKTQM2MibX h?2 Mm K#2` Q7 KQ/2H 2tT2`BK2Mi HHv M/ Bb /2b9 X3#Xk/ BM b2+iBQM

AM i?2 7QHHQrBM; irQ b2+iBQMb- r2 /2KQMbi` i2 ?Qr i?2 KBti bBQM M/ +H bbB}+ iBQM i bFb `2 +QKTmi2/X

3FHSFTTJPO PVUQVU

AM i?2 `2;`2bbBQM b2iiBM;- i?2 " v2bB M M2m` H M2irQ`F QmiT M/ Bib mM+2`i BMivX a KTH2b Q7 KQ/2H r2B;?ib M/ ;;`2; i?2 T`2/B+iBQMb Q7 2 +?2`2bmHib BM : mbbB M KBtim`2 rBi? i?2 M/ p `B M+2,

$$= \frac{1}{S}\sum_{s}^{S} (x j w^{(s)}) \quad \text{U9XRkV}$$

$$^2 = \frac{1}{S}\sum_{s}^{S} (x j w^{(s)})^2 + (x j w^{(s)})^2 \quad \text{U9XRjV}$$

r?2`2 $w^{(s)}$ q(w j ) Bb b KTH2 7`QK i?2 p `B iBQM H TQbi2`BQ`X h Q7 i?2 KBtim`2 K2 M M/ p `B M+2 + M #2 7QmM/ BM TT2M/Bt

$MBTTJaDBUJPO PVUQVU

AM i?2 +H bbB}+ iBQM b2iiBM;- i?2 " v2bB M M2m` H M2irQ`F #BHBiv /Bbi`B#miBQM Q7 +H bb2b 7Q` #BM `v Q# #BHBiB2b

$$\log p(x) = \log p_1(x) \; ::: \; \log p_C(x)$$

"v b KTH2b 7`QK i?2 p `B iBQM H TQbi2`BQ` + M #2 Q#i BM2/ 7QH HQ;@T`Q# #BHBiv /Bbi`B#miBQM $\log p(x) = \log p(x j w^{(s)})$ i? i i?2 HQ;@/Bbi`B#miBQMb `2 7`QK /Bz2`2Mi KQ/2HbX h? /Bbi`B#miBQM Bb Q#i BM2/ mbBM; i?2 GQ;J2 M1tT QT2` iBQM

$$\log \hat{p}(x) = \log \left[\frac{1}{S}\sum_{s=1}^{S} \exp \log p \; x j w^{(s)} \right] \quad \text{U9XR9V}$$

r?2`2 w$^{(s)}$ Bb i?2 b KTH2 7`QK i?2 p `B iBQM H$^{(s)}$ q(T|Q)X i?2 `2BQb -Bb ?2M+2 7Q`i? `272``2/ iQ bó p2` ;BM; Qp2` KQ/2Hb rBi?b KTH2 Q7 +H bbB}+ iBQMX

h?2 T`2/B+i2/ H #2H # b2/ QM i?2 KBtim`2 /Bbi`B#miBQM Bb i?

$$\hat{y} = \arg\max_c \log \hat{p}(x)$$

.*.0 OFVSBM OFUXPSLT

h?2 KmHiB@BMTmi KmHiB@QmiTmi UJAJPV +QM};m`2/ M2m` Biv rBi?BM bBM;H2 M2irQ`F iQ +im HBb2 KmHiBTH2 BM/2T2 i` BMBM;- i?2 bm#M2irQ`Fb `2 ;Bp2M /Bz2`2Mi BMTmib bm+? i i?2 T`2/B+iBp2 i bF BM/2T2M/2MiHvX i BM72`2M+2- i?2 bm#M BMTmi M/ i?2 T`2/B+iBQMb Q7 2 +? bm#M2irQ`F `2 ;;`2; i2/ /B+iBQMX (h?Bb Bb BHHmbi9 X}; h?2 JAJP +QM};m` iBQM + M #2 K2Mi2/ #v KQ/B7vBM; i?2 2tBbiBM; `+?Bi2+im`2 Q7 M2m` H M

Ç JQ/B7v i?2 BMTmi HMv2` iQ BMTmi F BKmHi M2QmbHv

Ç JQ/B7v i?2 QmiTmi HMv2` iQ Tr2/B+iBQM ?2 /b

q?2`2 i?2 ?vT2`T M K2i2`Qi2b i?2 MmK#2` Q7 bm#M2irQ`Fb BM M2irQ`FX

U Vh` BMBM; U#VAM72`2M+2

6B;m`2 9XJAJP +QM};m`2/ +H bbB}+ iBQM M2irQ`F /m`BM; i` BMBM; BM; i` BMBM;- i?2 M2irQ`F `2+2Bp2b /Bz2`2Mi BK ;2b 7Q` 2 +? bm#M2 ;Bp2b T`2/B+iBQMb 7Q` Bib QrM BMTmiX .m`BM; BM72`2M+2 i?2 b K2 BK ;2- M/ i?2 QmiTmi T`Q# #BHBiB2b Q7 i?2 bm#M2irQ`Fb }M H T`2/B+iBQMX / Ti2/ 7`QK (RN) p bB 2i H (

AM i?2 M2ti b2+iBQMb- r2 T`2b2Mi i?2 bT2+B}+ KQ/B}+ iBQMb M/ QmiTmi i? i `2 M2+2bb `v 7Q` i?2 JAJP +QM};m`2/ M2irQ` `2;`2bbBQM M/ +H bbB}+ iBQM i bFbX

.*.0 JOQVU

h?2 BMTmi Q7 i?2 JAJP M2irQ`F Bb F2v T `i Q7 2Mbm`BM; i? i +im HBb2 /m`BM; i` BMBM;X A7 `2;mH ` /2i2`KBMBbiB+ M2m BMTmi JAJP +QM};m`2/ M2irQ`F i?2 BMTmib Q7 i?2 `2;mH rQ`F x1;::::;xM gX .m`BM; i` BMBM;- i?2 BMTmib `2 BM/2T2M/2MiH / i b2i #v KmHiBTHvBM; i?2 rB;?i b BM Q7 i?2 M2m`QMb #2ir22M i?2 BMTmib M2irQ`F i?2M +QM+ i2M i2/X rBi? i?2 MmK#2` Q7 bm#M2irQ`Fb `2[mB`2b i? i i?2`2 Bb M 2[m H KQmMi Q7 / i 7Q` 2 +? bm#M b KTH2b `2 /Bb+ `/2/ B7 i?2 M2m`QM / H2 bBx2 Q7 i?2 2TQ+?X i BM72`2M+2- i?2 b K2 M2irQ`F Bb /mTHB+ i2/

3FHSFTTJPO JOQVU

6Q` i?2 `2;`2bbBQM i b-F Q7 i?2 i #mH ` / i Bb bi +F2/ HQM /BK2MbBQM iQ ;Bp2 i?2 BMTmi BM bBx2 R Q7 `2;`2bbBQM + b2- 6Q` /BK2MbBQM H / i - i?2 BMTmi /BK2MbBQM Bb i?2 MmK#2` Q7 `2;`2bbBQM 72 im`2b BM i?2 / i X hQ ++QKK 72 im`2b Q7 i?2 }`bi HBM2 ` H v2` BM i?2 M2irQ`F Bb KmHiBTH

$MBTTJaDBUJPO JOQVU

6Q` i?2 +H bbB}+ iBQM i b- i?2 BMTmi Bb KTH2/ M/ +QM+ i2M i2/ HQM M2Hb /BK2MbBQM- `2bmHiBM; BM BMTmi b?`2 /Bb i` i2/ QM }9XX9 h?Bb BMTmi Bb i?2M T bb2/ iQ i?2 BMTmi +QMpQ Bb /Dmbi2/ iQ i?2 MmK#2` Q7 +QHQm`b +? M Bb i?2 MmK#2` Q7 bm#M2irQ`FbX

6B;m`2 9XRX Hmbi` iBQM Q7 ?Qr BK ;2b `2 +QM+ i2M i2/ #27Q`2 #2BM +QMpQHmiBQM H H v2` BM i?2 +H bbB}+ iBQM KQ/2HX h?2 BMTmi BK +? MM2Hb /BK2MbBQMX

.*.0 MPTT GVODUJPO

h?2 JAJP M2m` H M2irQ`F Bb i` BM2/ mbBM; i?2 bmK Q7 i?2 M
ULGGV 7Q` 2 +M? lQm7 #iM2irQ`FbX h?2 HQbbD7bQB`b +KQBKMTBm#i2/+?b
7QHHQrb,

$$L(D_b; w) = \sum_m \log p(D_{b;m} | w)$$

r?2`l2gp(D_{b;m}|w) Bb i?2 LGG 7Q` bm#QM2iKQMFBb#Xi+?

.m`BM; p HB/ iBQM M/ BM72`2M+2-r2+QKTmi2 i?2 LGG mbBM
h?Bb`2bmHib BM i?2 7QHHQrBM; HQbb,

$$L(D_b; w) = \log p(D_{b;m} | w)$$

r?2`p2(D_{b;m}|w) Bb i?2 LGG 7Q` i?2 2Mb2K#H2 T`2/B+iBQM-r?B+? Bb
7Q` i?2 `2;`2bbBQM M/ +H bbB}+ iBQM i bFbX

3FHSFTTJPO /--

6Q` i?2 `2;`2bbBQM i bF-r2+QKTmi2 i?2 :mbXB7Q LG?2 BM 2
KBtim`2K2 M/ KBtim`2p `BQMM2`2bm#M2irQ`Fb- b/2`Bp2/BM TT
",

$$\log p(D_b|w) = \frac{1}{2}\sum_{i=1}^{N_b} \log(\sigma^2_{;i}) + \frac{(y_i - \mu_{;i})^2}{\sigma^2_{;i}}$$

r?2`2 r?2`;i2 M/ ;i `2 KBtim`2 QmiTmMb Qm#Mi2irQ`Fbi7Q` i?2
BMTmiBM i?2 #i+?X

$MBTTJaDBUJPO /--

6Q` i?2 +H bbB}+ iBQM i bF-r2+QKTmi2 2XM2vL2GbB2M#2[BMiBiQ2
HQ;@T`Q# #BHBiv/Bbi`B#miBQMb 7Q` 2 +?bm#M2irQ`F mbBM

$$\log p(D_b|w) = \sum_i \sum_c I[y_i = c] \log \left[\frac{1}{M} \sum_{m=1}^{M} \exp(\log p_{m;c}(x)) \right]$$

r?2`y2Bb i?2 2Mb2K#H2 T`p2(p_{m;c}(x)|w) BQMi?2/HQ; T`Q# #BHBiv 7Q` bm
m 7Q` +HXbb

.*.0 PVUQVU

h?2 QmiTmib Q7 JAJP M2m` H M2irQ`Fb /Bz2` 7`QK M Q`/BM `v ? b KmHiBTH2 T`2/B+iBQM ?2 /bX .m`BM; i` BMBM;- i?2 QmiTm >Qr2p2`- /m`BM; p HB/ iBQM M/ i BM72`2M+2- i?2 T`2/B+iB ;;`2; i2/ BMiQ M 2Mb2K#H2 T`2/B+iBQMX

AM i?2 7QHHQrBM; irQ b2+iBQMb- r2 T`2b2Mi ?Qr i?2 2Mb2K# 7Q` i?2 `2;`2bbBQM M/ +H bbB}+ iBQM i bFbX

3FHSFTTJPO PVUQVU

AM i?2 `2;`2bbBQM b2iiBM;- M2m` H M2irQ`F T`2/B+i $_1$ K:2: $M_M$ M/ bi M/ `/ /2pB iBQM $_1$ ::: $_M$ X h?2 T`2/B+iBQMb `2 ;;`2; i2/ : mbbB M KBtim`2 KQ/2H- r?2`2 +QKTQM2Mib- b /2b+`B#2/ BM TT2M

$$= \frac{1}{M} \sum_{m}^{M} \quad m$$

$$= \frac{1}{M} \sum_{m}^{M} (\;^2_m + \;^2_m) \qquad ^2$$

$MBTTJaDBUJPO PVUQVU

AM i?2 +H bbB}+ iBQM b2iiBM;- M2m` H M2irQ`F QmiTmi + i2;Q`B+ #BHBiv /Bbi`B#miBQM Q7 i?2 KBtim`2 Q7 +QKTQM2Mib; #BHBiv /Bbi`B#miBQM Bb @ BM- #BHBiB2bX

$$\log p(x) = \begin{bmatrix} \log p_1(x) \\ \vdots \\ \log p_M(x) \end{bmatrix} \begin{bmatrix} \log p_{1;1}(x) & ::: & \log p_{1;C}(x) \\ \vdots & \ddots & \vdots \\ \log p_{M;1}(x) & ::: & \log p_{M;C} \end{bmatrix}$$

h?2 2Mb2K#H2 HQ;@T`Q# #BHBiv /Bbi`B#miBQM Bb ; BM Q#i QT2` iBQM,

$$\log \hat{p}(x) = \log \left[\frac{1}{M} \sum_{m=1}^{M} \exp(\log p_m(x)) \right]$$

h?2 T`2/B+i2/ H #2H # b2/ QM i?2 KBtim`2 /Bbi`B#miBQM Bb i?

$$\hat{y} = \arg\max_c \log \hat{p}(x)$$

/BJWF NVMUJIFBEFE NPEFMT

> p bB 2iRJ/2 KQMbi` i2 i? iB7 i?2 BMTmi/ i Bb MQiBM/2T2M/
bm#M2irQ`Fb QmiTmi i?2 b K2 T`2/B+iBQMb- `2/m+BM; i?2 /B
2Mb2K#H2X > p bB 2i HX + HH i?Bb i?2 M Bp2 KmHiB?2 /2/ K
M/ ;Bp2b KmHiBTH2 QmiTmib- b9BX8 Hmbi` i2/ QM };m`2

6B;m`2 9X8X bi`m+im`2 Q7 M Bp2 KmHiB?2 /2/ +QMpQHmiBQM H M2
BMTmiBK ;2 Bb ;Bp2MiQi?2 M2irQ`F- #mii?2`2 `2 KmHiBTH2 T`2/B+i
QmiTmibX

q2 +QKT `2 i?2 T2`7Q`K M+2 Q7 i?2 M Bp2 KmHiB?2 /2/ KQ/2H
M/ BMp2biB; i2 ?Qr i?2 H +F Q7 /Bp2`bBiv KQM; i?2 bm#M2i
i BMiv2biBK i2b M/ T`2/B+iBp2 T2`7Q`K M+2X h?2 M Bp2 KmH
mbBM; i?2 b K2 HQbb 7mM+iBQM b i?2 JAJP KQ/2HX

.*.#0 OFVSBM OFUXPSLT

h?2 JmHiB AMTmi JmHiB " v2bB M PmiTmi UJAJ"PV +QK#BM2
K2i?Q/b, H2 `MBM; i?2 T`Q# #BHBiv /Bbi`B#miBQMb Qp2` i?2
M2irQ`F 2Mb2K#H2b 7`QK JAJPX q?BH2 #Qi? K2i?Q/b BKT`Qp
M2m` H M2irQ`F i?`Qm;? KQ/2H p2` ;BM;- i?2 TT`Q +?2b +QK

h?2 JAJ"P M2m` H M2irQ`F Bb 2bb2MiB HHv JAJP +QM};m`2/
`2 bQK2 //BiBQM H +QMbB/2` iBQM iQ K F2 r?2M TTHvBM;
iQ "LL i? i `2 /2b+`B#2/ BM i?Bb b2+iBQMX

q?2M K FBM; 7Q`r `/T bb- "LL b KTH2b Bib r2B;?ib 7`QK i?2
iBQMbX iBM72`2M+2- KQ`2 i? M bBM;H2 b2i Q7 r2B;?ib `2
Bb K /2 7`QK i?2b2 b2ib Q7 r2B;?ib iQ }M/ i?2 KQ/2H T`2/B+iB
KQ/2H HbQ K F2b T`2/B+iBQMb mbBM; M2Mb2K#H2 Q7 bm#M

bm#M2irQ`F T`2/B+iBM; BM/2T2M/2MiHv #27Q`2 #2BM; 2Mb2K
aQ r?2M +QK#BMBM; i?2 irQ K2i?Q/b r2 +?QQb2 iQ b KTH2 i?2
rQ`Fb iQM+2- i?2M 2Mb2K#HBM; Qp2` i?2 bm#M2irQ`Fb 7Q` 2
2Mb2K#HBM; Qp2` i?2 r2B;?ib KTH2bX aBM+2 2 +? r2B;?ib KT
`2 2Mb2K#H2/ Qp2`- i?2 }M H T`2/B+iQ` Bb M 2Mb2K#H2 Q7 2

.*.#0 JOQVU

h?2 BMTmiQ7 i?2 JAJ"P KQ/2H Bb 2t +iHv i?2 b K2 b 7Q` JAJP
p HB/ iBQM M/ i2biBM9 X kaXQ2` b2 i+ iBHQbMX

.*.#0 MPTT GVODUJPO

h?2 HQbb 7mM+iBQM 7Q` JAJ"P M2irQ`Fb 7mM+iBQMb HBF2 i
M2m` H M2irQ`F- 2t+2Tii? iBi? biQ ++QmMi7Q`JAJ"Pöb bm#
i?2 EG@/Bp2`;2M+2 i2`K BM i?2 1G"P BM i?2 b K2 r v b 7Q`
M2; iBp2 HQ; HBF2HB?QQ/ M22/biQ #2 /Dmbi2/ iQ rQ`F rBi?
.m`BM; i` BMBM;- 2 +? bm#M2irQ`F M22/biQ #2 i` BM2/ BM/2
7 +BHBi i2 i?Bb- i?2 HQbb Q7 2 +? BM/BpB/m H bm#M2irQ`F B
HQbb,

$$L(D_b;\ ) \quad \frac{1}{S}\sum_{s=1}^{S} \quad _b \ \log q(w^{(s)}j\ ) \quad \log p(w^{(s)}) \ + \sum_{m=1}^{M} \log p(D_{b;m}jw^{(s)}) \quad U9XR8V$$

r?2`D2b;m Bb i?2 / i BM i?2 7#Qi+b?m#M2irQ`Fb w(s) Bb i?2 r2B;?i
b KTH2X

.m`BM; p HB/ iBQM- bBM+2 i?2 bm#M2irQ`Fb `2 T`2/B+iBM; QM
i?2 LGG 7`QK i?2 T`2/B+iBQMb Q7 i?2 2Mb2K#H2 Q7 bm#M2irQ

$$L(D_b;\ ) \quad \frac{1}{S}\sum_{s=1}^{S} \quad _b \ \log q(w^{(s)}j\ ) \quad \log p(w^{(s)}) \quad \log p(D_bjw^{(s)}) \quad U9XReV$$

r?2`2p(DbjW(s)) Bb i?2 M2; iBp2 HQ; HBF2HB?QQ/ Q7 i?2 2Mb2K#
bm#M2irQ`FbX h?2 2Mb2K#H2 LGG Bb +QKTmi2/ /Bz2`2MiHv 7
iBQM i bFbX

3FHSFTTJPO FOTFNCMF /--

6Q` i?2 `2;`2bbBQMi bFb- i?2 bm#M2irQ`FM/2bi+1MT``2//B2+piB k@2
iBQM?B+? M22/biQ #2 +QK#BM2/ BMM bKmB#iM2 2rQ`Fb- i?2 2/2b+`B#

BM TT2M/Bt"X q2 i?2M Q#i BM i?2 2Mb2K#H2 LGG,

$$\log p(D_b|w^{(s)}) = \frac{1}{2}\sum_{i=1}^{n_b}\left(\log(\sigma^2_{y,i}) + \frac{(y_i - \mu_{y,i})^2}{\sigma^2_{y,i}}\right) \qquad (9)$$

r?2`2 $\sigma^2_{y,i}$ M/ $\mu_{y,i}$ `2 KBtim`2 QmiTmib Qp2`i?2 bm#M2irQ`Fb M/ i?2 HQ;T`Q##BHBiB2b-r2mb2i?2GQ;J2M1tTQT2`iBQMX

$$p(D_b|w^{(s)}) = \sum_{i}^{n_b}\sum_{c}^{C} I[y_i = c]\log\left(\frac{1}{M}\sum_{m=1}^{M}\exp\left(\log p_{m,c}(x|w^{(s)})\right)\right) \qquad (10)$$

r?2`y2Bb i?2 2Mb2K#H2 T`Q/BxiwB(Q Mb M?/2 HQ;T`Q##BHBiv7Q`rQ`m 7Q`+HrBbi?2 i?2 r2B;?ib KTH2X

iBM72`2M+2-i?2 KQ/2H r2BS,?bK2`b-bQ(BHM2Q`/2`iQ+H+mHi2HQ;@T`Q##BHBiB2b r2i F2i?2 K2M LGG Qp2`i?2 r2B;?ib K

$$p(D_b|w) = \frac{1}{S}\sum_{s=1}^{S} p(D_b|w^{(s)})$$

.*.#0 0VUQVU

h?2 QmiTmiQ7 JAJ"P M2m`H M2irQ`F Bb M2 `Hv i?2 b K2 b M2irQ`Fb-rBi?i?2K DQ`/Bz2`2M+2i?iJAJ"PB,?Mb2K#H2bXQp i i` BMBM;-i?2 M2irQ`F QmiTmibi?2 BM/BpBm/H T`2/B+iBQ i?2 LGG Bb+H+mHi2/BM/2T2M/2MiHv7Q`2+?T`2/B+iBQM?QMHv i?2 2Mb2K#H2 T`2/B+iBQM Bb QmiTmiii2/X AM+QMi`bi /Bz2`2M+2#2ir22Mi?2 p HB/iBQM M/i2bi QmiTmiQ7 JAJ"JAJ"P b KTH2bib Q7 r2B;?ib $\sigma^{(s)}_{y,i}$ $\mu^{(s)}_{y,i}$ i?2 bi iBK2-i?2 2Mb2K#H2 QmiTmi Bb Qp2`#Qi?i?2 T`2/B+iBQM ?2/b M/r2B;?ib KTH2bX

3FHSFTTJPO PVUQVU

6Q``2;`2bbBQM T`2/B+iBQM Mb-i?2+QmKiBmM2/BMiQ KBtim`2 K Qp2`i?2 bm#M2irQ`Fb M/r2B;?ib KTH2bX h?2 KBtim`2 Bb+ K

TT2M/Bt "MBS?KBtim`2 +QKTQM2Mib,

$$= \frac{1}{M S} \sum_{m}^{M,S} m$$

$$^2 = \frac{1}{M S} \sum_{m}^{M,S} (m + {}^2_m) \quad ^2$$

$MBTTJaDBUJPO PVUQVU

6Q` +H bbB}+ iBQM T`2/B+iBQMb i?2 QmiTmi +QMbBbib Q7 H
+H bb2b- b r2HH b T`2/B+i2/ +H bb T`Q# #BHBiB2b `
Qp2` i?2 bm#M2irQ`Fb }`bi- i?2M Qp2` i?2 r2B;?ib KTH2bX "Q
`2 /QM2 mbBM; GQ;J2 M1tTX

$$\log \hat{p}(x|w) = \log \left(\frac{1}{S} \sum_{s=1}^{S} \exp \left(\log \frac{1}{M} \sum_{m=1}^{M} \exp \left(\log p_m (x|w^{(s)}) \right) \right) \right)$$

h?2 T`2/B+i2/ +H bb H #2H Bb i?2M

$$\hat{y} = \arg\max_c \log \hat{p}(x|w)$$

.PEFM BSDIJUFDUVSFT

AM i?Bb b2+iBQM r2 T`2b2Mi i?2 i?`22 KQ/2H `+?Bi2+im`2b r2

- Ç JmHiBH v2` T2`+2Ti`QM 7Q` `2;`2bbBQM
- Ç J2/BmK*LL 7Q` +H bbB}+ iBQM
- Ç qB/2 _2bL2i 7Q` +H bbB}+ iBQM

U \JmHiBH v2` T2`+2Ti`QM U#V2/BmK*LL

6B;m`2 9X2X };m`2 b?Qrb i?2 KQ/2H `+?Bi2+im`2b 7Q` i?2 KmHiBH v
BM i?2 `2;`2bbBQM i bFb UH27iV M/ i?2 J2/BmK*LL mb2/ BM i?2 +H
LQM@JAJP +QM};m`2/ M2irQ`Fb bm+? b bi M/M/=M2Xm` "HLLM2irQ`F ?
rBHH QMHv ? p2 bBMS,HQ2mBTmTmbi##2miQ`2 2Mb2K#HBM;X JAJP M2irQ`
b K Mv BMTmib M/ QmiTmib b bT2+B}2M rBM? J2A2 "P `BKH2HM? p2
BMTmib-#mi b K Mv QmiTmib bi?2 iBK2b iS-iHBF;?ib v2bBKTM2/m` I
M2irQ`FX h?2 MmK#2` BM i?2 7mHHv +QMM2+i2/ U6*V H v2`b BM/B+
72 im`2b Q7 i?2 H v2`-r?BH2 i?2 MmK#2` BM i?2 +QMpQHmiBQM H H
QmiTmi+? MM2HbX 1 +? H v2` Bb 7QHHQr2/ #v _2Gl +iBp iBQM 7m
Q7 i?2 }M H H v2`X h?2 +iBp iBQM 7mM+iBQMb `2 QKBii2/ 7`QK i?2

.VMUJMBZFS QFSDFQUSPO .-1

6Q` `2;`2bbBQM i bFb- r2 mb2 KmHiBH v2` T2`+2Ti`QM UJG
+QMM2+i2/ U6*V H v2`b rBi? _2Gl +iBp iBQM 7mM+iBQMb b B
h?2 `+?Bi2+im`2 r b /2b+`B#2/R#NX> p bB 2i H (

.FEJVN$//

6Q` BK ;2 +H bbB}+ iBQM r2 mb2 M `+?Bi2+im`2 # b2/ QM i?2
#v 6Q`i 2iR HX (b BHHmbi` i2/ QM #2HQr K*LL4?+ QMpQHmiBQM H
H v2`b 7QH HQr2/ #v @ +QMM2+i2/ H v2`b- rBi? i?2 QMM H 6* H v2
72 im`2b- i?2 b2 i?2 MmK#2` Q7 +H bb2b BM i?2 i M2K#2` Q7
bm#M2irQ`Fb BM i?2 KQ/2HX h?2 Tm`TQb2 Q7 i?Bb KQ/2H Bb i
b? HHQr M2irQ`F `+?Bi2+im`2X h?Bb M2irQ`F ? b i?2 + T +
r?BH2 biBHH #2BM; 7 bi iQ 2p Hm i2X

8JEF 3FT/FU

h?2 qB/2 _2bL2i Bb # b2/ QM i?2 TQTmH ` `2bB/m H M2irQ`F
BMi`Q/m+2/ #v >2 2i HX (_2bL2i Bb #mBHi mbBM; Bib M K2b F
H2 `MBM; 7` K2rQ`F- r?B+? miBHBb2b bFBT +QMM2+iBQMb i?
+QMpQHmiBQM H H v2`b Q7 i?2 M2irQ`FX db BM H `Q`; QB}M }
`QmM/ i?2 H v2`b Q7 i?2 #HQ+FX aBM+2 M2m` H M2irQ`Fb Q`/
+QKTHB+ i2/ 7mM+iBQMb M2irQ`Fb BMTmi r b //2/ iQ i?2 Qmi
i?2 QmiTmi Bb - BirQmH/ BMbi2 / H2 `M i?2F(x) = H(x) Hx M+iBQ
"v T bbBM; i?2 bB;M H i?`Qm;? i?2 bFBT +QMM2+iBQMb- i?2
iQ #2 T`2b2`p2/ i?`Qm;?Qmi i?2 M2irQ`F TBT2HBM2X h?2 qB/
#v w ;Q`mvFQ 2i HX- BKT`Qp2b QM i?2 Q`B;BM H _2bL2iX "v
Q7 72 im`2b BM i?2 +QMpQHmiBQM H v2`b- qB/2 _2bL2i `2 +

6B;m`2 9X#X bB+ _2bL2i #HQ+F 7`QK Qm` qB/2 _2bL2iX h?2 BMTmi bB
r2B;?i H v2`b M/ Bb //2/ iQ i?2 QmiTmi Q7 i?2 H v2`bX h?Bb K F2b
`2bB/m H 7mM+iBQMX

H2p2H Q7 H2bb /22T `2;mH ` _2bL2iX b `2bmHi- qB/2 _2bL
KQ`2 2{+B2MiH v (M `2;mH ` _2bL2ibX q2 mb2 qB/2 _2bL2i `+?
/22T2` Hi2`M iBp2 iQ i?2 J2/BmK*LL `+?Bi2+im`2- bQ i? iQm`
#Qi? b? HHQr M/ /22T `+?Bi2+im`2X qB/2 _2bL2i + M #2 +Q
bT2+B}+ iBQMb 7Q` rB/i? M/ /2Ti?X h?2 KQ/2Hb `2 M K2/QM i
L@Fô r?2`2 N2 Bb i?2 /2Ti? Q7 #HQ+F M/ F Bb i?2 rB/i? 7 +iQ`- b T
9XXR

:`QmT L K2	PmiTmi aBx2				"HQ+F hvT2
*QMp R	32 32		3 3; 16		
*QMp k	32 32	3	3; 16	k	N
		3	3; 16	k	
*QMp j	16 16	3	3; 32	k	N
		3	3; 32	k	
*QMp 9	8 8	3	3; 32	k	N
		3	3; 64	k	
p; TQQH	1 1		8 8		
GBM2 `	*		100		

h #H2 9XR `m+im`2 Q7 i?2 qB/2 _2bL2iX :`QmTb +QMpk- +QMpj M/ +Q
bB+ _2bL2i #HQ+Fb 9XXXRM}; m`2 QM T +QMp j M/ +QMp 9- /QrMb KTHB
BM i?2 }`bi +QMpQHmiBQM H H v2`X h?2 bFBT +QMM2+iBQM HbQ /Q
+QMpQHmiBQM H H v2`X h?2 }M H HBM2 ` H v2` QmiTmib +H bb b+Q`2b 7Q
6B;m`2 / Ti2/ 7`QK h #H2 R BM (

q2 mb2 i?2 bT2+B}+ +QM}; m` iBQM q2 mb2 Bi Qp2` HH vB2H
#2bi `2bmHib 7Q` qB/2 _2bL2ib QM #Qi? *A6 _@Ry M/ *A6 _@
w ;Q`vmFQ 2i HX >B;?2` ++m` +v rBHH #2ii2` b2`p2 i?2 Tm`T
2biBK iBQM- #2+ mb2 Bi Bb 2 bB2` iQ +QKT `2 Qm` `2bmHib iQ

7JTVBMJTBUJPOT BOE QMPUT

AM i?Bb b2+iBQM r2 2tTH BM bQK2 Q7 i?2 MQM@bi M/ `/ pBb
mb2 7Q` Qm` `2bmHibX q2 2tTH BM #Qi? ?Qr i?2v `2 K /2 M/ ?

3FMJBCJMJUZ %JBHSBNT

_2HB #BHBiv /B ;` Kb `2 mb27mH 7Q` pBbm HBbBM; ?Qr r2HH
2``Q` Q7 KQ/2H K i+?2b Bib mM+2`i BMivX AM `2HB #BHBiv
+m` +v `2 THQii2/ ; BMbi Bib mM+2`i BMiv 2biBK i2b QM i?Q

#2ir22M 2``Q`f ++m` +v M/ mM+2`i BMiv`2T`2b2Mib i?2 KQ/2H
/Bz2`2M+2b BM ?Qr`2;`2bbBQM M/ +H bbB}+ iBQM KQ/2Hb `2
;` Kb /Bz2`#2ir22M i?2 i bFb ?Qr2p2`i?2v b? `2 i?2 b K2 Qp2`
+ b2b i?2 `2HB #BHBiv /B ;` K Bb # `THQi- rBi? i?2 ?2B;?i Q
iQ i?2 K2 M 2``Q`f ++m` +v Q7 T`2/B+iBQMb K /2 rBi? M mM+2
` M;2 BM/B+ i2/ #v i?2 rB/i? Q7 i?2 # `X KQ/2Höb T102/B+iBQ
#BMbX h?2 ` M;2b Q7 i?2b2 #BMb /2T2M/b QM r?2i?2` i?2 KQ/2
`2;`2bbBQM Q` i?2 +H bbB}+ iBQM i bFX

U V U#V

6B;m`2 9 X3X KTH2b Q7 `2HB #BHBiv /B ;` Kb 7Q``2;`2bbBQM UH27iV
q?BH2 `2;`2bbBQM `2HB #BHiBv /B ;` Kb mb2b [m MiBH2 #BMb i? i B
i?2B`rB/i?- +H bbB}+ iBQM /B ;` Kb mb2 2[m HHvrB/2 #BMb-rBi? +Q
/2MbBiv BM #BMbX

3FHSFTTJPO

AM i?2 `2;`2bbBQM + b2 Qm`` `2HB #BHBiv /B ;` Kb `2 K /2 #v #
T`2/B+i2/ p `B M+2,

$$o\ (B_k) = \frac{1}{jB_kj} \sum_{i2B_k} {}^{2}_{i}$$

"2+ mb2 T`2/B+i2/ p `B M+2b + M p `v BM b+ H2 r2 mb2 [m MiBH
#BM +QMi $\frac{1}{10}$ BM QM ;i?2 T`2/B+iBQMbX 6m`i?2`KQ`2- #2+ mb2 Q7
p `B M+2b- r2 THQi i?2 `2HB #BHBiv /B ;` Kb rBi? HQ;@b+ H
BKT`Qp2bi?2`2 / #BHBiv Q7 i?2 THQi-/2bTBi2 K FBM;BMi2`T
b2i i?2 bi2T ?2B;?ib iQ i?2 K2 M b[m `2/ 2``Q` Q7 i?2 T`2/B+iB

$$Ja1(B_k) = \frac{1}{jB_kj} \sum_{i2B_k} (\ _i \quad y_i)^2$$

r?2`2 `2 i?2 T`2/B+i$y_i$ QM2bi?2 / `;2ib BM B_k iX2 q#2bM?Qr BM TT2M/Bt
. i? i i?2 QTiBK H p `B M+2 Bb 2[m H iQ i?2 Ja1X

GQr2` # `b K2 Mb H2bb 2``Q`- M/ +QM}/2Mi T`2/B+iBQMb `2 B
K2 MBM; i? i +QM}/2Mi T`2/B+iBQMb `2 QM i?2 H27i bB/2 Q7 i?
Bb /Qii2/ HBM2 BM/B+ iBM; i?2 x-2bBHH 7mMiH29 8MKK}Am`2
Mv# `b `2 HQr2`i? M i?2 HBM2 BM/B+ i2-Bi BM/B+ i2b i? i i?
BM i? i ` M;2- b Bi b?Qrb i? i T`2/B+iBQMb `2 +HQb2` iQ i?2 i
T`2/B+ibX *QMp2`b2Hv- B7 # `b `2 #Qp2 i?2 HBM2- Bi BM/B
`2HB #BHBiv /B ;` KX3MB6B;Mn 2i KTH2 Q7 M mM/2`+QM}/2Mi
KQ/2H- b KmHiBTH2 Q7 Bib # `b `2 #2HQr i?2 B/2 H HBM2X
pBbm HBb2/ BM `2/X

"v `2T2 iBM; i?2 i` BMBM; Q7 B/2MiB+ H KQ/2Hb rBi? /Bz2`2Mi
iQ ;2i T`2/B+iBQMb 7`QK KmHiBTH2 BMbi M+2b Q7 i?2 b K2 KQ
HHQrb mb iQ + H+mH i2 i?2 p `B M+2 Qp2` i?2B` T`2/B+iBQ
#BBM$_k$X aBM+2 i?2 #BMb `2 + H+mH i2/ 7`QK i?2 mM+2`i BMiv
2tT2+i i?2 #BMb iQ #2 B/2MiB+ H 7Q` /Bz2`2Mi KQ/2Hb-2p2M
BMBiB HBb2/X hQ + H+mH i25iK2Q+/2HK#BiM?2/#BMb 7QK /2 iQ +Q
$\frac{1}{10}$ Q7 i?2 T`2/B+iBQMb Q7 HH `2T2iBiBQMbX h?Bb r v-#BMb `2
Hi?Qm;? 2 +? #BM rBHH MQi M2Q+2/b2/ BHiB+QMi 7BQK 2 +? BM
`2T2iBiBQMX h?2 K2 M Qp2` i?2 `2T2iBiBQMb Bb i?2M i?2 pB
M/ i?2 bi M/ `//2pB iBQM Q7 i?2 T`2/B+iBQMb K2 bm`2 + M #2
#BM- HHQrBM; mb 95%Q+pQBMbm/2HMB2BMi2`p Hb 7Q` i?2 # `b ?2B;?

$MBTTJaDBUJPO

AM i?2 + b2 Q7 +H bbB}+ iBQM KQ/2H- i?2 KQ/2H +QM}/2M+
++m` +v Q7 Bib T`2/B+iBQMbX "2+ mb2 #Qi? +QM}/2M+2 M/
[0;1]- r2 m10b2#BMb Q7 2[m H rB/i? [0;1]X?2 ?BM;K2 Mb i? i b KTH2
r?2`2 i?2 KQ/2H T`2/B+i rBi? +QM}[0:9;1:0]2 BMbiQ?`2i2/MB;M2iQ i?2
`B;?iKQbi #BM- +QM}/2M[0:8;0:9]Bb i?2 M ;2}`bi #BM 7`QK i?2 `B;
bQ QMX

6Q` 2 +? #BM i?2 ac(B$_k$) `B+v H+mH i2/ b BM (

$$+(B_k) = \frac{1}{jB_kj} \sum_{i2B_k} I(\hat{y}_i = y_i)$$

r?2`2 jB$_k$j Bb i?2 KQmMi Q7 T`2/B+iBQMb BM B$_k$ M/ I(y$_i$ = $\hat{y}_i$) #BbM M BM/B+ iQ`
7mM+iBQM1i?B7 B?b2 T`2/B+iBQM `2+Q`/2/i?2 `rBb2X aBKBH `Hv-
p2` ;2 +QM}/2M+2 BM 2 +? #BM Bb + H+mH i2/ b

$$+Q(B_k) = \frac{1}{jB_kj} \sum_{i2B_k} \hat{p}_i$$

r?2`2 Bb i?2 +QM}/2M+2 K2i`B+ T`Q#+ B2QMX AM i?2 +H bbB}+ iB
mb2 i?2 bQ7iK t T`Q# #BHBiv b +QM}/2M+2 K2 bm`2X aBM+2
++Q`/BM; iQ i?2B` +QM}/2M+2b- r2 rQmH/ 2tT2+i i? i 7Q` T2`
i?2 p2` ;2 +QM}/2M+2 BM #BM rQmH/ 2[m H i?2 p2` ;2 ++m`

$$+QM7(B_k) = +(B_k)$$

M/ i?2 `2HB #BHBiv /B ;` K rQmH/ 7QHHQr i?2 B/2 H /B ;QM H
+ i2/ b /Qii2/ HBM2 X aBM+2 r2 /2pB iBQM 7`QK i?Bb BM/B+ i2b
iBQM- rBi? # `b #Qp2 i?Bb /B ;QM H BM/B+ iBM; mM/2`+QM}/2
i? M +QM}/2M+2 QM p2` ;2- M/ # `b mM/2` i?2 HBM2 BM/B+ i2
` v Bb HQr2` i? M +QM}/2M+2 QM p2` ;2X h?2 /2}+Bi BM # ` ?2
HBM2V Bb BM/B+ i2/ BM `2/X

"2+ mb2 i?2 #BMb ? p2 }t2/ rB/i?- r2 mb2 +QHQm` ;` /B2Mi iQ B
i?2 T`QTQ`iBQM Q7 b KTH2b BM #BM Qmi Q7 i?2 iQi H KQm
BM; i?2 b KTH2 /2MbBiv r2 + M b22 r? i #BMb z2+i K2i`B+b H
+ HB#` iBQM 2``Q` U2tTH BM2/ BM H i2` b2+iBQMV i?2 KQbi
7Q` 2 +? #BM- b 9e% B M;i?2 M+2 BMi2`p H Q7 i?2 ++m` +v Q7
`2T2iBiBQMbX AM THQib rBi? 2``Q` # `b i?2 +im H #BM ?2B;
Qp2` 5 `2T2iBiBQMbX

4VCOFUXPSL EJWFSTJUZ EJBHSBNT

> p bB 2iR HX (Qr i? i i?2 bm#M2irQ`Fb BM JAJP +QM};m`2/ M2
HBF2 BM/2T2M/2MiHv i` BM2/ M2m` H M2irQ`Fb #2+ mb2 2 +? b
BMTmib /m`BM; i` BMBM;- r?B+? H2 /b iQ /Bp2`bBiv KQM; bm#
M Bp2 KmHiB?2 /2/ KQ/2Hb `2 i` BM2/ mbBM; i?2 b K2 BMTmi
KQM; Bib bm#M2irQ`FbX

AMbTB`2/ #v > p bB 2i HX (i 2H X- 2 BMp2biB; i2 i?2 /Bz2`2M
/Bp2`bBiv Q7 i?2 bm#M2irQ`Fb BM i?2 JAJP- M Bp2 KmHiB?2 /
#v BMp2biB; iBM; i?2 QTiBKBb iBQM i` D2+iQ`v Q7 i?2 KQ/2H
/Q i?Bb #v +?2+FTQBMiBM; T`2/B+iBQMb QM i?2 p HB/ iBQM
+?2+FTQBMi b p2b i?2 BM/BpB/m H HQ;@bQ7iK t T`Q# #BHBi
b K2 bBM;H2 # i+?X h?2 }`bi +?2+FTQBMi Bb i F2M #27Q`2 i` B
2TQ+? /m`BM; i` BMBM;X h?Bb vB2H/b b2`B2b Q7 TQBMib i? i
T`2/B+i2/ i bi ;2b i?`Qm;?Qmi i` BMBM;X h?2 +?2+FTQBMib ?
M_bm#M2irQ`Fb_ 6 bKTH2b bbb2b VX

1 +? +?2+FTQBMi 7Q` 2 +?n bm#M2irQ`F Bb BQM H TQBMi r?B+?
/B{+mHi iQ BMi2`T`2i i?2 /Bz2`2M+2b Q7 i?2 bm#M2irQ`F T`2
QTiBKBb iBQM i` D2+iQ`v Q7 i?2 bm#M2irQ`Fb BM i?2 7mM+i
i?2 /BK2MbBQM HBiv Q7 i?2 +?2+FTQBMib mbBM; S* X q?BH2
K2i?Q/b 7Q` /BK2MbBQM HBiv `2/m+iBQM- bm+? b i@aL1 M

S* 7Q` Bib bBKTHB+Biv M/ #2+ mb2 Bi Bb /B`2+i T`QD2+iBQ
BM i?2 bT +2 T`2@T`QD2+iBQM K Tb iQ +QMbBbi2Mi TQBMi B

q2 +`2 i2 i?2 S* THQib rBi? i?2 7QHHQrBM; bi2Tb,

- Ç *QM+ i2M i2 rBi?M M/M rBi?M 7Q` 2 +? KQ/2HX
- Ç S2`7Q`K S* QM i?2 +QM+ i2M i2/ bQ7iK t QmiTmib 7Q` H irQ@/BK2MbBQM H bm#bT +2
- Ç SHQi i?2 +QM+ i2M i2/ bQ7iK t QmiTmib BM i?2 irQ@/BK2 i2/ #v S* iQ pBbm HBb2 ?Qr i?2 /Bz2`2Mi K2K#2`b BM i?2 +QMp2`;2 BM i?2 7mM+iBQM bT +2

1SJODJQBM $PNQPOFOU "OBMZTJT

S`BM+BT H *QKTQM2Mi M HvbBb US* V Bb /BK2MbBQM HB r?2M TTHB2/ iQ / i - H2 `Mb M2r # bBb i? i 2tTH BMb i?2 K p `B M+2 #2ir22M / i TQBMibX

h?2`2 `2 irQ K2i?Q/b 7Q` T2`7Q`KBM; S* X PM2 K2i?Q/ BMp p Hm2 /2+QKTQbBiBQM Uao.V iQ }M/ i?2 M2r # bBb Q7 i?2 S* K2i?Q/ }M/b i?2 2B;2Mp2+iQ`b Q7 i?2 / ie3X +qQ2mbE2 M+2bK2 iQBM/ K2i?Q/ bBM+2 r2 +QMbB/2` Bi bBKTH2` iQ BKTH2K2MiX

h?2 S* H;Q`Bi?K Bb K /2 mT Q7 i?2 7QHHQrBM; bi2Tb,

H;Q`Bi?KSkBM+BT H *QKTQM2Mi M HvbBb

$\bar{X} = X \frac{1}{N} \sum_{i=1}^{N} x_i$. am#i` +i i?2 K2 M

$X^{-} = \frac{1}{d} X X^{T}$. * H+mH i2 i?2 +Qp `B M+2 K

6BM/ 2B;2Mp=[H0; d] M/ 2B;2Mp2=[Q0;:::;vd] Q7 i?2 +Qp `B@ M+2 K i`Bt

aQ`i #v i?2B` +Q``2bTBQMQBM2` Q7 H `;2bi iQ bK HH2bi

$B = X^{T} V_n$ r?2`2 $V_n = [v_0;:::;v_n]$ M/ Bb MmK#2` Q7 mb2/.+QKTQ2M2Mib iQ @/BK2MbBQM H bT +2 `2imBM

S* `2im`Mb i?2 / i T`QD2@/BK2MbBQM H bT +2X "v Q`/2`BM p2+iQ`b Q7 i?2 +Qp `B M+2 K i`Bt #v i?2 bBx2 Q7 i?2 2B;2Mp 2B;2Mp2+iQ`b Q` +QKTQM2Mib i? i b # bBb rQmH/ 2tTH BM i }`biX IbBM; i?2b2 +QKTQM2Mib b # bBb rBHH mbm HHv `2br b i?2v T`QpB/2 +Hmbi2`BM; Q7 +Q``2H i2/ TQBMib b r2HH b TQBMibX 6Q` Qm` Tm`TQb2b- r2 ? p2 7Q2m/BK2MibBQMb; `2/mB i?2 }`bi M/ b2+QM/ +QKTQM2Mi b # bBb T`QpB/2b i?2 KQbi /

Measures of diversity

In addition to visualising the optimisation trajectory of the subnetworks in classification models, we compute the similarities between subnetworks as an empirical measure of the diversity. We let the output for each of the M subnetworks be a categorical distribution $P_i(\hat{y})$ for $i = 1, ..., M$, and consider two types of similarity measures, the first being the disagreement between the predicted classes:

$$D_{\text{Disagreement}}(P_1(\hat{y}), ..., P_M(\hat{y})) = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbf{I}(\arg\max_{\hat{y}} P_i(\hat{y}) \neq \arg\max_{\hat{y}} P_j(\hat{y})) \tag{4.19}$$

where $I(\cdot)$ is an indicator function that is 1 if the predicted class is same, and 0 otherwise.

Since we ensemble the classification models by averaging the output distributions, it therefore makes sense to measure the similarity between the output distributions of the individual subnetworks. The second similarity measure is the average KL-divergence of the output distributions [69]:

$$D_{\text{KL}}(P_1(\hat{y}), ..., P_M(\hat{y})) = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} \text{KL}(P_i(\hat{y}) || P_j(\hat{y})) \tag{4.20}$$

where $\text{KL}(\cdot||\cdot)$ is the KL-divergence.

4.6 Evaluation metrics

In the following section, we present the various metrics that are used to evaluate the predictive performance and uncertainty estimates of our models. The first two metrics, RMSE and accuracy, are used purely to assess the predictive performance of our models. The other metrics incorporate the predictive uncertainty and are therefore used as a measure for the calibration and quality of the uncertainty estimates [11].

4.6.1 Root mean squared error (RMSE)

The root mean squared error (RMSE) is an error measure to evaluate the predictive performance of regression models [14]. For a test set of N samples, the RMSE is evaluated as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\mu_i - y_i)^2}$$

where y_i is the ith target and μ_i is the ith prediction.

4.6.2 Accuracy

The predictive performance for our classification models are evaluated using accuracy. For a test set with N samples, the accuracy is given as:

$$\text{acc} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}[\hat{y}_i = y_i]$$

where $\mathbf{I}(\hat{y}_i = y_i)$ is an indicator function that is 1 if the prediction $\hat{y}_i$ is correct and 0 otherwise.

4.6.3 Brier score

The Brier score is a metric that incorporates both the accuracy and uncertainty estimates of a model. For a C class classification problem, the Brier score over a test set of size N is defined as

$$\text{BS} = \frac{1}{N}\frac{1}{C}\sum_{i=1}^{N}\sum_{c=1}^{C}\left(\mathbf{I}[y_i = c] - p_c(x_i)\right)^2$$

where $\mathbf{I}[y_i = c]$ is an indicator variable that is 1 if the predicted class is equal to the ground truth class and 0 otherwise, and $p_c(x_i)$ is the predicted softmax probability that the input x_i belongs to class c [14, 70].

4.6.4 Negative log-likelihood (NLL)

The negative log-likelihood (NLL) lets us aggregate the model accuracy and uncertainty estimates in a single scalar metric.

Regression

For the regression tasks, we use the Gaussian negative log likelihood (GNLL)

$$-\log p(\mathcal{D}|\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left(\log\left(\sigma_i^2\right) + \frac{\left(y_i - \mu_i\right)^2}{\sigma_i^2}\right) \tag{4.21}$$

which can be viewed as an extension of the mean squared error to also include the variance of the given prediction. A low GNLL indicates that the predictions of the model fit the test data well, so when comparing two models, the one with the lower score presents a better combination of prediction accuracy and uncertainty estimation.

The GNLL we use as a metric is slightly different from the one used as a loss function in that we take the mean rather than the sum over the data points. This returns a smaller number that we consider more readable. N is the number of samples in the test set, y_i is the ith target, and μ_i and σ_i are predictions of the mean and standard deviation of the ith test point.

Classification

The NLL is also useful as a performance metric for classification models. Here, we use the classification NLL:

$$-\log p(\mathcal{D}|\mathbf{w}) = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{C}\mathbf{I}[y_i = c]\log p_c(x_i)$$

The NLL punishes models that are confidently wrong, as such answers, where the correct label is assigned very low probabilities (close to 0), result in a large positive addition to the negative loss due to the logarithmic nature of the function. Therefore, NLL as a metric correlates positively with both of our desired traits: accuracy and precise uncertainty estimates. Notice, we once again take the mean over the N test points rather than the sum. y_i is the ith target and $\log p_c(x_i)$ is the log-softmax probability of class c given an input x_i.

4.6.5 Expected calibration error (ECE)

Reliability diagrams are good visual tools for showing miscalibration, but when comparing several models with each other, it can be difficult to gauge the differences in calibration from a collection of plots. We want a summary statistic that allows for easy comparison of the calibration of our models.

Classification

In the classification case, the expected calibration error (ECE) [11] is a weighted average of the difference between the confidence and accuracy in each bin B_k

$$\text{ECE} = \sum_{k=1}^{K}\frac{|B_k|}{N}|\text{acc}(B_k) - \text{conf}(B_k)|$$

where $|B_k|$ is the number of samples in bin B_k and N is the total number of samples in the test set. By using a weighted average, a few outliers will not give us an incorrect understanding of how well a model is calibrated.

Regression

The ECE can also be calculated in the regression case, where it is calculated similarly, replacing accuracy with mean squared error, and confidence with variance:

$$\mathrm{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{N} |\mathrm{MSE}(B_k) - \mathrm{Var}(B_k)| \tag{4.22}$$

Since we do binning by quantiles for regression, each bin will have the exact same number of samples, so it follows that $|B_1| = |B_2| = \ldots = |B_K|$. This means that each bin is weighted equally. From this it is clear that the choice of how the observations are binned is very important for the ECE, i.e. using quantile bins will give a very different result than using linear evenly spaced bins would. Our choice to use quantile-based bins for regression ECE prioritises consistency in binning with the reliability plots, for which quantile-based bins provide clear readability advantages.

4.7 Datasets

In this work, different datasets are used to test the presented methods on a variety of tasks. Different datasets are required for training on the regression and classification tasks, for which we explore the capabilities of Bayesian neural networks and subnetwork ensembles to make accurate predictions and uncertainty estimates.

4.7.1 Regression tasks

In the regression case we decided to generate a toy dataset, since it allows us full control of the function that we are trying to fit. It also allows us to know the exact aleatoric uncertainty in the dataset which in turn lets us better evaluate our models' uncertainty estimates. To try the models on a high-dimensional problem, we also create a multi-dimensional toy dataset. This lets the problem stay within our control while being more complex than the one-dimensional dataset. Finally, we use a regression dataset from UCI [71], Communities and Crime [72], to try our models on real world data. Using real world data allows us to better infer whether the various tested methods apply well to other real world regression problems.

One-dimensional toy dataset

The one-dimensional toy dataset is a simple synthetic dataset with data generated from the function

$$f(x) = x + 0.3\sin(2\pi(x+\epsilon)) + 0.3\sin(4\pi(x+\epsilon)) + \epsilon \tag{4.23}$$

where $\epsilon \sim \mathcal{N}(0, 0.02^2)$. The function is as given by Blundell et al. [16].

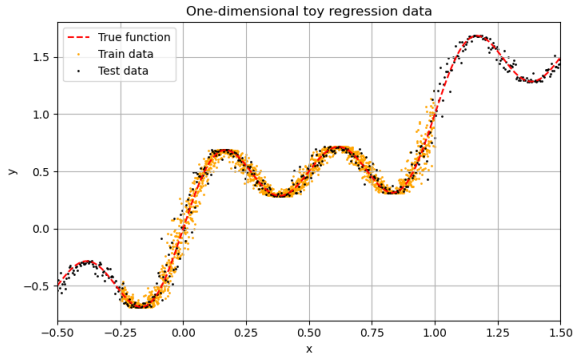


Figure 4.9. Plot of the one-dimensional regression data generated with the equation given by Blundell et al. [16]. The training data is generated in the range $[-0.25; 1.0]$ while the test data is generated in the range $[-0.5; 1.5]$

We generate $N_{\text{train}} = 2000$ training data-points, with a validation set of $N_{\text{val}} = 500$ and a test set of $N_{\text{test}} = 500$.

This dataset presents a one-dimensional regression task which gives a good baseline for testing the functionality of the different regression models. Because the test data is generated in a bigger range, some test data points will be out-of-distribution. This allows us the observe how the models trained on this data reacts when faced with data it should be uncertain about.

Multi-dimensional toy dataset

The multi-dimensional toy dataset is based on the same function used for the one-dimensional dataset. It is projected to 64 dimensions by using a random (1×64) vector with values sampled from a univariate standard Gaussian distribution. Additionally, we change the noise ϵ so that we have

$$\epsilon_x = \begin{cases} \sigma \quad \text{for} \quad x \in]-0.5; 0.5[\\ \sigma \cdot (1 + 4 \cdot (x - \frac{1}{2})) \quad \text{for} \quad x \in [0.5; 1.5] \end{cases}$$

where $\sigma = 0.02$. This function makes the noise increase linearly with x to 5 times its initial amount from $x = 0.5$. By making the noise heteroscedastic, we aim to make fitting the problem more challenging.

Because we expect the problem to be more complex, we generate $N_{\text{train}} = 20000$, $N_{\text{val}} = 5000$ and $N_{\text{test}} = 5000$ data points for the training set, validation set and test

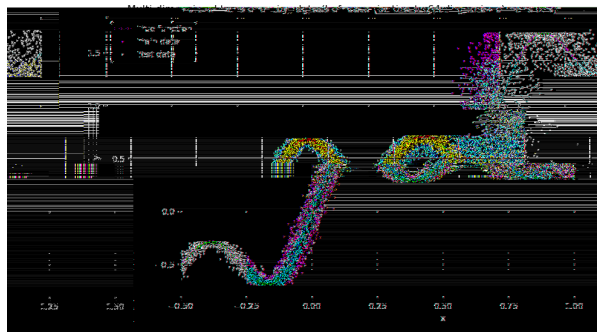


Figure 4.10. Plot of the Multi-dimensional toy data training set and test set. The noise increases with x for $x \geq 0.5$.

set respectively. The data is visualised in figure 4.10. The plot shows how the noise grows as x increases when $x \geq 0.5$.

By having the dataset be a projection of a one-dimensional dataset we preserve some of the advantages of the one-dimensional toy data, like being able to easily understand and visualise the true function.

Communities and Crime dataset

The Communities and Crime dataset [72] is a regression dataset of 1994 datapoints with 127 features. Out of the 127 features, 122 are predictive features. Because 22 of the predictive features have missing values, we remove them and are left with 100. The prediction target of the dataset is per capita violent crime. This target feature is a scalar in the range $[0; 1]$.

We divided the dataset into a training set with 70% of the data, a validation set with 10% of the data, and a test set with the final 20% of the data.

Because no out-of-distribution data exists for the Community and Crime data, we made it ourselves. We did so by adding Gaussian noise with $\mu = 0$ and $\sigma = 0.5$ to the test data. With this amount of noise we aim to moderately challenge the regression models.

Data standardisation

All features and targets for the regression datasets are standardised so that all training features are in the range $[-1, 1]$. The target y_i and each feature $x_{j,i}$ is normalised as:

$$\bar{\mathbf{y}} = 2 \cdot \left(\frac{\mathbf{y} - \min(\mathbf{y})}{\max(\mathbf{y})_{\text{train}} - \min(\mathbf{y})_{\text{train}}} \right) - 1$$
$$\bar{\mathbf{x}}_i = 2 \cdot \left(\frac{\mathbf{x}_i - \min(\mathbf{x}_i)_{\text{train}}}{\max(\mathbf{x}_i)_{\text{train}} - \min(\mathbf{x}_i)_{\text{train}}} \right) - 1$$

where $\min(\mathbf{y})_{\text{train}}$, $\max(\mathbf{y})_{\text{train}}$ and $\min(\mathbf{x})_{\text{train}}$,$\max(\mathbf{x})_{\text{train}}$ are the minimum and maximum values of x and y in the training data. By using the minimum and maximum from the training data we avoid data leakage.

4.7.2 Classification tasks

For the classification tasks we use the CIFAR-10 and CIFAR-100 datasets [73]. The CIFAR datasets are commonly used for benchmarking image classification performance of deep ensemble models [15,17–19]. The images in CIFAR are 32 by 32 pixels and have 3 colour channels.

For classification with few classes, the image dataset CIFAR-10 [73] is used. CIFAR-10 is made up of $50,000$ training images and $10,000$ test images, each belonging to one of 10 classes. Each class is equally represented in both the test and training set, with each class having $5,000$ training images and $1,000$ test images. A validation set of $5,000$ images is randomly sampled from the training set. Because it is randomly sampled, the validation set does not contain an equal amount of images from all classes.

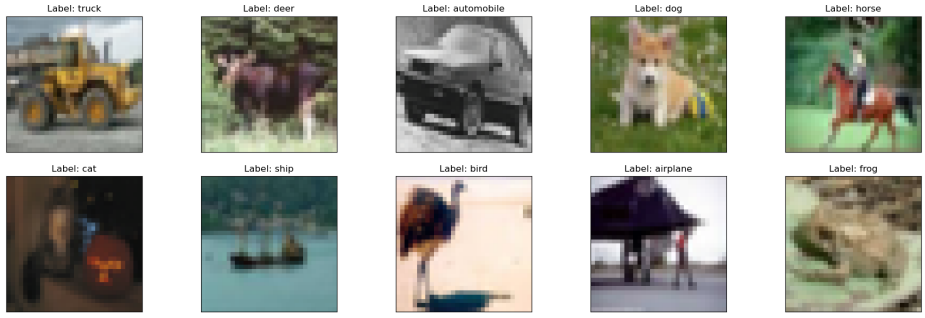


Figure 4.11. Examples of images from the CIFAR-10 dataset. The images clearly depict the subject of their label despite being low resolution.

For a more complex image classification task we use CIFAR-100. This dataset is very similar to CIFAR-10 in that is has the same number of training images and test images. Since it has 10 times as many classes, each class has 10 times fewer images, with each class having 500 images in the training set and 100 images in the test set. Like with CIFAR-10, a validation set of $5,000$ is randomly split from the training set, which means that the validation set is not class balanced.

Out-of-distribution data

In order to test the robustness of our models when faced with unfamiliar data, we use the CIFAR-10-C dataset [74] as an out-of-distribution test set. The CIFAR-10-C dataset has various types of corruption that is applied to the original test set of CIFAR-10 with varying severities from 1 to 5 as seen in figure 4.12. For this work, since we are not conducting a full robustness study for all our models, we will only be testing with impulse noise corruption with severity 5. By using a high severity noise corruption we seek to evaluate the uncertainty estimates of our models in a situation where they should be uncertain, regardless of how well they perform on the base dataset. We also expect impulse noise to be challenging for the classifiers, which should create an opportunity for more uncertain models to perform well.

Data processing

The CIFAR-10 and CIFAR-100 images are normalised using the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ of the images in the training data. Each image $\boldsymbol{x}$ is normalised as:

$$|\boldsymbol{x}| = \frac{(\boldsymbol{x} - \boldsymbol{\mu})}{\boldsymbol{\sigma}}$$

CIFAR-10-C and CIFAR-100-C are also normalised, but in order to avoid data leakage, we use the mean and standard deviations of their corresponding in-distribution datasets, e.g. for CIFAR-10-C we use the mean and standard deviation of the CIFAR-10 training data to standardise.

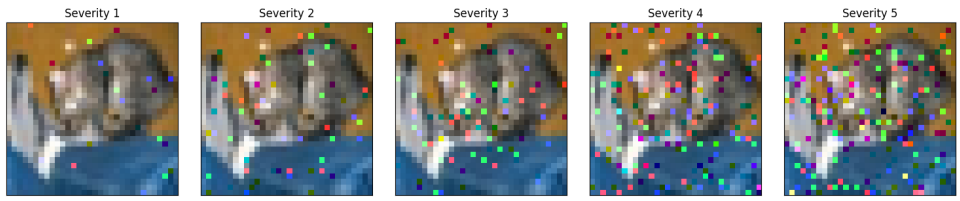


Figure 4.12. This figure shows an example image from CIFAR-10-C with the impulse noise corruption. The severity of the corruption intensifies from left to right.

4.8 Experimental setup

In this section, we provide an overview of the experimental setup for the different experiments we will conduct in this thesis. Moreover, we describe our choice of models and hyperparameters for each experiment.

4.8.1 Experiments

We test five types of models: one baseline model, one Bayes by Backprop BNN model, and three subnetwork ensemble models: MIMO, naive multiheaded, and MIMBO.

The baseline model is a standard deterministic neural network. It corresponds to a MIMO-configured neural network with $M = 1$ subnetwork. For the subnetwork ensemble models, we investigate how using $M = \{2, 3, 4, 5\}$ subnetworks affects the predictive performance and uncertainty estimates. BNN corresponds to a MIMBO model with $M = 1$ subnetwork.

All models are trained on the three regression datasets and two classification datasets. For both regression and classification datasets, the models are tested on both in-distribution test data and out-of-distribution test data. We use two network architectures for the classification datasets: MediumCNN and the deeper Wide ResNet architecture.

4.8.2 Sampling efficiency of Bayesian neural networks

Dusenberry et al. [15] show that increasing the number of samples at inference can yield an increase in model performance. However, beyond a certain number of samples, the effect of adding another sample diminishes.

We perform a similar experiment to determine the appropriate amount of samples S to use at inference. We evaluate the sampling efficiency for the MediumCNN BNN and MIMBO models on the CIFAR-10 dataset. We evaluate the models on accuracy, Brier score, NLL and ECE for $S = \{1, 2, 4, 8, 16, 32\}$.

4.8.3 Diversity in subnetworks

Fort et al. [17] showed that diversity is important for the performance of ensemble models. In order to evaluate the diversity of the MIMO, Naie and MIMBO models, we calculate the two diversity metrics: Disagreement and KL-divergence between each subnetwork in the models. We use the checkpoints of the subnetworks we used for

the PCA plots to calculate the diversity metrics on the final checkpoint. This means that only a single batch of the validation set is evaluated. We choose to evaluate the checkpoint data so we can compare the measured diversity with the optimisation trajectory plots in PCA space, which will help us make sense of them.

4.8.4 Training details

In our implementation we used some techniques to help regularise and reduce training time. They are described in this section.

Saving the best model

During training we save the model at the epoch where it has the best validation loss for regression models and the best validation accuracy for classification models. This ensures that the saved model does not get impacted by overfitting. Saving the model this way has a regularising effect.

Early Stopping

We use early stopping when training our networks, as we noticed the different models converged at very different rates. With early stopping, we can save significant time during training. Additionally, early stopping functions as regularisation as it prevents overfitting by stopping training if the validation loss stops decreasing. However, the regularising effect of early stopping does not apply to our models, as we save the model when it peaks in terms of loss. We implemented early stopping such that if the NLL has not decreased for 10 epochs, training stops.

Experiment repetitions

All experiments are repeated five times with differently initialised but identically trained models. We do this to reduce how much results would be skewed from a particular model being poorly initialised and thus performing worse than what would be representative for the model.

When results are calculated, each metric is averaged over the 5 repetitions, with the resulting mean μ being used as the final result. A 95% confidence interval is then calculated with the standard error of the mean:

$$\mu \pm 1.96 \cdot \frac{\sigma}{\sqrt{5}}$$

With confidence intervals we have a chance of knowing if our results are somewhat skewed from outliers.

4.8.5 Hyperparameters

In order to successfully train our models we need to select fitting hyperparameters. Because of the amount of hyperparameters that can be tuned for each model, for some we simply select a fixed value across all models that we believe to be a fitting value. There are different hyperparameters for Bayesian and non-Bayesian models, as well as for regression and classification models.

These fixed parameters for the MLPs, MediumCNNs and Wide ResNets are displayed in Table 4.2, 4.3 and 4.4 respectively. The parameters are fixed because initial experiments demonstrated that they have little to no impact on the performance if they are chosen appropriately or we left them out of the hyperparameter sweeps due to time-constraints

Parameter	**Chosen value**
`Epochs`	1500
`Batch` size	128
`Learning rate` α	$10^{-3}, 3 \times 10^{-4}$
`Optimiser`	`Adam`
`Scheduler`	`ReduceLROnPlateau`
`Scheduler patience`	100

Table 4.2. Hyperparameter choices for all regression models.

Parameter	**Chosen value**
`Epochs`	200
`Batch size*`	256
`Learning rate` α	10^{-3}
`Optimiser`	`Adam`
`Scheduler`	`ReduceLROnPlateau`
Scheduler patience	5

Table 4.3. Hyperparameter choices for all MediumCNN models. *The default batch size is listed in the table, but is changed to 128 for Naive $M = 4, 5$ and MIMBO $M = 2, 3, 4, 5$ due to hardware limitations

Parameter	Chosen value
`Epochs`	200
`Batch size*`	256
`Learning rate` α	10^{-3}
`Optimiser`	`Adam`
`Scheduler`	`ReduceLROnPlateau`
`Scheduler patience`	5

Table 4.4. Hyperparameter choices for all Wide ResNet models. *The default batch size is listed in the table, but is changed to 128 for Naive $M = 4, 5$ and MIMBO $M = 2, 3, 4, 5$ due to hardware limitations

Below, we give a brief explanation for the choice of each hyperparameter.

Epochs: The number of epochs is selected so that we ensure that all models converge during training. To select the specific numbers in the tables, we ran test runs of various models to see how many epochs would be necessary for them to converge. Because all models are trained with early stopping the epochs parameter only acts as an upper bound.

Batch size: The batch size was chosen to be 256 by default. We selected it to be as high as possible considering hardware limitations. In some cases we had to reduce the batch size. Because the 1D toydata training set was only $2,000$ points, we reduced the batch size to 128 so that MIMO and MIMBO always are be able to train on more than one batch in each epoch. Since the real batch size for subnetwork models is multiplied by the number of subnetworks, a single batch could take up more than half the training set. For some Naive models and all MIMBO models with the Wide ResNet architecture it was necessary to reduce the batch size to 128 to avoid running out of GPU memory.

Learning rate: Initial learning rate is generally fixed across models, but for the Bayesian models, BNN and MIMBO, a lower learning rate α is necessary to avoid divergence. Thus we use $\alpha = 3 \times 10^{-4}$ for the Bayesian models.

Optimiser: During training, the models are trained using the Adam optimiser with an initial learning rate of α. We could have considered SGD with momentum, but we decided against using SGD as it had stability issues with our Bayesian models.

Scheduler: We use a learning rate scheduler to help with model convergence during training. Our chosen learning rate scheduler halves the learning rate if there is no improvement in the chosen metric after a certain number of epochs, with that number being the `Schedler patience`. For regression tasks, the scheduler halves the learning rate if there is no improvement in the validation loss for 100 epochs. For classification tasks, the learning rate is halved if there is no improvement in validation accuracy for 5 epochs. In test runs we found that these choices for the patience led to the learning rate being annealed at appropriate times.

To ensure that the Bayesian and non-Bayesian models were trained in a functionally equivalent way, the non-Bayesian models were trained with weight decay. Since the weights of the Bayesian models are regularised by the Gaussian prior, which works as L2-regularisation, the non-Bayesian models should be regularised as well. We do this using weight decay built into the Adam optimiser, since it is implemented as L2-regularisation [75]. Having similar regularisation strengthens the comparisons of different models, as less regularised models are more likely to overfit.

Hyperparameter sweep

We do a more thorough hyperparameter tuning for hyperparameters related to the regularisation of the models. This includes the variance of the prior for Bayesian models and strength of the l2 regularisation for non-Bayesian models. In both cases, we use the σ hyperparameter which for the Bayesian models is the standard deviation of the Gaussian prior, and for the non-Bayesian models is a factor in the calculation of the L2 regularisation strength λ:

$$\lambda = \frac{1}{2\sigma^2}$$

Moreover, since the Wide ResNet architecture includes dropout layers, we tune the dropout rate for the Wide ResNet models. For Wide ResNets we swept over three values for the dropout rate: $\{0, 0.15, 0.3\}$ In preliminary testing we found that any more than 0.3 would be excessive.

Since both regression and classification models use Gaussian priors for Bayesian models and L2 regularisation for non-Bayesian models, we sweep over the same values of σ for all of them:

$$\sigma = \{1, 3, 5, 10, 30, 50, 5000\}$$

The values that we sweep over were selected to represent a wide array of regularisation strengths, with $\sigma = 1$ meaning strong regularisation and $\sigma = 5000$ meaning essentialy no regularisation.

As an exception, we did not sweep for the hyperparameters used for the Wide ResNet Baseline models. In order to ensure that the Baseline was comparable with the results of other studies, we trained the baseline as described in the paper on Wide ResNets by Zagoruyko et al. [67]. They use a higher initial learning rate of $\alpha = 0.1$, SGD optimiser with Nesterov momentum of 0.9, a weight decay of 0.0005 and batch size of 128. Additionally they use a different scheduler for the learning rate, where the learning rate is multiplied by 0.2 at $60, 120, 160$ epochs.

The results of our hyperparameter sweeps can be found in appendix E.

CHAPTER 5

Results

This section presents the results from the experiments described in the previous section. We begin by showing how the performance of Bayesian models are affected by the number of times weights are sampled.

Then we present the results of the models we trained on the aforementioned datasets. The performance of each model are summarised with relevant metrics and presented in tables for each test dataset.

The name of each metric is followed by an arrow indicating whether it should be maximised (↑) or minimised (↓). Each metric presented in a table is the mean over 5 identically trained but differently initialised models with the same configuration. The 95% confidence interval is denoted as a number in a parenthesis after the mean. This number is what should be added and subtracted from the last decimal of the mean to find the upper and lower bounds of the confidence interval.

For each task, the tables for the in-distribution and out-distribution results are presented side by side, with the in-distribution results on the left and the out-of-distribution results on the right. The best metric is highlighted with bold numbers.

After presenting the results on all datasets, we investigate the diversity of the subnetworks in the MIMO, Naive, and MIMBO models. The optimisation trajectories of subnetworks in the various subnetwork models are presented to explore how subnetwork diversity affects model performance and calibration.

5.1 Sampling effectiveness

The Bayesian models, such as BNN and MIMBO, learn an approximate posterior distribution over the model weights w. At inference time, this means we can construct an ensemble model by sampling multiple sets of weights from the posterior and average over the individual models.

We sample $S = \{1, 2, 4, 8, 16, 32\}$ times from the posteriors of the Bayesian models trained on CIFAR-10 to explore how many samples are needed before we get diminishing returns on the classification metrics: accuracy, Brier score, ECE and NLL. In

this experiment, we consider only the BNN and MIMBO models with the MediumCNN architecture, as MediumCNN models are faster to perform inference on than the Wide ResNet models.

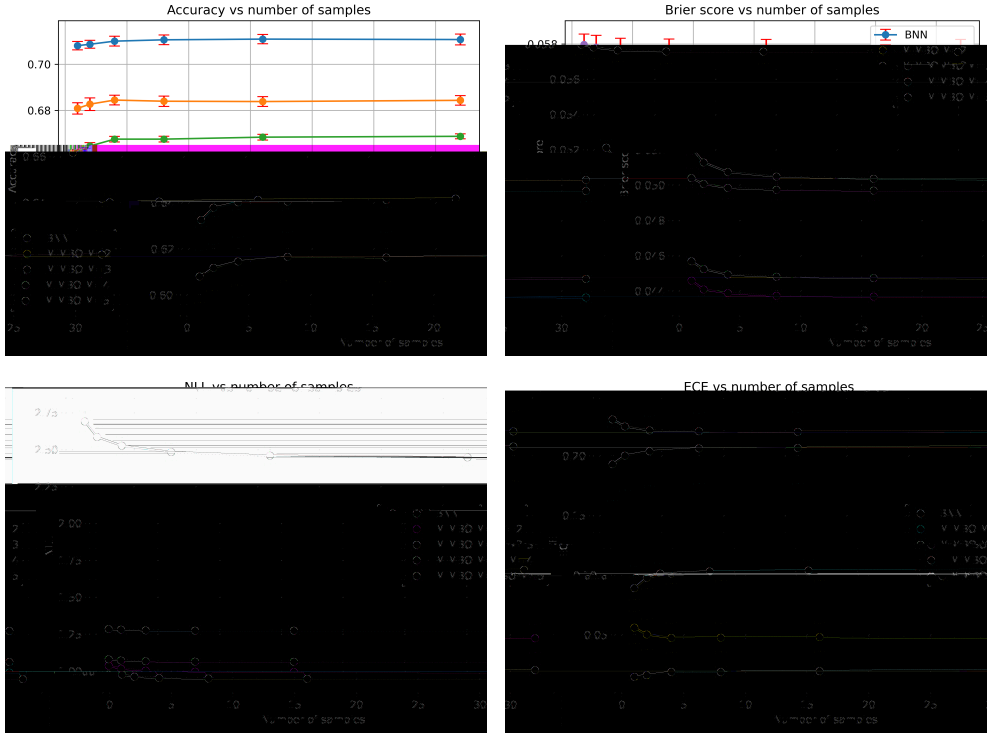


Figure 5.1. Number of samples vs Accuracy, Brier score, ECE and NLL for Bayesian models with the MediumCNN architecture, evaluated on CIFAR-10. Each result is the mean over $n = 5$ repetitions, and the error bar indicate a 95% confidence interval, $\mu \pm 1.96 \cdot \frac{\sigma}{\sqrt{n}}$, where σ is the standard deviation. The plots show that increasing the number of samples up to a limit results in improved performance in the accuracy, Brier score, NLL, and in some cases the ECE.

Figure 5.1 illustrates the effect of sampling multiple sets of weights from learned weight posterior $q(\mathbf{w}|\theta)$ and then averaging the resulting models. Sampling multiple times from the posterior yields improved performance for all models on the accuracy, Brier score, and NLL, and improvements in the ECE for BNN and MIMBO $M = 2$. For MIMBO $M > 2$, increasing the number of samples results in a small increase in the ECE. Generally, the sampling effect diminishes after $S = 8$ times, and the improvement in performance becomes mostly negligible. For the remaining classification experiments, we therefore use $S = 10$ samples at inference for the Bayesian models.

5.2 Regression results

In this section, we present the results for our three regression tasks on the one-dimensional toy dataset, multi-dimensional toy dataset and Communities and Crime dataset. The models are evaluated on root mean squared error (RMSE), Gaussian negative log-likelihood (GNLL) and expected calibration error (ECE).

5.2.1 One-dimensional toy dataset

In the following section, the results on the one-dimensional toy dataset are presented.

(a) In-distribution

Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	0.0778(7)	**−2.43(1)**	0.00194(12)
MIMO $M = 2$	0.0779(2)	−2.33(1)	0.00249(12)
MIMO $M = 3$	0.0802(26)	−2.30(2)	0.00231(18)
MIMO $M = 4$	0.115(19)	−2.12(7)	0.00479(211)
MIMO $M = 5$	0.127(9)	−2.00(3)	0.00741(221)
Naive $M = 2$	**0.0776(4)**	−2.37(2)	0.00214(9)
Naive $M = 3$	0.0780(3)	−2.34(2)	0.00220(9)
Naive $M = 4$	0.0778(3)	−2.34(2)	0.00221(8)
Naive $M = 5$	0.0777(2)	−2.37(3)	0.00221(7)
BNN	0.0793(16)	−2.29(3)	**0.00191(20)**
MIMBO $M = 2$	0.0809(20)	−2.15(4)	0.00316(48)
MIMBO $M = 3$	0.121(4)	−1.70(3)	0.00768(94)
MIMBO $M = 4$	0.152(3)	−1.43(2)	0.00762(107)
MIMBO $M = 5$	0.281(7)	−0.229(80)	0.0533(35)

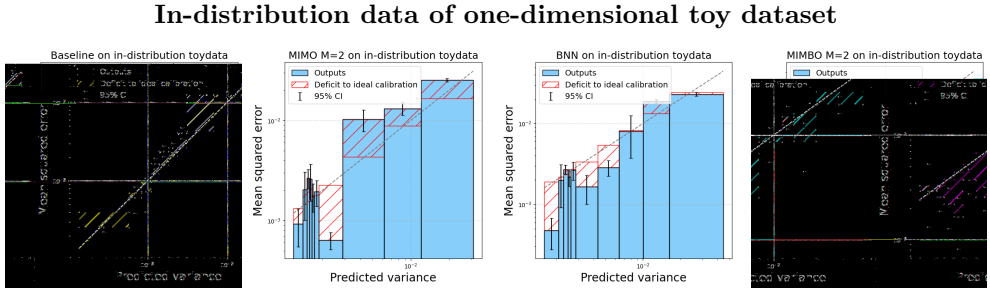
(b) Out-of-distribution

Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	0.596(109)	−0.114(667)	0.289(171)
MIMO $M = 2$	0.566(125)	0.891(590)	0.336(203)
MIMO $M = 3$	0.395(62)	0.387(941)	0.268(119)
MIMO $M = 4$	0.748(250)	0.527(380)	0.622(170)
MIMO $M = 5$	0.924(89)	1.46(76)	0.617(78)
Naive $M = 2$	0.598(66)	1.25(185)	0.258(119)
Naive $M = 3$	0.598(108)	4.38(391)	0.273(142)
Naive $M = 4$	**0.546(135)**	2.78(339)	0.247(129)
Naive $M = 5$	0.648(102)	1.80(99)	0.382(133)
BNN	0.595(19)	**−0.442(83)**	0.205(61)
MIMBO $M = 2$	0.547(87)	0.0788(3066)	**0.156(81)**
MIMBO $M = 3$	0.572(39)	0.943(368)	0.282(57)
MIMBO $M = 4$	0.862(64)	3.84(114)	0.670(119)
MIMBO $M = 5$	0.679(9)	3.81(121)	0.410(18)

Table 5.1. Results on in-distribution test data (left) and out-of-distribution test data (right) for the one-dimensional toy dataset. Bold indicates best result for each metric.

The in-distribution results for the one-dimensional regression task are presented in table 5.1(a). The table shows that Naive $M = 2$ achieves the best RMSE, while Baseline and BNN achieve the best GNLL and ECE respectively. In general, the Naive models, Baseline, MIMO $M = 2, 3$ and BNN all achieve comparable results on RMSE, GNLL and ECE. From the table, we can also observe that increasing the number of subnetworks for the MIMO and MIMBO models leads to worse performance and calibration, as indicated by the significantly higher RMSE, GNLL and ECE. However, this trend is not observed for the Naive models, which all have comparable predictive performance, regardless of the number of subnetworks.

Among the MIMO and MIMBO models, the best performing models are MIMO $M = 2$ and MIMBO $M = 2$. We investigate how well-calibrated these models are compared to the Baseline and BNN by analysing their reliability diagrams.

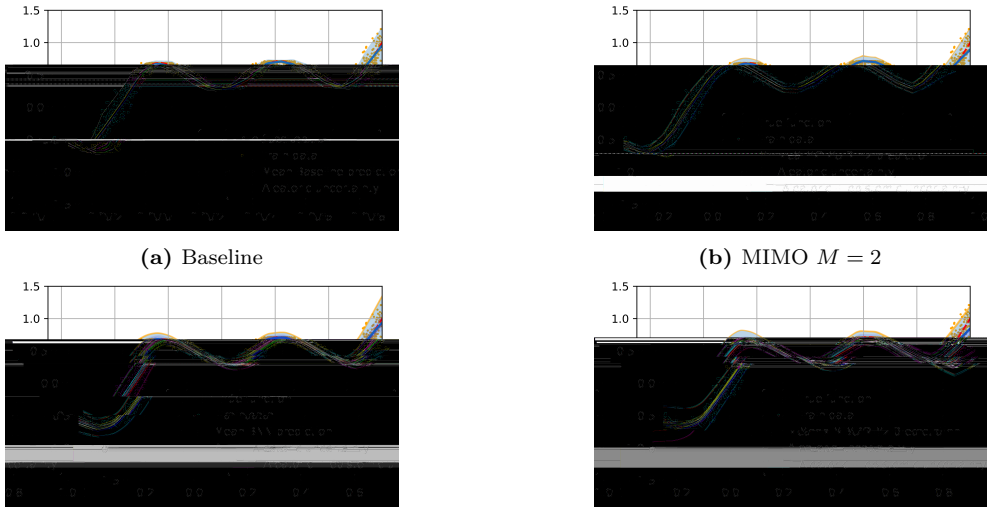


(a) Baseline **(b)** MIMO $M = 2$ **(c)** BNN **(d)** MIMBO $M = 2$

Figure 5.2. Reliability diagrams for Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$ evaluated on in-distribution test data of one-dimensional toy dataset. The reliability diagrams of Baseline (left) and MIMO $M = 2$ are overconfident in predictions with large predicted variances. In comparison, the reliability diagrams for BNN (centre right) and MIMBO $M = 2$ (right) are generally less confident, particularly for predictions with large predicted variances.

The reliability diagrams for Baseline on figure 5.2(a) looks quite well-calibrated, although it is slightly overconfident in predictions with large predicted variances and slightly underconfident in predictions with small predicted variances. MIMO $M = 2$ and BNN are models that directly branch off the Baseline, by either adding a subnetwork or making the Network Bayesian, and MIMBO $M = 2$ combines both of these properties. However, as seen on figure 5.2(b), 5.2(c) and 5.2(d), the reliability diagrams do not look more well-calibrated, even though BNN achieves a slightly smaller ECE.

Consider the regression plots for Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$ that show the predictions and uncertainty for each model.



(a) Baseline

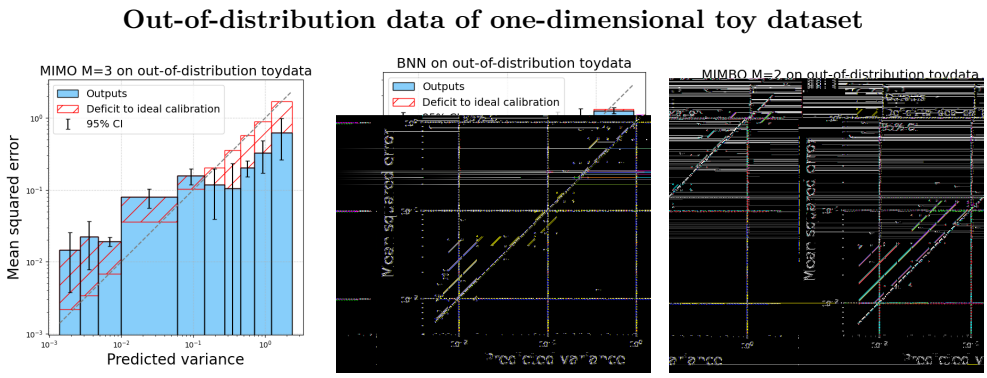
(b) MIMO $M = 2$

(c) BNN

(d) MIMBO $M = 2$

Figure 5.3. Regression plots for Baseline, MIMO $M = 2$ BNN and MIMBO $M = 2$ on the in-distribution test data of the one-dimensional toy dataset. The predictions for all four models approximate the true function accurately. Compared to the Baseline and MIMO $M = 2$, BNN and MIMBO $M = 2$ tend to predict larger variances. This is especially noticeable where the slope of the function is zero. The models that are visualised are the ones with the lowest GNLL of the five repetitions.

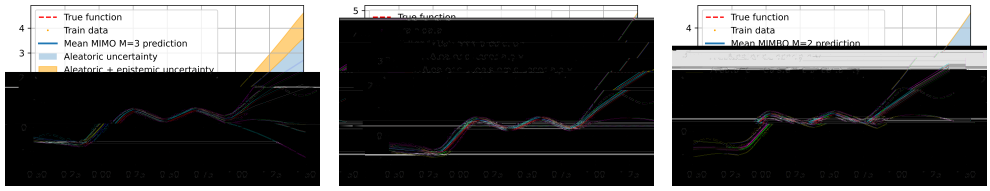
Figure 5.3 shows that all models are able to learn the underlying function from the noisy training data quite well. Furthermore, the models are able to quantify the aleatoric uncertainty of the training data accurately. We can observe on figure 5.3(c) and 5.3(d) that BNN and MIMBO tend to predict larger variances than Baseline and MIMO $M = 2$ (figure 5.3(a) and 5.3(b) respectively), especially where the slope of the function is zero. The larger predicted variances means the models are more uncertain, which explains why BNN and MIMBO $M = 2$ are less overconfident compared to Baseline and MIMO $M = 2$. 5 4 moetf the theOn-distributio4 ans go4



(a) MIMO $M = 3$ **(b)** BNN **(c)** MIMBO $M = 2$

Figure 5.4. Reliability diagrams for MIMO $M = 3$, BNN and MIMBO $M = 2$ evaluated on out-of-distribution test data of one-dimensional toy dataset. On the out-of-distribution test data, all three models become increasingly overconfident. MIMO $M = 3$ is the sole model that becomes underconfident in predictions with large predicted variances.

The reliability diagrams on figure 5.4 illustrate that MIMO $M = 3$, BNN and MIMBO $M = 2$ overall become overconfident on the out-of-distribution data. Specifically for MIMO $M = 3$, it becomes underconfident in predictions with large predicted variances. To understand why this is the case, consider the regression plots on the out-of-distribution data, on figure 5.5.



(a) MIMO $M = 3$ **(b)** BNN **(c)** MIMBO $M = 2$

Figure 5.5. Regression plots for Baseline, MIMO $M = 3$ BNN and MIMBO $M = 2$ on the out-of-distribution test data of the one-dimensional toy dataset. On the out-of-distribution intervals $[-0.50, -0.25]$, the uncertainty is small and constant. In the other out-of-distribution interval $[1.0, 1.5]$, the uncertainty increases as the models evaluate out-of-distributions samples further from the training data. The uncertainty for MIMO $M = 3$ is much larger than that of BNN and MIMBO $M = 2$, in particular due to the epistemic uncertainty. The models that are visualised are the ones with the highest GNLL of the five repetitions.

The regression plots on figure 5.5 illustrate that the uncertainty of the models in the interval $[-0.50, -0.25]$ is small, which leads to the overconfidence in predictions with small predicted variances, as seen on the reliability diagrams on figure 5.4. In the out-

of-distribution interval $[1.0, 1.5]$, the uncertainty of the three models increases as they evaluate out-of-distribution samples further from the training data. As seen on the figure, the uncertainty of BNN is slightly larger than that of MIMBO $M = 2$, but the uncertainty of MIMO $M = 3$ is larger than both, which explains the underconfidence in predictions with large predicted variances on the reliability diagram for MIMO $M = 3$ on figure 5.4(a).

5.2.2 Multi-dimensional toy dataset

In the following section, the results for the multi-dimensional toy dataset are presented.

(a) In-distribution

Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	**0.109(0)**	**−2.27(0)**	0.00908(32)
MIMO $M = 2$	0.110(1)	−2.26(0)	0.00902(43)
MIMO $M = 3$	**0.109(0)**	−2.26(0)	0.00964(32)
MIMO $M = 4$	0.110(0)	−2.25(0)	0.00948(33)
MIMO $M = 5$	0.110(0)	−2.24(1)	0.00942(32)
Naive $M = 2$	0.110(0)	**−2.27(0)**	**0.00899(52)**
Naive $M = 3$	**0.109(0)**	**−2.27(0)**	0.00955(24)
Naive $M = 4$	**0.109(0)**	**−2.27(0)**	0.00934(24)
Naive $M = 5$	**0.109(0)**	**−2.27(0)**	0.00900(27)
BNN	0.110(0)	−2.22(1)	0.0185(61)
MIMBO $M = 2$	0.111(0)	−2.17(2)	0.0230(51)
MIMBO $M = 3$	0.113(2)	−2.07(3)	0.0225(60)
MIMBO $M = 4$	0.116(3)	−1.95(5)	0.0355(48)
MIMBO $M = 5$	0.122(10)	−1.90(6)	0.0281(74)

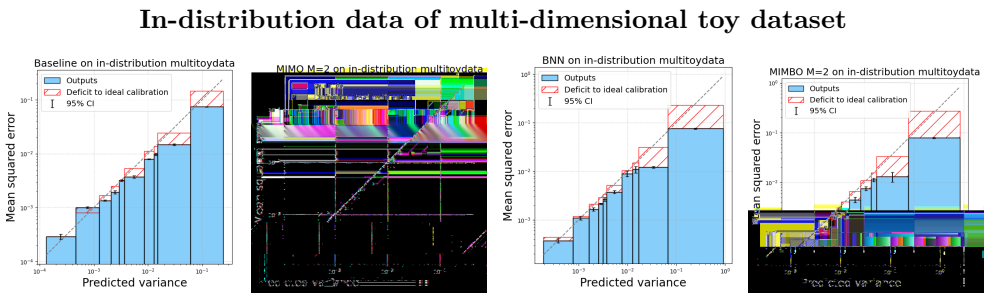
(b) Out-of-distribution

Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	0.533(54)	−0.110(67)	1.01(23)
MIMO $M = 2$	0.487(114)	−0.0857(1033)	1.86(83)
MIMO $M = 3$	0.528(80)	−0.0614(505)	2.05(46)
MIMO $M = 4$	0.418(43)	0.0696(1316)	4.53(228)
MIMO $M = 5$	0.370(85)	−0.151(131)	1.59(43)
Naive $M = 2$	0.381(85)	−0.241(113)	**0.917(233)**
Naive $M = 3$	**0.364(35)**	−0.170(152)	1.347(699)
Naive $M = 4$	0.388(26)	−0.117(35)	1.677(457)
Naive $M = 5$	0.402(71)	**−0.304(170)**	0.975(546)
BNN	0.389(64)	0.187(99)	5.31(172)
MIMBO $M = 2$	0.397(42)	0.202(105)	6.35(148)
MIMBO $M = 3$	0.432(71)	0.113(143)	3.79(169)
MIMBO $M = 4$	0.400(36)	0.415(94)	3.95(101)
MIMBO $M = 5$	0.436(71)	0.285(91)	1.78(80)

Table 5.2. Results of testing on in-distribution test data (left) and out-of-distribution test data (right) from the multi-dimensional toy dataset. Bold indicates best result for each metric.

Table 5.2(a) shows the results on the in-distribution test data of the multi-dimensional toydata. Overall, the Baseline, MIMO models and Naive models achieve very similar results on RMSE, GNLL and ECE. The BNN and MIMBO models achieve slightly worse RMSE and GNLL, and a much higher ECE.

Given the larger ECE for the Bayesian models, it makes sense to analyse how their predictions and uncertainty estimates differ compared to non-Bayesian models. We consider the Baseline, MIMO $M = 2$ (most well-calibrated MIMO model),BNN and MIMBO $M = 2$ (most well-calibrated MIMBO model).

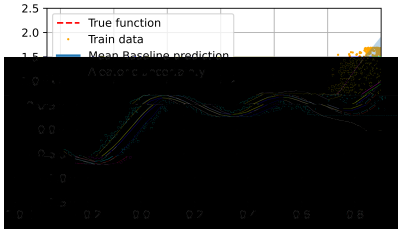


(a) Baseline **(b)** MIMO $M = 2$ **(c)** BNN **(d)** MIMBO $M = 2$

Figure 5.6. Reliability diagrams for Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$ evaluated on in-distribution test data of multi-dimensional toy dataset. The reliability diagram for BNN (centre right) and MIMBO $M = 2$ (right) have a much bigger calibration deficit than the Baseline (left) and MIMO $M = 2$ (centre left) in the two right-most bins, contributing to a higher ECE for BNN and MIMBO $M = 2$.

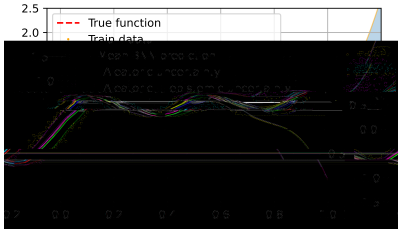
The reliability diagrams for the four models on figure 5.6 look very similar. The reliability diagram of Baseline on figure 5.6(a) is already quite well-calibrated, although slightly underconfident. MIMO $M = 2$ and BNN are both models that directly branch off the Baseline, by either adding a subnetwork or making the network Bayesian, and MIMBO $M = 2$ does both. The reliability diagram of MIMO $M = 2$ on figure 5.6(b) is almost identical to the Baseline, while the reliability diagram on 5.6(c) and 5.6(d) show that the BNN and MIMBO $M = 2$ become more underconfident. This is especially evident on the two right-most bins on the reliability diagrams, where BNN and MIMBO $M = 2$ have a much wider calibration deficit than Baseline and MIMO $M = 2$. Recall that the axes on the reliability diagrams are log-scaled, meaning that the calibration deficits on the right-most bins contribute much more towards the ECE.

The reason behind the larger calibration deficit can be explained by inspecting the regression plots for the Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$.

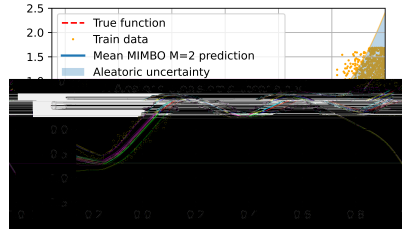


(a) Baseline

(b) MIMO $M = 2$



(c) BNN



(d) MIMBO $M = 2$

Figure 5.7. Regression plots for Baseline, MIMO $M = 2$ BNN and MIMBO $M = 2$ on the in-distribution test data of the multi-dimensional toy dataset. In the interval $[-0.25, 0.8]$, the predictions for all four models approximate the true function quite well, and they predict variances that accurately quantify the uncertainty in the training data. In the interval $[0.8, 1.0]$, the predictions begin to deviate from the true function, and the predicted variances are larger, especially for BNN and MIMBO $M = 2$. The models that are visualised are the ones with the highest GNLL of the five repetitions.

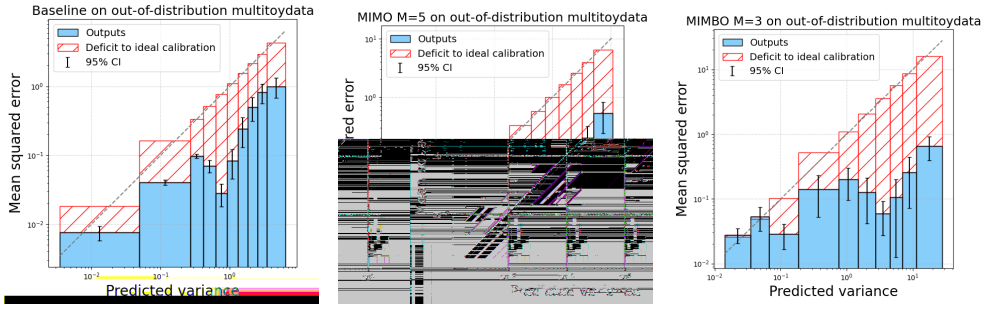
The regression plots on figure 5.7 show the predictions for the Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$ models on in-distribution data. On figure 5.7(b) and 5.7(c), we observe that the predicted uncertainty for BNN and MIMBO $M = 2$ in the interval $[0.8, 1.0]$ is much larger than the predicted uncertainty for Baseline and MIMO $M = 2$. This makes the Bayesian models underconfident, which explains why the calibration deficits become larger on figure 5.6(c) and 5.6(d).

We now turn our attention to the out-of-distribution results of the multi-dimensional toy dataset presented in table 5.2(b). Naive $M = 3$ achieves the best RMSE, while Naive $M = 2$ and Naive $M = 5$ achieve the best GNLL and ECE respectively. Both Naive $M = 2$ and Naive $M = 5$ achieve better performance than the Baseline on all three metrics.

For up to $M = 5$ subnetworks, the Bayesian models, BNN and MIMBO, achieve better RMSE than their non-Bayesian counterparts (Baseline and the MIMO models) with the same number of subnetworks. However, the Bayesian models are generally much worse calibrated, as they achieve a much larger GNLL and ECE. Among the Bayesian models, MIMBO $M = 3$ achieves the lowest GNLL, while MIMBO $M = 5$ achieves the lowest ECE.

We therefore investigate how MIMBO $M = 3$ compare to the Baseline and MIMO $M = 5$, which achieves the lowest GNLL among the MIMO models.

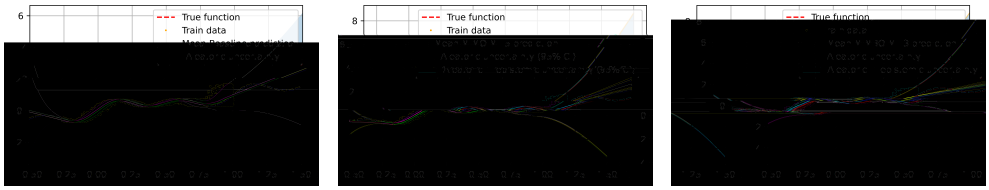
Out-of-distribution data of multi-dimensional toy dataset



(a) Baseline **(b)** MIMO $M = 5$ **(c)** MIMBO $M = 3$

Figure 5.8. Reliability diagrams for Baseline, MIMO $M = 3$ and MIMBO $M = 3$ evaluated on out-of-distribution test data of multi-dimensional toy dataset. The Baseline (left) is overconfident on the out-of-distribution data. MIMO $M = 5$ (centre) is still overconfident, whereas MIMBO $M = 2$ (right) alleviates some of the overconfidence issues and is overall better calibrated, which is reflected in the lower ECE.

Figure 5.8 shows that all models become overconfident on the out-of-distribution data. The Baseline is already quite underconfident, as shown on figure 5.8(a). The reliability diagrams for MIMO $M = 5$ and MIMBO $M = 3$ on figure 5.8(b) and 5.8(c) respectively become more confident for predictions wth small predicted variances but even more underconfident in predictions with large predicted variances, which contribute to a higher overall ECE, due to the log-scaled axes.



(a) Baseline **(b)** MIMO $M = 5$ **(c)** MIMBO $M = 3$

Figure 5.9. Regression plots for Baseline, MIMO $M = 5$ and MIMBO $M = 3$ on the test data of the multi-dimensional toy dataset. In the out-of-distribution intervals $[-0.5, -0.25]$ and $[1.0, 1.5]$, the uncertainty increases rapidly as the models evaluate out-of-distribution samples further from the training data, particularly on the right hand side, where heteroscedastic noise is present in the training data. The models that are visualised are the ones with the highest GNLL of the five repetitions.

The regression plots on figure 5.9(b) and 5.9(c) show that MIMO $M = 5$ and MIMBO

$M = 3$ are more underconfident in the out-of-distribution interval $[1.0, 1.5]$ than the Baseline, as they are more uncertain and predict much larger variances. In the other out-of-distribution interval $[-0.5, -0.25]$, MIMO $M = 5$ and MIMBO $M = 3$ are less uncertain, which the diagrams in figure 5.6(c) and 5.6(d) also indicate.

5.2.3 Community and Crime results

In the following section, the results on the Communities and Crime dataset are presented.

(a) In-distribution

Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	**0.140**(**0**)	−1.68(2)	0.00480(72)
MIMO $M = 2$	**0.140**(**1**)	**−1.72**(**1**)	0.00401(74)
MIMO $M = 3$	0.141(1)	−1.71(1)	0.00442(80)
MIMO $M = 4$	0.141(1)	−1.70(1)	**0.00388**(**50**)
MIMO $M = 5$	0.142(1)	−1.70(0)	0.00443(58)
Naive $M = 2$	0.141(1)	−1.68(1)	0.00518(50)
Naive $M = 3$	**0.140**(**1**)	−1.70(0)	0.00399(40)
Naive $M = 4$	**0.140**(**1**)	−1.69(2)	0.00550(53)
Naive $M = 5$	**0.140**(**2**)	−1.67(3)	0.00464(60)
BNN	0.143(1)	−1.69(2)	0.00836(180)
MIMBO $M = 2$	0.142(1)	−1.65(1)	0.00941(80)
MIMBO $M = 3$	**0.140**(**0**)	−1.65(1)	0.00781(81)
MIMBO $M = 4$	0.142(2)	−1.63(5)	0.00828(104)
MIMBO $M = 5$	**0.140**(**1**)	−1.63(2)	0.00942(50)

(b) Out-of-distribution

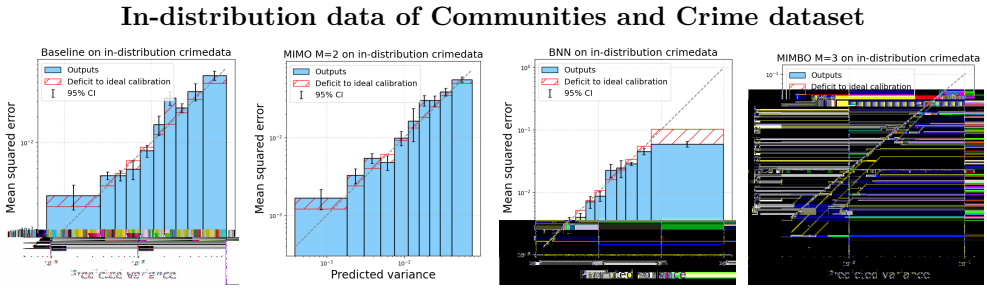
Model	RMSE ↓	GNLL ↓	ECE ↓
Baseline	**0.167**(**4**)	−0.870(370)	0.0131(18)
MIMO $M = 2$	0.182(6)	0.101(613)	0.0151(19)
MIMO $M = 3$	0.185(5)	−0.196(304)	0.0144(16)
MIMO $M = 4$	0.190(4)	0.0111(3050)	0.0149(15)
MIMO $M = 5$	0.185(2)	−0.245(341)	0.0116(8)
Naive $M = 2$	0.170(5)	−0.808(450)	0.0139(38)
Naive $M = 3$	0.170(3)	−1.05(18)	0.0135(11)
Naive $M = 4$	0.170(6)	−0.928(315)	0.0135(44)
Naive $M = 5$	0.172(3)	−0.834(103)	0.0147(12)
BNN	0.192(10)	−1.18(6)	0.0222(34)
MIMBO $M = 2$	0.171(2)	**−1.37**(**4**)	0.00784(33)
MIMBO $M = 3$	0.177(1)	−1.29(3)	0.00612(86)
MIMBO $M = 4$	0.179(2)	−1.30(2)	**0.00513**(**27**)
MIMBO $M = 5$	0.176(2)	−1.30(3)	0.00581(80)

Table 5.3. Results of testing on in-distribution test data (left) and out-of-distribution test data (right) from the Communities and Crime dataset. Bold indicates best result for each metric.

Table 5.3(a) shows the in-distribution results for the Communities and Crime dataset. All models generally achieve similar performance and calibration. Multiple model achieves the lowest RMSE of 0.140, but MIMO $M = 2$ achieves the lowest GNLL, while MIMO $M = 4$ achieves the lowest ECE. Changing the number of subnetworks for the MIMO, Naive, and MIMBO models seems to have little to no impact on their predictive performance.

Among the Bayesian models, BNN achieves a lowest GNLL, but MIMBO $M = 3$ achievs the joint lowest RMSE and lowest ECE. Compared to the MIMO models, the BNN and MIMBO models achieve a slightly lower GNLL and a much higher ECE.

We investigate the calibration of the Baseline, MIMO $M = 2$, BNN and MIMBO $M = 3$ further by inspecting their reliability diagrams.



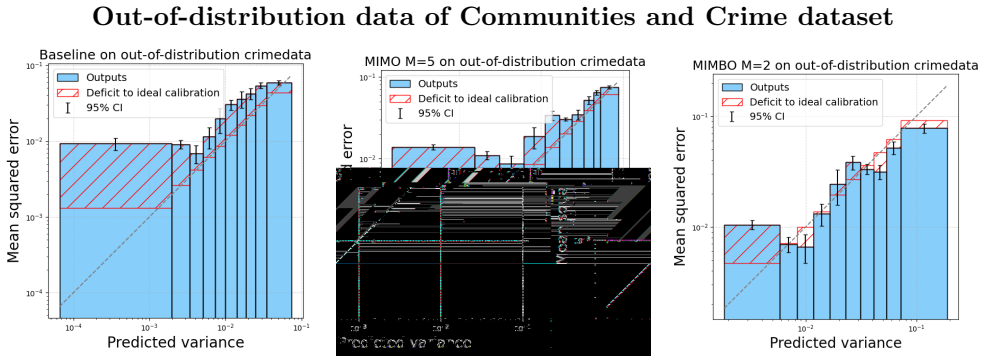
(a) Baseline **(b)** MIMO $M = 2$ **(c)** BNN **(d)** MIMBO $M = 3$

Figure 5.10. Reliability diagrams for Baseline, MIMO $M = 2$, BNN and MIMBO $M = 2$ evaluated on in-distribution test data of Communities and Crime dataset. The reliability diagram for Baseline is slightly overconfident in all of its predictions (left). Configuring subnetworks (MIMO), making the network Bayesian (BNN) or a combination of both (MIMBO) results in less overconfidence. However, in the case of BNN (centre right) and MIMBO (right), the models become too underconfident, leading to overall worse ECE.

The reliability diagrams on figure 5.10(a) and 5.10(b) look similar, and they are overall quite well-calibrated, although slightly overconfident. In comparison, the reliability diagram for BNN on figure 5.10(c) is also quite well-calibrated, except for the rightmost bin which has a notable calibration deficit. Recall that the axes are log-scaled, so the calibration deficit of the rightmost bin will make a large contribution to the ECE, even if the rest of the bins are perfectly calibrated. The reliability for MIMBO $M = 3$ on figure 5.10(d) is overall too underconfident, leading to a higher ECE.

We are interested in how the models perform under datasaet shift, and now focus on the out-of-distribution results for the Communities and Crime dataset in table 5.3(b). The table shows that the Baseline achieves the best RMSE, but MIMBO $M = 2$ and MIMBO $M = 4$ achieve the best GNLL and ECE respectively. MIMBO $M = 2$ has comparable RMSE to the Baseline, but with significantly lower GNLL and ECE. MIMBO $M = 2$ also outperforms BNN on all three metrics. One can also observe that the MIMBO models, which were the least well-calibrated on the in-distribution data, are the most well-calibrated on the out-of-distribution data.

We investigate how the distribution shift in the data affects the calibration, and why the MIMBO models achieve the best calibration on the out-of-distribution data. We compare MIMBO $M = 2$ to the Baseline and MIMO $M = 5$, which achieves the lowest GNLL and ECE among the MIMO models.



(a) Baseline **(b)** MIMO $M = 5$ **(c)** MIMBO $M = 2$

Figure 5.11. Reliability diagrams for Baseline, MIMO $M = 5$ and MIMBO $M = 2$ on out-of-distribution test data for the Communities and Crime dataset. The red shaded area indicates the gap to the ideal calibration, and the error bar represents the 95% over 5 repetitions. Each bin contains 10% of the predictions. Both Baseline and MIMO $M = 5$ are overconfident, in contrast to the more well-calibrated MIMBO $M = 2$.

Figure 5.11 illustrates that Baseline and MIMO $M = 5$ are more overconfident than MIMBO $M = 2$ on the out-of-distribution data. MIMBO $M = 2$ suffers from less overconfidence issues, and overall looks better calibrated, which why MIMBO $M = 2$ achieves a much lower ECE than the other two models.

5.2.4 Summary of regression results

The results for the three regression datasets show that many models are able to achieve good predictive performance on the in-distribution data. We find that there is not one model that performs the best in all three metrics, on neither in-distribution nor out-of-distribution data.

Through inspection of the reliability diagrams, we find that the Baseline is generally quite well-calibrated on the in-distribution data for all three datasets. However, on out-of-distribution data, the Baseline is overconfident on the one-dimensional toy dataset and Communities and Crime dataset, as seen in figure 5.4(a) and 5.11(a). Adding subnetworks or making the network Bayesian tends to make the models more uncertain than the Baseline. For cases where the Baseline is overconfident, such as the out-of-distribution case of the one-dimensional toy dataset and Communities and Crime dataset, this results in better calibrated models.

Overall, the relatively low complexity of the in-distribution regression tasks means that the Baseline is well-calibrated which makes the increased uncertainty in MIMO and MIMBO models undesirable. However, on out-of-distribution data, where Baseline can be overconfident, MIMO and MIMBO improve the calibration significantly.

5.3 Classification results

In this section we present the results of our experiments on the classification tasks. Since we trained models on the two datasets CIFAR-10 and CIFAR-100 we have divided the results on each dataset into separate sections. The summary metrics presented in the table: Accuracy, Brier score, Negative log likelihood (NLL) and Expected calibration error (ECE), are calculated as described in 4.6. Each table contains the results of all the models tested on both the in-distribution and out-of-distribution datasets.

5.3.1 CIFAR-10

We first present the results of the models trained on the CIFAR-10 dataset. CIFAR-10 presents a relatively simple image classification task. The models are evaluated on the test set of the in-distribution CIFAR-10 as well as the out-of-distribution CIFAR-10-C with severity 5. Since we have trained models with two distinct architectures, we present them separately. We start by evaluating their performance on the in-distribution dataset after which we see how the performance changes on the out-of-distribution data.

MediumCNN results

We begin by considering the models with the MediumCNN architecture. The results of testing on the CIFAR-10 in-distribution data and CIFAR-10-C out-of-distribution data is presented in table 5.4.

(a) MediumCNN on CIFAR-10.

Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.724(7)	0.0477(13)	1.93(10)	0.215(6)
MIMO $M = 2$	0.696(15)	0.0435(15)	1.12(24)	0.0697(355)
MIMO $M = 3$	0.669(11)	0.0445(13)	**0.943(31)**	0.0224(74)
MIMO $M = 4$	0.647(17)	0.0486(30)	1.03(7)	0.0815(325)
MIMO $M = 5$	0.607(15)	0.0567(14)	1.24(4)	0.170(32)
Naive $M = 2$	0.715(9)	0.0520(14)	3.27(16)	0.221(8)
Naive $M = 3$	**0.725(2)**	**0.0424(32)**	1.22(31)	0.134(62)
Naive $M = 4$	0.705(5)	0.0463(53)	1.79(95)	0.143(66)
Naive $M = 5$	0.715(5)	0.0498(36)	2.98(100)	0.193(36)
BNN	0.711(5)	0.0504(8)	2.47(12)	0.221(5)
MIMBO $M = 2$	0.684(5)	0.0437(6)	0.998(79)	0.0477(245)
MIMBO $M = 3$	0.669(2)	0.0448(3)	0.950(9)	**0.0203(86)**
MIMBO $M = 4$	0.641(6)	0.0497(12)	1.07(3)	0.104(18)
MIMBO $M = 5$	0.618(21)	0.0576(14)	1.28(6)	0.208(58)

(b) MediumCNN on CIFAR-10-C

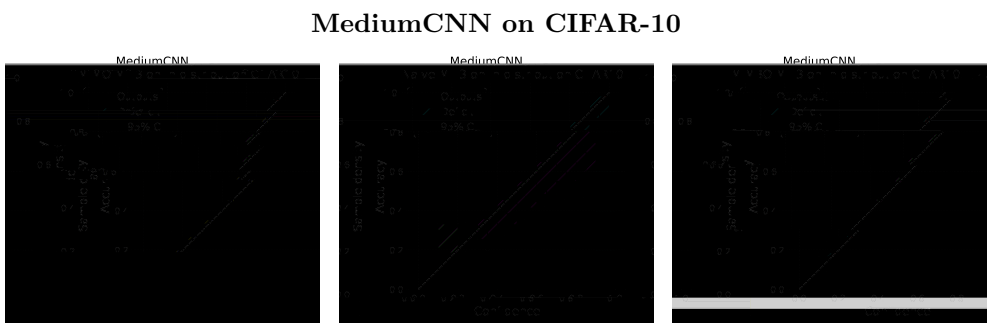
Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.390(24)	0.111(5)	7.08(75)	0.532(28)
MIMO $M = 2$	0.409(12)	0.0853(47)	2.79(83)	0.284(52)
MIMO $M = 3$	0.373(15)	0.0829(38)	1.99(13)	0.223(44)
MIMO $M = 4$	0.379(25)	0.0782(49)	1.75(12)	0.151(69)
MIMO $M = 5$	0.449(9)	**0.0684(8)**	1.53(3)	0.0627(258)
Naive $M = 2$	0.400(7)	0.112(1)	9.48(104)	0.542(8)
Naive $M = 3$	0.406(29)	0.0946(149)	3.65(140)	0.379(134)
Naive $M = 4$	0.400(21)	0.0967(119)	4.92(307)	0.405(102)
Naive $M = 5$	0.395(17)	0.108(13)	9.37(385)	0.501(105)
BNN	0.427(14)	0.102(3)	6.09(26)	0.483(13)
MIMBO $M = 2$	0.385(39)	0.0852(69)	2.23(25)	0.277(54)
MIMBO $M = 3$	0.393(18)	0.0780(33)	1.81(14)	0.178(42)
MIMBO $M = 4$	0.448(19)	0.0686(21)	**1.52(5)**	**0.0478(162)**
MIMBO $M = 5$	**0.469(11)**	0.0688(18)	1.56(7)	0.102(35)

Table 5.4. Results of MediumCNN models on the in-distribution dataset CIFAR-10 (left) and the out-of-distribution dataset CIFAR-10-C (right). Best results in each metric are bold. The value of M denotes the number of subnetworks in the model.

The CIFAR-10 in-distribution results for MediumCNN in Table 5.4(a) show that Naive $M = 3$ achieves the best accuracy and Brier score, being slightly more accurate than the Baseline. MIMO $M = 3$ and MIMBO $M = 3$ achieve the best NLL and ECE respectively. The difference in performance for MIMO $M = 3$ and MIMBO $M = 3$ is negligible, as the models are equally accurate and achieve similar results in all other metrics. The BNN model performs similarly to the Baseline in terms of accuracy, but achieves worse results on all metrics. In general, MIMBO models perform very similarly to the MIMO models with the same number of subnetworks. The non-subnetwork models, Baseline and BNN, have slightly higher accuracy than their MIMO-configured counterparts, MIMO and MIMBO. However, the MIMO-configured models perform better in metrics that evaluate uncertainty estimates, achieving lower Brier scores, NLL and ECE.

As a general trend, the accuracy of MIMO models and MIMBO models decreases as the number of subnetworks increases, with MIMO $M = 5$ and MIMBO $M = 5$ having the lowest accuracy. However, up to certain point, adding subnetworks improves calibration. For MediumCNN this point is at $M = 3$ for both MIMO and MIMBO as they achieve their best NLL and ECE at this point. This trend is not mirrored by the Naive models where there seems to be no consistent trend in the accuracy as Naive $M = 2$ is as accurate as Naive $M = 5$. However, the NLL of the Naive models is best at $M = 3$, which is the same as for MIMO and MIMBO.

The models with the highest in-distribution accuracy, i.e. Baseline, BNN and the Naive models, are generally more poorly calibrated than the MIMO and MIMBO models. To understand why, we look into the quality of the uncertainty estimates of MediumCNN models by interpreting their reliability diagrams. In figure 5.12 the reliability diagrams of Naive $M = 3$, MIMO $M = 3$ and MIMBO $M = 3$ are visualised, as they are the top performers in each metric. Thus, we can compare the accurate but comparatively poorly calibrated models with the less accurate but more well-calibrated ones.

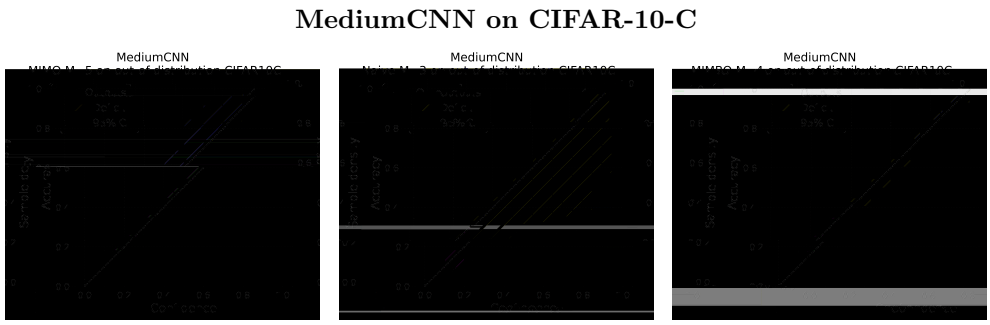


(a) MIMO $M = 3$ **(b)** Naive $M = 3$ **(c)** MIMBO $M = 3$

Figure 5.12. Reliability diagrams for the best performing models in each metric on the CIFAR-10 in-distribution data with the MediumCNN architecture. As the columns of most of the Naive model's bins are below their ideal height as indicated by the red deficit markings, it is overconfident. In contrast, the MIMO and MIMBO models are both well-calibrated as the each bin reaches close to their ideal height.

The similarity in calibration between MIMO and MIMBO measured by the ECE and NLL is reflected in the reliability plots in figure 5.12, as the two models are both well-calibrated and have nearly identical reliability diagrams. In contrast, the Naive model is overconfident and has high confidence in most of its predictions, as indicated by the high sample density in the bin for confidences in the range $[0.9; 1.0]$. This overconfidence appears to be an advantage with regards to maximising accuracy, but in return it hurts the calibration of the model.

The prediction accuracy of all MediumCNN models drops significantly when tested on out-of-distribution data. While the Baseline and Naive models has the best accuracy on the in-distribution data, the MIMO and MIMBO models perform the best on out-of-distribution data, as seen in table 5.4(b). In particular, MIMO $M = 5$ and MIMBO $M = 4, 5$ goes from being the least accurate on the in-distribution data to being both the most accurate and most well-calibrated on the out-of-distribution data, meaning they perform the best in all metrics. The BNN model now outperforms the Baseline in all metrics and while it cannot compete with the MIMO and MIMBO models on uncertainty estimates, it does present one of the five best accuracies. The Naive models have remarkably similar accuracies, however, they vary greatly in how well-calibrated they are, with Naive $M = 3, 4$ achieving better NLL and ECE than Naive $M = 2, 5$. While the best Naive models outperform the Baseline model, they are significantly worse than the best models on the out-of-distribution data, especially in terms of calibration.



(a) MIMO $M = 5$ **(b)** Naive $M = 3$ **(c)** MIMBO $M = 4$

Figure 5.13. The figure shows the reliability diagrams for the best calibrated MediumCNN MIMO, Naive and MIMBO models tested on CIFAR-10-C out-of-distribution data. The most accurate model MIMO $M = 5$ is underconfident as indicated by its columns being higher than the ideal line. In contrast, the Naive $M = 3$ model is very overconfident, with large deficits to the ideal. The MIMBO $M = 4$, which had the lowest ECE, is well-calibrated.

The reliability diagrams on figure 5.13(a) and 5.13(c) show that the MIMO $M = 5$ model is slightly overconfident, which brings its ECE score below that of MIMBO $M = 4$, which, makes predictions with well-calibrated uncertainty estimates. The Naive $M = 3$ model, which had the best accuracy on the in-distribution data, now displays even more overconfidence than on the in-distribution data, since the calibration deficits are larger in its reliability diagram in figure 5.13(b) than in figure 5.12(b). More than half of its predictions are made with confidence in the range $[0.9, 1.0]$, which results in poor calibration considering its accuracy of around 40%.

Wide ResNet results

We now look at the results of the Wide ResNet models on the CIFAR-10 and CIFAR-10-C datasets. Wide ResNet models use the architecture described in section 4.4.3 and are much deeper than the MediumCNN models. Note that the Wide ResNet Baseline model was trained using hyperparameters found in the literature to ensure that it is up to par with other Wide ResNets trained on CIFAR-10. Table 5.5 shows the results on the in-distribution dataset CIFAR-10 and the out-of-distribution CIFAR-10-C dataset with severity 5.

(a) Wide ResNet on CIFAR-10

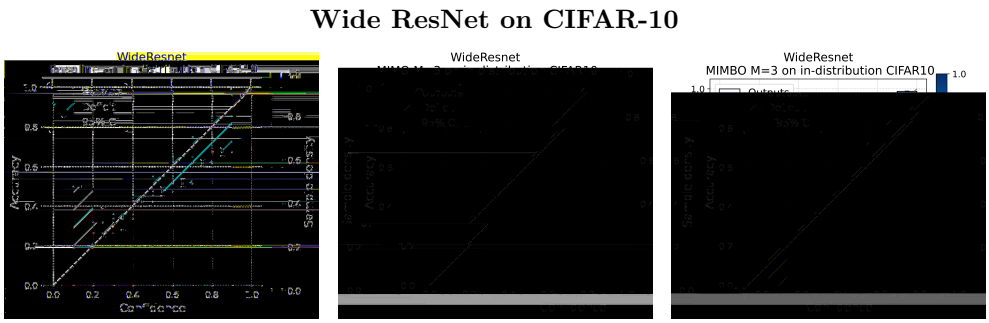
Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	**0.902**(2)	**0.0151**(**3**)	**0.346**(**5**)	0.0453(20)
MIMO $M = 2$	0.891(3)	0.0168(2)	0.402(12)	0.0354(37)
MIMO $M = 3$	0.885(2)	0.0170(2)	0.358(6)	**0.0150**(**18**)
MIMO $M = 4$	0.872(2)	0.0189(3)	0.387(4)	0.0223(35)
MIMO $M = 5$	0.846(8)	0.0229(13)	0.472(28)	0.0526(64)
Naive $M = 2$	0.892(2)	0.0175(2)	0.462(12)	0.0657(22)
Naive $M = 3$	0.862(5)	0.0212(6)	0.499(13)	0.0658(22)
Naive $M = 4$	0.851(4)	0.0227(5)	0.523(23)	0.0639(71)
Naive $M = 5$	0.890(1)	0.0177(2)	0.469(15)	0.0677(16)
BNN	0.892(2)	0.0175(3)	0.497(19)	0.0677(17)
MIMBO $M = 2$	0.889(1)	0.0168(2)	0.386(4)	0.0253(23)
MIMBO $M = 3$	0.872(6)	0.0188(7)	0.384(15)	0.0305(40)
MIMBO $M = 4$	0.854(4)	0.0226(4)	0.472(10)	0.0824(103)
MIMBO $M = 5$	0.825(4)	0.0275(8)	0.584(19)	0.122(12)

(b) Wide ResNet on CIFAR-10-C

Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.323(20)	0.108(5)	3.13(21)	0.461(28)
MIMO $M = 2$	0.348(31)	0.102(6)	3.54(32)	0.375(48)
MIMO $M = 3$	0.322(15)	0.102(4)	2.94(19)	0.364(36)
MIMO $M = 4$	0.337(33)	0.0944(60)	2.42(18)	0.309(58)
MIMO $M = 5$	0.317(13)	0.0917(23)	2.26(6)	0.278(29)
Naive $M = 2$	0.284(5)	0.113(4)	3.77(38)	0.486(36)
Naive $M = 3$	0.279(13)	0.109(3)	3.17(17)	0.437(24)
Naive $M = 4$	0.262(9)	0.113(4)	3.44(27)	0.464(33)
Naive $M = 5$	0.287(26)	0.114(9)	4.01(60)	0.493(60)
BNN	0.241(24)	0.134(5)	6.18(27)	0.637(32)
MIMBO $M = 2$	0.324(27)	0.102(6)	3.51(34)	0.414(38)
MIMBO $M = 3$	**0.351**(**21**)	0.0874(41)	2.19(12)	0.273(37)
MIMBO $M = 4$	0.310(27)	0.0884(32)	2.09(10)	0.272(29)
MIMBO $M = 5$	0.321(19)	**0.0839**(**29**)	**1.90**(**6**)	**0.221**(**34**)

Table 5.5. Results of Wide ResNet models on the in-distribution dataset CIFAR-10 (left) and the out-of-distribution dataset CIFAR-10-C (right). Best results in each metric are bold. The value of M denotes the number of subnetworks in the model.

With the Wide ResNet architecture on CIFAR-10, the Baseline model performs the best in all metrics except ECE, where MIMO $M = 3$ achieves the best result. The accuracy of the Baseline is above 90%, which is significantly more than what the best MediumCNN model achieves. In general, the deeper Wide ResNet models outperform their MediumCNN counterparts on all metrics. The BNN model achieves slightly worse results than the Baseline overall. The Naive models all perform worse than the Baseline model, likely in part because the Baseline uses optimised hyperparameters from the Wide ResNet paper by Zagoruyko et al. [67]. Like with the MediumCNN models, the accuracy of the MIMO and MIMBO models decreases as the number of subnetworks M increases. The NLL of the MIMO and MIMBO models is lowest at $M = 3$, meaning the models with $M = 3$ subnetworks have the best trade-off between accuracy and calibration. We investigate how this is reflected in the reliability diagrams for Baseline, MIMO $M = 3$ and MIMBO $M = 3$.

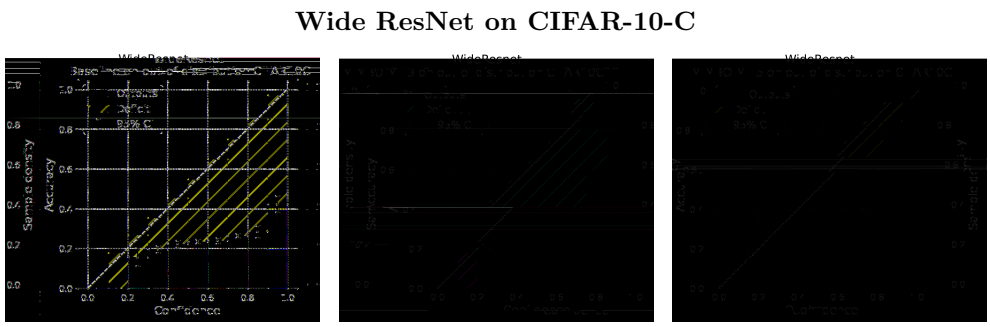


(a) Baseline **(b)** MIMO $M = 3$ **(c)** MIMBO $M = 3$

Figure 5.14. The figure shows the reliability diagrams for the Wide ResNet models that achieve the top 3 NLLs on the in-distribution data: Baseline, MIMO $M = 3$ and MIMBO $M = 3$. The baseline has the highest accuracy as seen in table 5.5(a) and has high confidence on most predictions. However, it is overconfident as evident from the noticable deficit to the ideal calibration. The more well-calibrated MIMO $M = 3$ and MIMBO $M = 3$ predict less samples with a confidence above 90% but are overall slightly less accurate than the Baseline.

With the deeper Wide ResNet architecture, not only does the accuracy increase, but also the number of samples that are predicted with higher confidence. This is the case for all the models with reliability diagrams featured in figure 5.14, but it is especially prominent for the Baseline model. The Baseline exhibits overconfidence because most of its bins are below their ideal height as indicated by the deficit. In contrast, the MIMBO $M = 3$ model is slightly underconfident as many bins are over the ideal height. The MIMO $M = 3$ model outperforms both the Baseline and MIMBO in terms of ECE and is almost perfectly calibrated, as the neglible deficits on its reliability diagram in figure 5.14(b) indicate. Overall, MIMO $M = 3$ appears to produce the most accurate uncertainty estimates on CIFAR-10. However, the improved calibration comes at the cost of a 1.7 percentage points decrease in test accuracy and a slightly higher NLL, compared to the Baseline.

When tested on the out-of-distribution CIFAR-10-C data, the accuracy of all Wide ResNet models drops greatly. The Baseline model, which on the in-distribution data produced the best results in accuracy, Brier score and NLL is now outperformed by the best MIMO and MIMBO models. The Wide ResNet MIMBO $M = 5$ model performs the best in all metrics except accuracy, where MIMBO $M = 3$ is best, as observed in Table 5.5(b). The BNN model performs the worst in all metrics, so it appears that the MIMBO models benefit greatly from the MIMO configuration. The MIMO models also perform well and are comparable to the MIMBO and Baseline models in accuracy. However, the MIMBO $M = 3, 4, 5$ models are more well-calibrated as indictated by the Brier score, NLL and ECE, where they outperform all MIMO models.



(a) Baseline **(b)** MIMBO $M = 3$ **(c)** MIMBO $M = 5$

Figure 5.15. The figure shows the reliability diagrams for the Baseline, MIMBO $M = 3$ and MIMBO $M = 5$ models, showing the change in calibration as subnetworks are added. The baseline still predicts most samples with high confidence, whereas the MIMBO models predict more samples with moderate confidence.

The reliability diagrams on figure 5.15 illustrate the overconfidence issues of models that are well-fitted on in-distribution data when predicting on out-of-distribution data. Figure 5.15(a) shows that the Baseline predicts the label of around half of the test samples with a confidence above 90%. While this was reasonable on the in-distribution data due to the high test accuracy, it results in poor calibration on the out-of-distribution data. The model with the best uncertainty estimates on the out-of-distribution data, MIMBO $M = 5$, predicts far less with high confidence, instead predicting with 30% to 50% confidence most often, as seen on figure 5.15(b). This alleviates the overconfidence, although MIMBO $M = 5$ is still clearly overconfident.

Comparison of MediumCNN and Wide ResNet on CIFAR-10

The results on CIFAR-10 reveal some interesting differences between the two architectures. MediumCNN models that have lower accuracies than the Wide ResNet models on the in-distribution data typically have higher accuracy than the Wide ResNet models on the out-of-distribution data. On out-of-distribution data, MediumCNN models reach 47% accuracy with MIMBO $M = 5$, compared to Wide ResNet models that peak at 35% with MIMBO $M = 3$. Moreover, models that are relatively inaccurate on the in-distribution data, e.g. MIMBO $M = 5$ for both MediumCNN and Wide ResNet, perform well on the out-of-distribution in terms of both accuracy and calibration. Overall, this implies that fitting closely to the in-distribution data can increase vulnerability to out-of-distribution data or data shift.

There is a large drop in the accuracy of all models from CIFAR-10 to CIFAR-10-C. To investigate why the difference is as large as it is, we can look into which classes the models tend to predict on the two datasets.

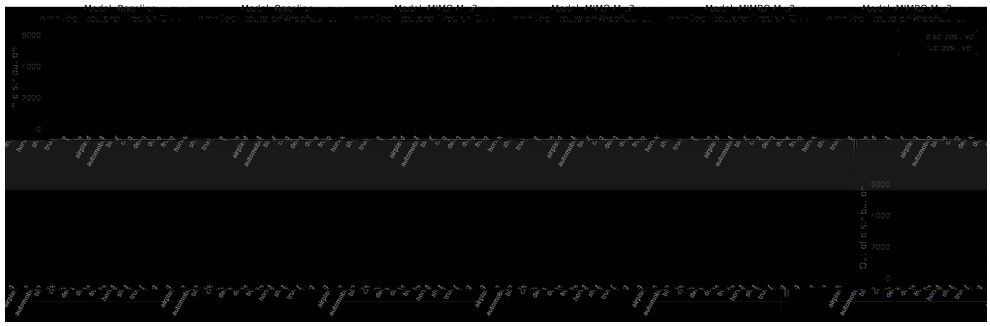


Figure 5.16. Class distribution of predicted labels on CIFAR10 and CIFAR-10-C for Baseline, MIMO $M = 3$ and MIMBO $M = 3$ with both MediumCNN and Wide ResNet architectures. The total height of each bar is the mean of the number of predictions for each class over 5 repetitions, and the error bars indicate the 95% confidence interval. The top row contain the results on CIFAR-10, the bottom row has the results on CIFAR-10-C. While the distribution is approximately uniform on the in-distribution data, the distribution of predictions on out-of-distribution data is much more skewed towards the "frog" label.

Figure 5.16 shows the distribution of predicted classes for the Baseline as well as MIMO $M = 3$ and MIMBO $M = 3$ on both CIFAR-10 and CIFAR-10-C. While all models predict classes with a nearly uniform distribution on the in-distribution data, the distribution shifts drastically on the out-of-distribution data. Most notably, most models predict a single label, "frog", for approximately half the test points. It appears that the models that fit the in-distribution data the best, such as the Wide ResNet Baseline and MIMO $M = 3$ model are the ones predicting "frog" the most often, while the models that were less well-fitted on the in-distribution tend to predict "frog" less often.

This is perhaps why we see that MIMBO models perform the best on the out-of-distribution data for both architectures. MIMO models with a corresponding number of subnetworks perform comparably or slightly worse. The MIMO and MIMBO models improve the calibration of the prediction by lowering the overall confidence for which predictions are made. This effect can be seen in Figure 5.15. From the results for the CIFAR-10 models, it appears that the MIMO-configuration alleviates some of the overconfidence issues of the models it is applied to. This effect appears to grow with the number of subnetworks. The reduced overconfidence comes at the cost of accuracy on the in-distribution data, as MIMO and MIMBO generally are less accurate than the Baseline and BNN models. However, the more uncertain MIMO and MIMBO models are more accurate on the out-of-distribution data. This trend is present in both the MediumCNN and Wide ResNet models. Observing how the class probabilities change as noise is added to the data gives us additional insight into how each model behaves in the in-distribution and out-of-distribution cases.

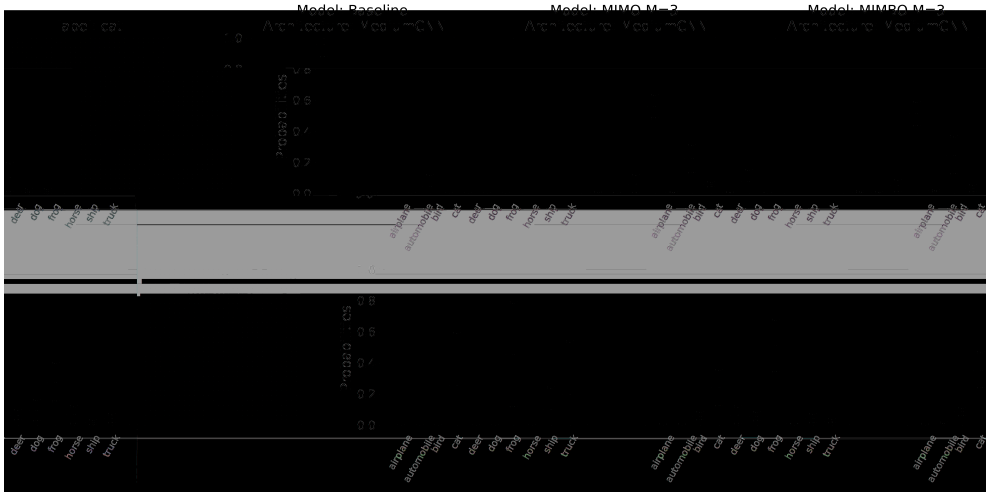


Figure 5.17. Predicted class probabilities on a test image from CIFAR-10 (top row) and CIFAR-10-C (bottom row) for MediumCNN Baseline, MIMO $M = 3$, MIMBO $M = 3$. The probabilities for the true class is highlighted in green. The shown probabilities are averages of 5 repetitions with error bars showing the 95% confidence intervals. The difference between Baseline and MIMO $M = 3$ shows the reduced certainty that comes with the MIMO-configuration while MIMBO $M = 3$ shows the additional uncertainty of having the network ensemble over both subnetworks and weight samples. The baseline tends to be confident in a few classes, while MIMO and MIMBO have more equally distributed confidences.

Figure 5.17 shows the predicted probabilities of three MediumCNN models on a test image from the in-distribution data and the out-of-distribution data. The specific image and models were chosen to exemplify how the MIMO configuration and Bayesian neural network properties affect the predicted class probabilities when predicting on in-distribution data and out-of-distribution data. From the example in figure 5.17 we observe that while the Baseline is correctly confident in its prediction on the in-distribution image, it is more confidently wrong than e.g. MIMBO $M = 3$, which shows more uncertainty as it assigns some probability to most classes. The predicted probabilities of the MIMO and MIMBO models on the in-distribution data are similar, as they assign the most probability to the correct "cat" class, followed by the "dog" class. On the out-of-distribution image we see that the "frog" label is among the two most probable classes for all three models despite being assigned almost no probability on the in-distribution image. This sudden spike in probability of the "frog" class reflects the overpredicting of the class in figure 5.16.

With the Wide ResNet architecture, MIMO and MIMBO are also more uncertain than the Baseline model, as seen in figure 5.18.

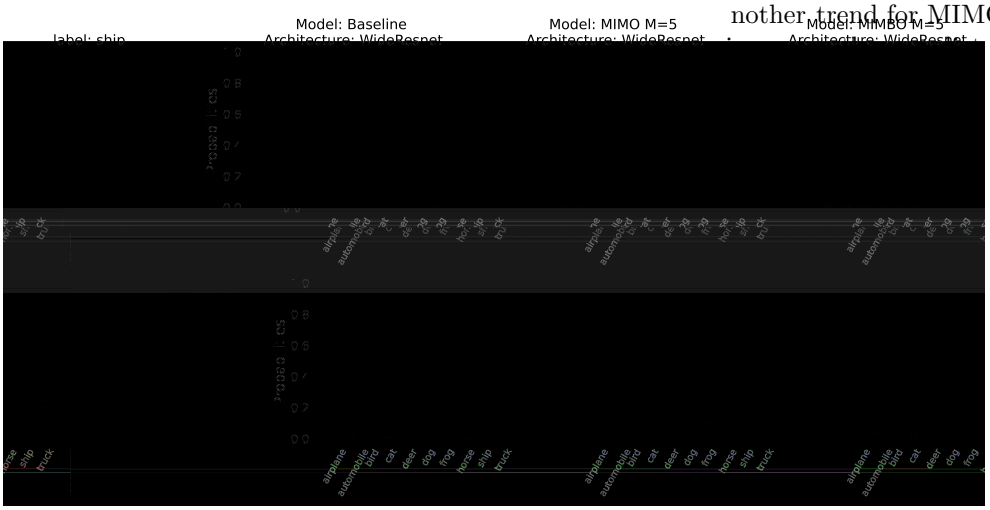


Figure 5.18. Predicted class probabilities on a test image from CIFAR-10 (top row) and CIFAR-10-C (bottom row) for Wide ResNet Baseline, MIMO $M = 5$, MIMBO $M = 5$. The probabilities for the true class is highlighted in green. The shown probabilities are averages of 5 repetitions with errors bars showing the 95% confidence intervals. The Baseline tends to be confident in a few classes, while MIMO and MIMBO have more equally distributed probabilities.

The probabilities on the out-of-distribution example are more uniformly distributed than for MediumCNN as seen on figure 5.18, which could be due to the increased number of subnetworks, the change in architecture, or because it is another example image. Since the top confidence is reduced for MIMO and MIMBO models in both architectures and examples, we can recognise the trend that MIMO and MIMBO models generally predict with reduced confidence. This property is also observable in the reliability diagrams in figures

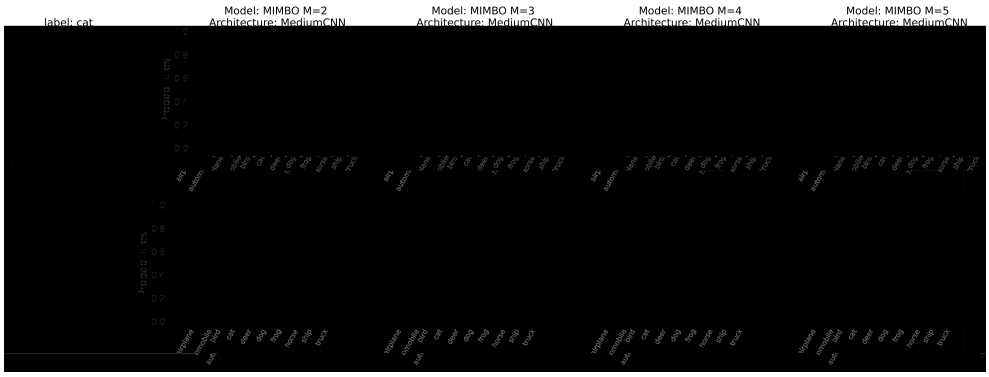


Figure 5.19. The plot shows MediumCNN MIMO models with different number of subnetworks predict on an example image from the CIFAR-10 test set. As the number of subnetworks decrease, the probability distributions become more "smooth", with the top probability losing probability while other classes gain.

In figure 5.19 we see how the MIMO models softmax probabilities change when the number of subnetworks in the model increases. The top probability becomes smaller while the probabilities of classes that at $M = 2$ had almost no probability gain enough to be visible on the plot. Plots for MIMBO models as well as Wide ResNet MIMO and MIMBO models can be found in the appendix F.

5.3.2 CIFAR-100

The in-distribution CIFAR-100 dataset and corresponding out-of-distribution dataset CIFAR-100-C are far more complex problems than the CIFAR-10 problems in the previous section due to the large increase to the number of classes. Therefore, the results will show how well the various models perform when trained and evaluated on a more task. The results for MediumCNN and Wide ResNet models on CIFAR-100 and CIFAR-100-C are presented in tables 5.6 and 5.7 respectively. Each table includes results on both the CIFAR-100 in-distribution data and the CIFAR-100-C out-of-distribution data for their respective architecture.

MediumCNN results

As with CIFAR-10, we start with the models using the MediumCNN architecture. Table 5.6 shows the results for MediumCNN on the CIFAR-100 and CIFAR-100-C dataset. The results for this architecture are worse than what would realistically be seen in other classification contexts, so the results will for these models will not be discussed beyond this section. They are however kept in the thesis for completeness.

(a) MediumCNN on CIFAR-100

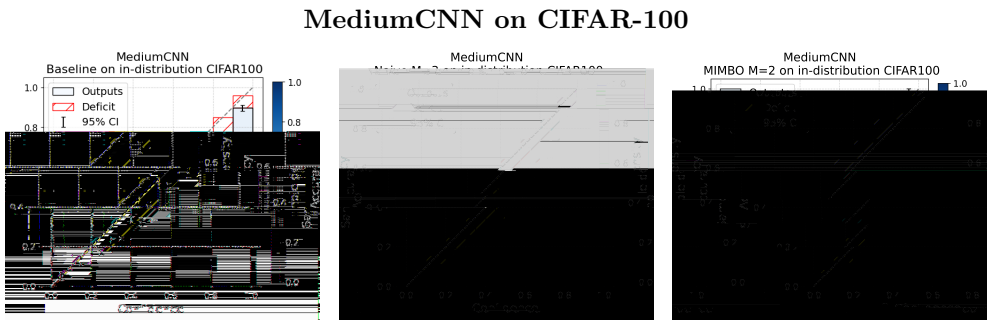
Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.338(8)	0.00809(10)	2.76(6)	0.101(16)
MIMO $M = 2$	0.301(16)	0.00832(19)	2.92(16)	0.0518(98)
MIMO $M = 3$	0.284(8)	0.00866(11)	3.12(11)	0.0901(361)
MIMO $M = 4$	0.235(99)	0.00895(43)	3.41(53)	0.109(48)
MIMO $M = 5$	0.273(15)	0.00907(8)	3.38(6)	0.165(10)
Naive $M = 2$	0.336(9)	0.00810(18)	2.77(12)	0.0972(383)
Naive $M = 3$	**0.353(11)**	**0.00791(22)**	**2.64(15)**	0.0864(487)
Naive $M = 4$	0.338(12)	0.00802(16)	2.71(10)	0.0785(264)
Naive $M = 5$	0.337(11)	0.00818(19)	2.83(15)	0.132(38)
BNN	0.343(10)	0.00808(13)	2.75(7)	0.114(22)
MIMBO $M = 2$	0.295(8)	0.00833(9)	2.90(7)	**0.0510(180)**
MIMBO $M = 3$	0.268(24)	0.00872(26)	3.18(25)	0.0842(186)
MIMBO $M = 4$	0.173(117)	0.00929(45)	3.80(58)	0.0889(637)
MIMBO $M = 5$	0.115(113)	0.00959(34)	4.11(53)	0.0701(748)

(b) MediumCNN on CIFAR-100-C severity 5

Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.0899(96)	0.0112(3)	5.80(25)	0.323(37)
MIMO $M = 2$	0.129(31)	0.00981(41)	4.32(51)	0.122(75)
MIMO $M = 3$	**0.145(30)**	**0.00944(9)**	**3.71(6)**	0.0708(167)
MIMO $M = 4$	0.104(42)	0.00958(15)	4.00(28)	0.0227(105)
MIMO $M = 5$	0.135(17)	0.00952(5)	3.84(7)	0.0435(222)
Naive $M = 2$	0.0929(230)	0.0109(8)	5.74(96)	0.276(84)
Naive $M = 3$	0.0823(88)	0.0114(5)	6.33(87)	0.335(49)
Naive $M = 4$	0.0894(152)	0.0108(4)	5.62(59)	0.265(47)
Naive $M = 5$	0.0916(27)	0.0112(5)	5.93(43)	0.330(54)
BNN	0.117(18)	0.0107(6)	5.20(74)	0.266(68)
MIMBO $M = 2$	0.0753(501)	0.00989(31)	4.39(37)	0.0891(808)
MIMBO $M = 3$	0.137(20)	0.00947(11)	3.80(14)	0.0516(201)
MIMBO $M = 4$	0.0897(575)	0.00963(20)	4.11(36)	**0.0211(150)**
MIMBO $M = 5$	0.0767(714)	0.00971(21)	4.24(40)	0.0381(411)

Table 5.6. Results of the MediumCNN models on the in-distribution dataset CIFAR-100 (left) and the out-of-distribution dataset CIFAR-100-C (right). Best results in each metric are bold. The value of M denotes the number of subnetworks in the model.

The difficulty of the CIFAR-100 dataset is clear from the results in table 5.6 where all models are far less accurate than they were on the CIFAR-10 dataset. Despite the overall decline in accuracy, the Naive $M = 3$ model remains the most accurate model of the MediumCNN models on CIFAR-100 and achieves the best Brier score and NLL, as seen in table 5.6(a). However, it only outperforms the Baseline slightly. The BNN and Baseline models perform very similarly in all metrics. While all MIMO and MIMBO models achieve worse accuracy and NLL than the Baseline and BNN, MIMBO $M = 2$ is better calibrated and achieves the best ECE, followed by MIMO $M = 2$. We can observe that increasing the number of subnetworks for MIMO and MIMBO models leads to worse performance on all metrics: the accuracy drops and the Brier score, NLL and ECE increase. For the Naive model, the metrics only change slightly as the number of subnetworks vary.

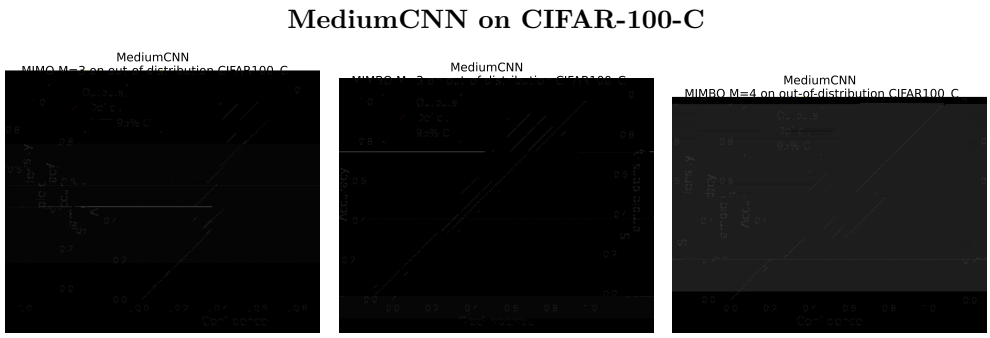


(a) Baseline **(b)** Naive $M = 3$ **(c)** MIMBO $M = 2$

Figure 5.20. The Figure shows the reliability diagrams of the two models with the lowest NLL, Baseline and Naive $M = 3$, along with the most well-calibrated model MIMBO $M = 2$ with MediumCNN architecture on CIFAR-100 in-distribution data. The Baseline model is not drastically overconfident, so $M = 2$ subnetworks are enough for MIMBO to be well-calibrated. The Naive model's diagram is almost identical to the Baseline and thus has the same calibration issues.

To understand how MIMO $M = 2$ and MIMBO $M = 2$ are more well-calibrated than the Naive and Baseline models while having larger NLLs, we can investigate their reliability diagrams. As evident from figure 5.20, the uncertainty estimates of the Baseline and Naive $M = 3$ are nearly identical, although the Naive model is slightly less overconfident. The model with the lowest ECE, MIMBO $M = 2$, does noticeably improve the calibration of the uncertainty estimates compared to the Baseline by seemingly predicting fewer samples with high confidence in the range $[0.9, 1.0]$, as illustrated by the lower sample density in that bin. This explains why it is less accurate and has higher NLL than the Baseline and Naive $M = 3$ models. However, it does become slightly underconfident in the high-confidence predictions while remaining overconfident in low-confidence predictions.

Like with CIFAR-10, we can see from Table 5.6(b) that the MediumCNN models are significantly less accurate on the out-of-distribution data compared to the in-distribution data. MIMO $M = 3$ achieves the best accuracy, Brier score and NLL but MIMBO $M = 3$ is comparable as it is less accurate but more well-calibrated. MIMBO $M = 4$ is the most well-calibrated as it has the lowest ECE with MIMO $M = 4$'s ECE being only slightly higher. The MIMO models are generally more accurate than the MIMBO model with the same number of subnetworks which means they achieve lower NLL, despite being less well-calibrated. This contrast between MIMO and MIMBO models is expressed in their reliability diagrams on figure 5.21.



(a) MIMO $M = 3$ **(b)** MIMBO $M = 3$ **(c)** MIMBO $M = 4$

Figure 5.21. Reliability diagrams for various models with MediumCNN architecture on CIFAR-100 in-distribution data. Since all models have low accuracy, the most well-calibrated model, MIMBO $M = 4$, predicts most samples with low confidence. While MIMBO $M = 4$ is the best calibrated according to ECE because the bin with most of the samples is at the ideal height, MIMO $M = 3$ and MIMBO $M = 3$ appear more well-calibrated.

The reliability diagrams of MIMO $M = 3$, MIMBO $M = 3$ and MIMBO $M = 4$ show that the models generally make predictions with low confidence. For MIMBO $M = 4$ in particular, the low confidence of its prediction matches its low accuracy, which results in the best ECE of all MediumCNN models on CIFAR-100-C, as seen in table 5.6(b). However, because MIMO $M = 3$ and MIMBO $M = 3$ are more accurate they score lower NLL than MIMBO $M = 4$. On the reliability diagrams this is expressed as MIMBO $M = 4$ having a larger share of its predictions with a confidence in the range $[0.0, 0.1]$.

Wide ResNet results

Table 5.7 show the results for Wide ResNet on the CIFAR-100 dataset and CIFAR-100-C dataset with severity 5.

(a) Wide ResNet on CIFAR-100

Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	**0.661(3)**	**0.00466(7)**	**1.34(2)**	0.0841(207)
MIMO $M = 2$	0.633(3)	0.00508(5)	1.56(4)	0.0887(92)
MIMO $M = 3$	0.605(5)	0.00522(6)	1.48(2)	**0.0283(57)**
MIMO $M = 4$	0.584(7)	0.00549(8)	1.54(3)	0.0577(55)
MIMO $M = 5$	0.531(6)	0.00622(2)	1.78(1)	0.113(10)
Naive $M = 2$	0.650(7)	0.00514(8)	1.57(2)	0.168(5)
Naive $M = 3$	0.541(14)	0.00627(16)	1.92(7)	0.146(8)
Naive $M = 4$	0.528(14)	0.00609(15)	1.76(5)	0.0667(125)
Naive $M = 5$	0.646(5)	0.00523(6)	1.63(2)	0.178(3)
BNN	0.644(4)	0.00537(4)	1.84(5)	0.191(5)
MIMBO $M = 2$	0.624(6)	0.00503(5)	1.45(3)	0.0381(44)
MIMBO $M = 3$	0.579(4)	0.00558(9)	1.57(3)	0.0692(156)
MIMBO $M = 4$	0.545(5)	0.00616(7)	1.76(2)	0.137(17)
MIMBO $M = 5$	0.478(24)	0.00704(16)	2.10(8)	0.176(18)

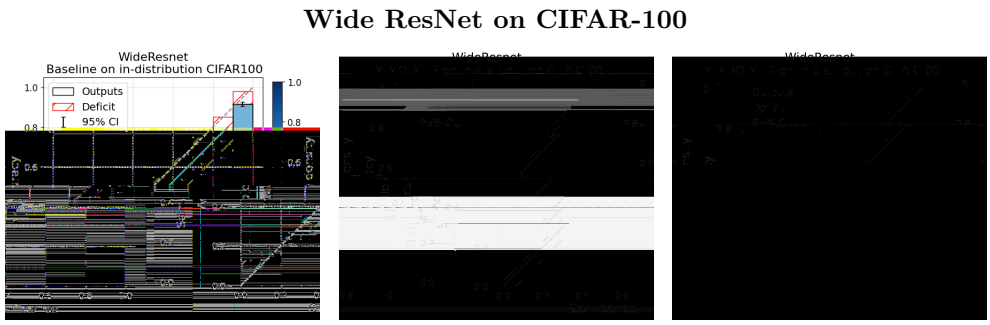
(b) Wide ResNet on CIFAR-100-C severity 5

Model	Accuracy ↑	Brier score ↓	NLL ↓	ECE ↓
Baseline	0.0709(21)	0.0116(5)	5.30(35)	0.351(60)
MIMO $M = 2$	0.0941(33)	0.0117(2)	7.76(49)	0.363(19)
MIMO $M = 3$	0.120(7)	0.0105(2)	5.87(29)	0.244(21)
MIMO $M = 4$	0.137(9)	0.00998(13)	4.97(20)	0.177(14)
MIMO $M = 5$	0.148(7)	0.00959(8)	4.28(15)	0.117(10)
Naive $M = 2$	0.0550(42)	0.0133(4)	9.20(71)	0.516(32)
Naive $M = 3$	0.0370(63)	0.0123(2)	6.90(37)	0.407(20)
Naive $M = 4$	0.0340(32)	0.0120(2)	6.89(14)	0.366(18)
Naive $M = 5$	0.0505(51)	0.0139(5)	9.64(64)	0.561(48)
BNN	0.0601(48)	0.0141(3)	10.67(89)	0.587(27)
MIMBO $M = 2$	0.101(7)	0.0111(1)	6.74(22)	0.314(11)
MIMBO $M = 3$	0.143(8)	0.00988(13)	4.86(21)	0.167(16)
MIMBO $M = 4$	0.153(19)	0.00954(18)	4.29(26)	0.104(19)
MIMBO $M = 5$	**0.169(15)**	**0.00928(13)**	**3.72(18)**	**0.0578(114)**

Table 5.7. Results of the Wide ResNet models on the in distribution dataset CIFAR-100 (left) and the out-of-distribution dataset CIFAR-100-C (right). Best result in each metric are bold. The value of M denotes the number of subnetworks in the model.

The Baseline is the best performing model on CIFAR-100 in every metric except ECE, where MIMO $M = 3$ achieves the best result. MIMO and MIMBO models provide improved calibration, as measured by the ECE, albeit at the cost of some accuracy. MIMO $M = 3$ is the most well-calibrated and achieves the best NLL among the MIMO models, despite dropping dropping 3 percentage points in accuracy compared to MIMO $M = 2$. This indicates that loss of accuracy is made up for by the improved calibration.

Other MIMO and MIMBO models are more well-calibrated than the Baseline. On the other hand, the BNN model is slightly less accurate than the Baseline and is significantly less well-calibrated. MIMBO $M = 2$ is an improvement to BNN in all metrics, but increasing the number of subnetworks only makes both accuracy and calibration worse. While the Wide ResNet Baseline model generally performs the best on CIFAR-100, MIMO and MIMBO are more well-calibrated.

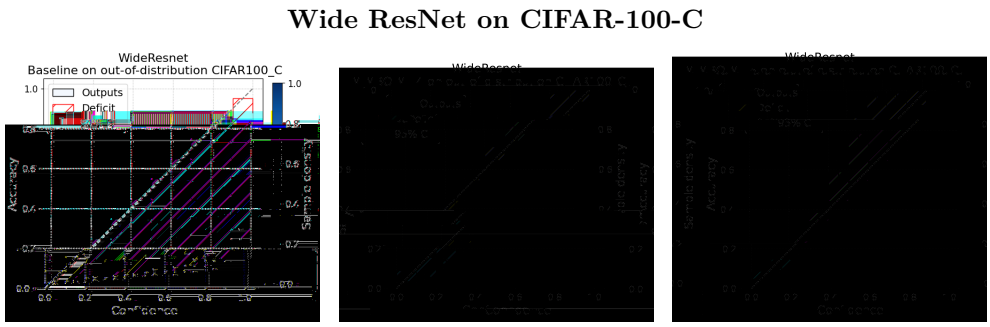


(a) Baseline **(b)** MIMO $M = 3$ **(c)** MIMBO $M = 2$

Figure 5.22. Reliability diagrams for the Baseline, MIMO $M = 3$ and MIMBO $M = 2$ models with Wide ResNet architecture on CIFAR-100 in-distribution data. The Baseline makes a majority of its predictions with very high confidence resulting in it being slightly overconfident. The MIMO and MIMBO models adjust the uncertainty estimates so that they better match the accuracy, resulting in better calibration.

From the reliability diagrams in figure 5.22 we observe that the Baseline is overconfident, similar to how it was on the CIFAR-10 dataset. Because it predicts with a confidence in the interval $[0.9, 1.0]$ for a majority of samples, its lack of accuracy on those samples make a large negative impact to the model's calibration. The more well-calibrated MIMO $M = 3$ model assigns fewer samples with a very large confidence, as shown in figure 5.22(b), thus matching its accuracy better. MIMBO $M = 2$ is still well-calibrated, but is slightly less so than MIMO $M = 3$.

The Wide ResNet results in Table 5.7(b) show that MIMBO $M = 5$ performs the best across all four metrics on out-of-distribution data. Other models with many subnetworks, such as MIMO $M = 5$ and MIMBO $M = 4$, are closest to MIMBO $M = 5$ in terms of predictive performance and calibration. The BNN model has higher accuracy than the Naive models, but has the worst calibration of any Wide ResNet model on CIFAR-100-C. We once again observe an inverse correlation between predictive performance on the in-distribution data and out-of-distribution data for all models, except the Naive models.



(a) Baseline **(b)** MIMBO $M = 4$ **(c)** MIMBO $M = 5$

Figure 5.23. The figure shows reliability diagrams for the Baseline, MIMBO $M = 4, 5$ models with Wide ResNet architecture on CIFAR-100-C out-of-distribution data. The shown models ECE decreases from left to right, meaning they become progressively more well-calibrated. The calibration of the MIMBO models improves as more prediction are made with a low confidence that better matches the models accuracy.

The reliability diagrams on figure 5.23 show that the Baseline is extremely overconfident. Even though it does not predict many samples with high confidence, it is still poorly calibrated due to its poor accuracy. MIMBO $M = 4$ and MIMBO $M = 5$ alleviate some of the overconfidence issues and predict most samples with an appropriate confidence. While the two MIMBO models remain overconfident in the low confidence bins, they become underconfident in the few samples that they predict with high confidence. Because so few samples are predicted with high confidence it does not impact the ECE much, as it is mostly impacted by bins with high sample density.

Comparison of MediumCNN and Wide ResNet on CIFAR-100

Like on CIFAR-10, observing how the distribution of predicted classes changes with the models and datasets explains the large drop in accuracy from CIFAR-100 to CIFAR-100-C.

Figure 5.24. The plot shows the distribution of predicted labels on CIFAR-10 (top row) and CIFAR-10-C (bottom row). The featured models are the Baseline, MIMO $M = 3$ and MIMBO $M = 3$ to give an understanding of how the prediction distribution changes with ensembling over bothweight samples and subnetworks. Each column presents the mean number of samples predicted with the respective label over 5 repetitions. 95% confidence intervals have been omitted for visual clarity.

From the distribution of predicted labels on figure 5.24 it appears that the Wide ResNet baseline, which is the most accurate on CIFAR-100, has a tendency to over-predict some classes on CIFAR-100-C. This tendency also appears with the MIMO $M = 3$ and MIMBO $M = 3$ Wide ResNet models, but to a lesser degree. For the MediumCNN models, this problem is not nearly as pronounced, as all the models with the MediumCNN architecture predict with a more uniform distribution on CIFAR-100-C.

The CIFAR-100 results exaggerate the trend of MIMO and MIMBO models being less confident in their predictions, as all models predict with overall lower confidence on CIFAR-100 and CIFAR-100-C, as can be seen in the reliability diagrams in figure 5.20, 5.21, 5.22 and 5.23.

The trend can also be observed for singular examples like in figure 5.25 and figure 5.26. These examples were chosen from the 10000 test images in CIFAR-100 and CIFAR-100-C to showcase how the behaviour of the models on the more complicated CIFAR-100 dataset compare to the behaviour of the models on the simpler CIFAR-10 dataset. We compare the Baseline, along with the MIMO and MIMBO models with the best NLL.

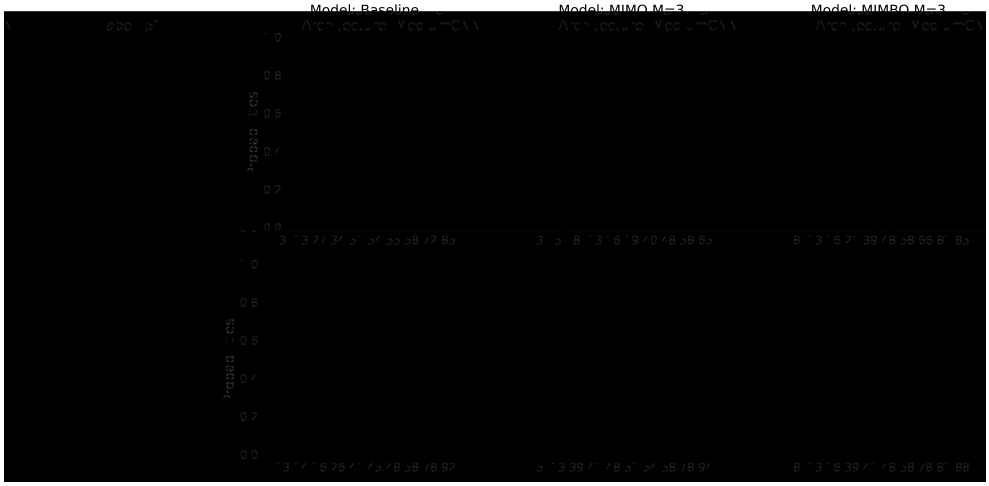


Figure 5.25. The figure shows how the MediumCNN Baseline, MIMO $M = 3$ and MIMBO $M = 3$ models assign probabilities to classes on an example image from the test sets of CIFAR-100 (top row) and CIFAR-100-C (bottom row). Only the probabilities of the 10 classes with the highest assigned probabilities are shown. The probability of the correct label is highlighted with green if plotted. The probabilities are means over 5 repetitions, with errors bars showing the 95% confidence intervals.

Whereas the MediumCNN Baseline model displays more confidence in its predictions on the in-distribution data than the MIMO and MIMBO models, the predicted confidences of the different models appear to be more similar on CIFAR-100 from the example in figure 5.25, perhaps due to the overall lower accuracy. The Baseline model becomes more confident in the out-of-distribution data than it was on the in-distribution data, while the confidences of MIMO and MIMBO appear lower and more evenly distributed on both the in-distribution and out-of-distribution data.

In the example on figure 5.26 the confidence of the Baseline model is greater than the confidence of the MIMO $M = 5$ and MIMBO $M = 3$ models, mirroring what we saw on the CIFAR-10 data. This is also the case for the out-of-distribution data. Thus it appears that MIMO and MIMBO still reduce the confidence and makes the confidence distribution more uniform in some cases.

Figure 5.26. The figure shows how the Wide ResNet Baseline, MIMO $M = 5$ and MIMBO $M = 5$ models assign probabilities to classes on an example image from the test sets of CIFAR-100 (top row) and CIFAR-100-C (bottom row). Only the probabilities of the 10 classes with the highest assigned probabilities are shown. The probability of the correct label is highlighted with green if plotted. The probabilities are means over 5 repetitions, with errors bars showing the 95% confidence intervals.

Overall, it seems that MIMO and MIMBO are more uniform in their prediction probabilities than the Baseline, although the results are less conclusive on CIFAR-100 than on CIFAR-10. Despite the results being less clear than CIFAR-10, we can still see the smoothing of the probability distribution when the number of subnetworks is increased.

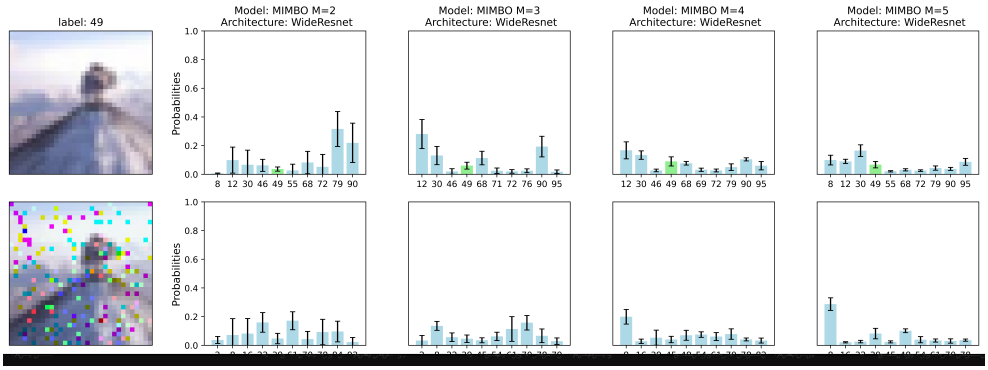


Figure 5.27. The figure shows MIMBO models an with increasing number of subnetworks making predictions on the sample example image from CIFAR-100 and CIFAR-100-C. The probability distribution for CIFAR-100 appears to smoothen as the number of subnetworks increases, while the top class seems to be assigned more probability for CIFAR-100-C, although the other top-10 classes are diminished.

As we see on figure 5.27, the probability distribution of the MIMBO models becomes more smooth as subnetworks are added. For this example, the effect is clear on the in-distribution data, while the top class seems to become more probable on the out-of-distribution data. We should however keep in mind that the figure only shows the top 10 largest probabilities, so we can only speculate as to how smooth the distribution would look if all 100 classes were shown.

5.3.3 Summary of classification results

The results on both the simpler CIFAR-10, the more complex CIFAR-100 datasets and their corresponding out-of-distribution variants point to some similar conclusions. On the in-distribution datasets, the Baseline or Naive models usually performs the best with regards to accuracy and NLL. The Baseline has the highest accuracy on both in-distribution datasets for the models using the Wide ResNet architecture, while Naive $M = 3$ is best with the MediumCNN architecture on both datasets. From their reliability diagrams in figures 5.15(a), 5.20(a), 5.22(a), 5.23(a), the Baselines and Naive models are clearly varying degrees of overconfident. While the overconfidence is noticeable on the in-distribution datasets, like on figure 5.20(b) and figure 5.22(a), it becomes more pronounced on the out-of-distribution datasets, like on figure 5.13(b) and figure 5.23(a). MIMO and MIMBO models are the most well-calibrated as they achieve the lowest ECEs, but since the better calibration comes at the cost of some accuracy, they do not necessarily outperform the most accurate models in NLL. With regards to optimising NLL, there is a trade-off between accuracy and calibration, and what exact model-configuration has the best combination depends on the model

architecture and the dataset it is tested on. The reliability diagrams, such as figure 5.12 5.22, show that both the MIMO and MIMBO models predict with less probability assigned to the most likely class than the Baseline and Naive models.

For both CIFAR-10-C and CIFAR-100-C, the distribution shift in the data make the models erroneously predict certain classes too often, as displayed in figures 5.16 and 5.24. This occurs to a lesser degree for MediumCNN models than Wide ResNet models.

5.4 Diversity of subnetworks

The results on CIFAR-10 and CIFAR-100 in the previous sections indicate that the subnetworks in the Naive models behave differently than the subnetworks in the MIMO and MIMBO models. Unlike the MIMO and MIMBO models, changing the number of subnetworks for the Naive models has no clear trend in regards to increasing or decreasing the performance on in-distribution data, as observed in tables 5.4(a)-5.7(a).

We visualise the optimisation trajectory of each subnetwork in a 2D PCA space on figure 5.28.

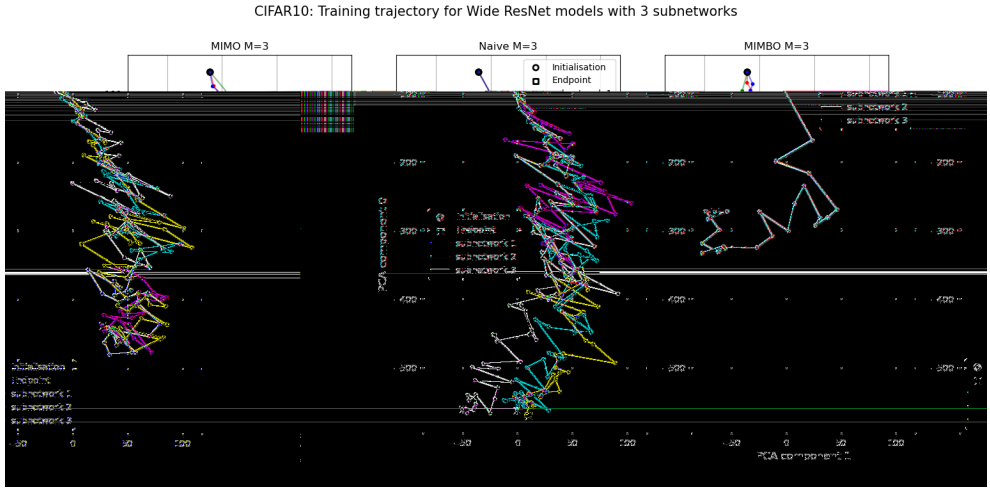


Figure 5.28. Optimisation trajectory for Wide ResNet models MIMO, Naive, and MIMBO with $M = 3$ subnetworks in the PCA space spanned by the PCA component 0 and 1. Each dot represents the log-probabilities for a batch of the same validation images for a subnetwork at one specific epoch during the training process. MIMO $M = 3$ and MIMBO $M = 3$ converge to different local optima in the function space, while the subnetworks for Naive $M = 3$ converge to the same optima.

Figure 5.28 shows that the subnetworks for MIMO $M = 3$ and MIMBO $M = 3$ converge to different local optima in the function space, visualised in a 2D PCA space. However, the subnetworks for the Naive $M = 3$ model follow the exact same trajectory and converge to the same optimum in the PCA space.

We compute the disagreement and average divergence in accordance with equations (4.19) and (4.20) for each observation in the first batch of the validation set, across 5 repetitions.

(a) MediumCNN on CIFAR-10

	$D_{\text{Disagreement}}$	D_{KL}
MIMO $M = 2$	0.442(245)	0.109(52)
MIMO $M = 3$	0.349(41)	0.074(17)
MIMO $M = 4$	0.396(50)	0.057(9)
MIMO $M = 5$	0.554(50)	0.082(20)
Naive $M = 2$	0.016(8)	0.001(0)
Naive $M = 3$	0.005(2)	0.000(0)
Naive $M = 4$	0.017(4)	0.001(0)
Naive $M = 5$	0.016(5)	0.001(0)
MIMBO $M = 2$	0.357(21)	0.148(30)
MIMBO $M = 3$	0.369(24)	0.90(16)
MIMBO $M = 4$	0.470(14)	0.089(14)
MIMBO $M = 5$	0.647(108)	0.125(56)

(b) Wide ResNet on CIFAR-10

	$D_{\text{Disagreement}}$	D_{KL}
MIMO $M = 2$	0.092(8)	0.037(5)
MIMO $M = 3$	0.131(12)	0.042(5)
MIMO $M = 4$	0.137(13)	0.033(3)
MIMO $M = 5$	0.184(15)	0.038(3)
Naive $M = 2$	0.002(2)	0.000(0)
Naive $M = 3$	0.000(0)	0.000(0)
Naive $M = 4$	0.000(0)	0.000(0)
Naive $M = 5$	0.000(0)	0.000(0)
MIMBO $M = 2$	0.111(11)	0.044(4)
MIMBO $M = 3$	0.141(10)	0.041(4)
MIMBO $M = 4$	0.200(20)	0.051(13)
MIMBO $M = 5$	0.263(33)	0.068(14)

Table 5.8. Similarity measures for subnetworks in MediumCNN models (left) and Wide ResNet models (right) on CIFAR-10 averaged over 5 repetitions with 95% confidence intervals. The similarity metrics are computed for the endpoints of the optimisation trajectory for each subnetwork. For both the MediumCNN and Wide ResNet, the metrics show that the average disagreement and KL-divergence of the subnetworks in the Naive models is 0 or close to 0 (up to three decimals), indicating little to no diversity. In comparison, the subnetworks in the MIMO and MIMBO models disagree far more often and therefore have greater diversity.

Table 5.8 displays the similarity measures for the subnetworks in the MIMO, Naive, and MIMBO models for $M = 2, 3, 4, 5$. The disagreement and KL-divergence for MediumCNN in table 5.8(a) is much larger for the MIMO and MIMBO models. The low disagreement and KL-divergence for the Naive models indicate that the predictions and output distributions for each subnetwork are very similar. This is in stark contrast to the MIMO and MIMBO models whose subnetworks are far more diverse.

The contrast is even greater for the Wide ResNet models, as observed in table 5.8(b). For all Naive models, the average disagreement and KL-divergence among the subnetworks is effectively 0.

CHAPTER 6

Discussion

Of our many results, we believe that the most successful experiment was for the classification tasks on CIFAR-10 and CIFAR-10-C. On this data our models managed to achieve high enough accuracy for the results to be useful for appraising whether the models uncertainty estimates are useful. This is also the case for our Wide ResNet CIFAR-100 and CIFAR-100-C results. Therefore, our discussion will revolve primarily around the results of these experiments.

6.1 Regarding the uncertainty estimates of subnetwork ensembles

On the classification tasks the Baseline models are overconfident on in-distribution data, as can be seen on figures 5.14(a) and 5.22(a). The overconfidence is more pronounced when they are tested on out-of-distribution data, as seen on figures 5.15(a) and 5.23(a). Similar to the Baseline, our results indicate that the BNN suffers from overconfidence issues.

On the other hand, the subnetwork ensemble models MIMO and MIMBO provide much more well-calibrated uncertainty estimates on both in-distribution and out-of-distribution data. Compared to the Baseline models, they are always less confident in the class they assign the most confidence to. When less confidence is assigned to the class with most confidence, the confidence must be redistributed to the other classes, since the distribution must sum to 1, which effectively smoothes out the predictive output distribution, as exemplified in figure 5.17. This means that when the Baseline is overconfident, the MIMO and MIMBO models reduce the overconfidence and produce well-calibrated uncertainy estimates. But if the Baseline is already well-calibrated, like on the in-distribution regression datasets, MIMO and MIMBO would become underconfident, resulting in worse calibrated uncertainty estimates. However, 5.17Of our r iUt e and

with both MediumCNN and Wide ResNet architectures, as illustrated in figures 5.12 and 5.14. It is especially evident for the Wide ResNet architecture that the MIMBO model becomes more underconfident and reduces the sample density for predictions with confidence in the range $[0.9; 1.0]$, when compared to the MIMO model with the same number of subnetworks.

This can be explained by the fact that MIMBO, in addition to averaging over its subnetworks, also averages over multiple weight samples, as it has the properties of a Bayesian neural network. In section 5.1, we investigated how the model averaging in Bayesian neural networks affect both prediction accuracy and uncertainty estimates. In figure 5.1, we observe how the ECE and NLL decreases while accuracy increases as the number of samples increases from 1 to 8 for MIMBO $M = 2, 3$, demonstrating how the MIMBO models benefit from averaging over models with sampled weights. However, as the results for MIMBO $M = 4, 5$ demonstrate, averaging over sampled model weights can lead to a worse ECE and negatively affect the calibration.

Furthermore, the results indicate that a MIMBO model is less uncertain than a MIMO model with an additional subnetwork, e.g MIMBO $M = 4$ is less uncertain than MIMO $M = 5$, as exemplified on figure 5.13 and 5.22. This shows that adding a subnetwork often provides more smoothing of the predictive output distribution than being Bayesian does. In the cases where the Baseline model is very overconfident, such as Wide ResNet on CIFAR-100-C, MIMBO $M = 5$ perform

Our results for both MediumCNN and Wide ResNet show a trend where the diversity of the MIMBO models is greater than the corresponding MIMO model with the same number of subnetworks. The trend does not quite hold in all cases, for instance the $D_{\text{Disagreement}}$ for MediumCNN MIMO $M = 2$ is larger than for MediumCNN MIMBO $M = 2$. This may be because the similarity measures are computed on only one batch from the validation set, and not the entire test set. Furthermore, the results for the Wide ResNet models show a trend where the disagreement, and the KL-divergence to a lesser extent, tends to increase as the number of subnetworks increases. It is possible that some the increased diversity is due to insufficient capacity.

It is apparent that this has an effect on the predictive output distributions of MIMO and MIMBO. As exemplified by figure 5.19 and 5.27, the output distribution of MIMBO with the Wide ResNet architecture becomes more uniform as the number of subnetworks increases from 2 to 5 on both in-distribution and out-of-distribution data. The models redistribute the confidence of the most certain class to the remaining classes. The reliability diagrams seem to suggest that this also applies to the entire test set in general. As seen on figures 5.12 and 5.22, the sample density of the bin for $[0.9; 1.0]$ is significantly lower for MIMO and MIMBO than Baseline and Naive, which means that the confidence of the most certain class is redistributed to the other classes when adding subnetworks.

6.3 The accuracy-calibration tradeoff

Although the MIMO and MIMBO models generally are more well-calibrated than the Baseline, they are usually also less accurate on in-distribution data. We observe that just as the predictive probabilities are increasingly smoothed with more subnetworks, the accuracy of the MIMO and MIMBO models on in-distribution data decreases. This means that there is a trade-off between accuracy and calibration, where, to choose the optimal model for a certain problem, one needs to consider how much of a loss in accuracy one can tolerate to improve the calibration. The NLL already weights both accuracy and calibration, so choosing the optimal model could be as simple as selecting the model with the lowest NLL.

There are multiple factors that could negatively influence the in-distribution accuracy of MIMO and MIMBO models. First is the reduction of confidence in the predicted class. As the predictive probabilities are smoothed, one could imagine how samples which were previously predicted correctly with low confidence could have the correct class' probability reduced so much that it would no longer be predicted. Another factor to consider is how well-fitted each subnetwork in the MIMO or MIMBO model is. Havasi et al. [19] show that after a certain number of subnetworks the accuracy of each subnetwork along with the accuracy of the ensemble would decrease for each additional subnetwork. We observed during training (see appendix G) that each individual subnetwork in the Wide ResNet MIMO and MIMBO models with more than

$M = 3$ subnetworks were unable to fit the training data properly. As additional subnetworks are added, the individual subnetwork becomes less fitted on the training data. This indicates that having more than $M = 3$ subnetworks causes each subnetwork to have insufficient capacity to fit the data as well as the Baseline, which would cause the MIMO or MIMBO model to be less accurate. This is in line with the findings of Havasi et al.

Unlike Havasi et al. who find that Wide Resnet MIMO networks with $M = 2, 3$ have higher accuracy and better calibration than the Baseline on the in-distribution test set, we consistently find that MIMO and MIMBO networks are less accurate, but more well-calibrated. This difference in findings could be caused by the difference in how the models are trained. Our Baseline is trained with parameters from Zagoruyko et al. [67], unlike the remaining models which had their hyperparameters determined with the same method. These differences could be why the MIMO $M = 2, 3$ and perhaps MIMBO $M = 2, 3$ models are less accurate and thus have higher NLL than the Baseline despite being more well-calibrated. Moreover, Havasi et al. train their MIMO model with batch repetition, which appears to boost its accuracy to be better than a deterministic neural network. If we had used batch repetition, we may have had MIMO models with accuracy comparable to our Baseline.

MIMBO, along with MIMO, excel on the out-of-distribution data. Since both the MediumCNN and Wide ResNet Baseline is very overconfident on CIFAR-10-C, the MIMBO models have an opportunity to excel, which they do by being the most accurate, most well-calibrated and overall achieving the best NLL. So in cases where a deterministic neural network would be massively overconfident, MIMBO models perform well. The success of the MIMBO is likely amplified by the difficulty of the chosen out-of-distribution data. The difficulty can be gauged by the large performance drop in all models. A lesser severity or a different corruption type might yield less conclusive results for MIMBO.

6.4 Future work

There are several aspects that we could improve on in the future.

A major advantage of MIMO is its computational efficiency at inference, requiring only one forward pass. Making the MIMO model Bayesian, i.e. MIMBO, with the current methods results in doubling the number of parameters in the model, which requires more memory and computational power to train and evaluate [76]. Achieving comparable performance to MIMO with MIMBO also requires multiple forward passes at inference. One aspect that could be studied further is to look into more parameter efficient Bayesian neural networks, for instance Rank-1 BNN by Dusenberry et al. [15] which also achieves well-calibrated uncertainty estimates.

The current out-of-distribution results for the CIFAR-10 and CIFAR-100 datasets are computed for just one type of corruption with the highest degree of corruption severity, resulting in a harsh dataset shift from the in-distribution data. While the MIMBO models with a large number of subnetworks outperform the other models and achieve the most well-calibrated uncertainty estimates in this scenario, it would be worthwhile to evaluate the uncertainty estimates of the models across all corruptions and severities. By comparing the models on out-of-distribution with different degrees of severity, we can establish if the MIMBO model also outperforms the other models when the dataset shift is less severe.

A simple way of improving the accuracy of our models would be to use data augmentation. In the Wide-ResNet paper, Zagoryuko et al. [67] achieve higher accuracy on CIFAR-10 with a Wide ResNet 28-10 model. It would most likely allow our Baseline to achieve comparable results and would likely increase the accuracy of our other models as well. Using batch repetition as introduced by Havasi et al. [19] would also likely slightly improve the accuracy of our MIMO and MIMBO models.

6.5 Conclusion

In this thesis, we introduce the novel MIMBO neural network for the purpose of improving uncertainty estimates of neural networks. We have implemented the Bayesian neural network, MIMO neural network and our MIMBO neural network, and all three models have been evaluated on regression and classification tasks in terms of predictive performance and uncertainty estimation.

The results of our experiments show that MIMBO models produce more well-calibrated uncertainty estimates than a deterministic neural network baseline and Bayesian neural networks, although this is at the cost of some predictive performance. Furthermore, MIMBO achieves comparable uncertainty estimates to MIMO on the CIFAR-10 and CIFAR-100 datasets, and achieves more favourable uncertainty estimates than MIMO on the out-of-distribution datasets CIFAR-10-C and CIFAR-100-C at severity 5.

Subnetwork ensemble models, such as MIMO and MIMBO, are less accurate than the Baseline models. The reduced accuracy correlates with the increase in diversity and uncertainty. However, methods such as data augmentation and batch repetition have the potential to increase accuracy without affecting subnetwork diversity.

In conclusion, MIMBO models trade-off some predictive performance for improved uncertainty estimates that are on par with MIMO and more well-calibrated than deterministic neural networks and Bayesian neural networks.

Appendices

Appendix A: Weighting schemes for the KL-divergence term

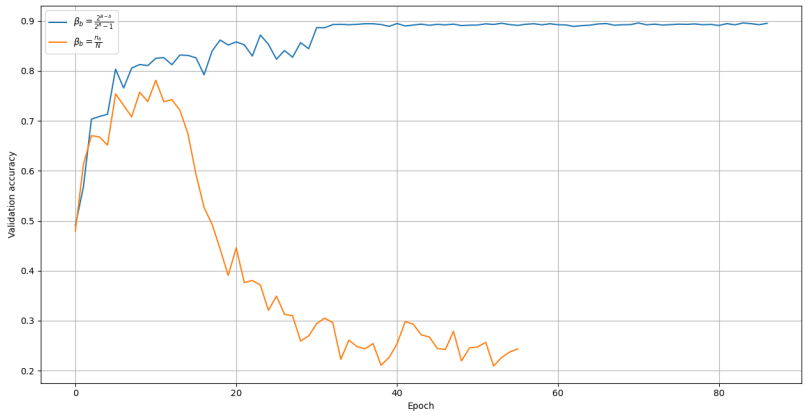


Figure 1. Validation accuracy for two Bayesian neural networks trained on a classification task, The networks utilise different weighting schemes for the KL-divergence term in the loss function. The weighting scheme $\beta_b = \frac{n_b}{N}$ makes the validation accuracy crash, while the weighting scheme $\beta_b = \frac{2^{B-b}}{2^B-1}$ makes ensures that the network trains until convergence.

Appendix B: Deriving the mixture model for regression

Deriving the mixture variance

In the regression task, the model predicts a mean μ and standard deviation σ for an input x. The mean and standard deviation are parameters in a Gaussian distribution, and the M predictions can be aggregated into a Gaussian mixture with M mixture components on the form [26]:

$$f_M(x) = \sum_m^M \pi_m \mathcal{N}(x|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m^2)$$

where π_m is the mixture weight. We first derive the first and second order moments of the Gaussian mixture.

$$\begin{aligned}
\mathbb{E}_{f_M}[x] &= \int x f_M(x)\,\mathrm{d}x \\
&= \int x \sum_m^M \pi_m \mathcal{N}(x|\mu_m, \sigma_m^2)\,\mathrm{d}x \\
&= \sum_m^M \pi_m \int x \mathcal{N}(x|\mu_m, \sigma_m^2)\,\mathrm{d}x \\
&= \sum_m^M \pi_m \mathbb{E}_{x\sim\mathcal{N}(x|\mu_m,\sigma_m^2)}[x] \\
&= \sum_m^M \pi_m \mu_m \\
\mathbb{E}_{f_M}[x^2] &= \int x^2 f_M(x)\,\mathrm{d}x \\
&= \int x^2 \sum_m^M \pi_m \mathcal{N}(x|\mu_m, \sigma_m^2)\,\mathrm{d}x \\
&= \sum_m^M \pi_m \int x^2 \mathcal{N}(x|\mu_m, \sigma_m^2)\,\mathrm{d}x \\
&= \sum_m^M \pi_m \mathbb{E}_{x\sim\mathcal{N}(x|\mu_m,\sigma_m)}[x^2] \\
&= \sum_m^M \pi_m (\mu_m^2 + \sigma_m^2)
\end{aligned}$$

The mean and variance of the Gaussian mixture with M mixture components can then be computed from the first and second order moments:

$$\begin{aligned}
\mu_* &= \mathbb{E}_{f_M}[x] = \sum_m^M \pi_m \mu_m \\
\sigma_*^2 &= \mathbb{E}_{f_M}[x^2] - \mathbb{E}_{f_M}[x]^2 = \sum_m^M \pi_m (\mu_m^2 + \sigma_m^2) - \mu_*^2 \Rightarrow \\
\sigma_* &= \sqrt{\left(\sum_m^M \pi_m (\mu_m^2 + \sigma_m^2) - \mu_*^2\right)}
\end{aligned}$$

For all our experiments, we choose to weight the mixture components evenly, such that $\pi_1 = \ldots = \pi_M = \frac{1}{M}$.

Decomposition into aleatoric and epistemic uncertainty

The mixture variance can be further decomposed into aleatoric and epistemic uncertainty [77]:

$$\begin{aligned}
\sigma_*^2 &= \frac{1}{M}\sum_m^M (\mu_m^2 + \sigma_m^2) - \mu_*^2 \\
&= \frac{1}{M}\sum_m^M \sigma_m^2 + \frac{1}{M}\sum_m^M \mu_m^2 - \mu_*^2 \\
&= \mathbb{E}[\sigma_m^2] + \mathbb{E}[\mu_m^2] - \mathbb{E}[\mu_m]^2 \\
&= \underbrace{\mathbb{E}[\sigma_m^2]}_{\text{Aleatoric uncertainty}} + \underbrace{\text{Var}[\mu_m]}_{\text{Epistemic uncertainty}}
\end{aligned}$$

From this expression, we see that the expectation of the predicted variance is the aleatoric uncertainty, while the variance of the predictions predicted mean is the epistemic uncertainty.

Appendix C: Proposition 1

The following mathematical proof is taken directly from the paper "Weight uncertainty in Neural Networks" by Blundell et al. [16]:

Proposition 1. *Let ϵ be a random variable having a probability density given by $q(\epsilon)$ and let $\mathbf{w} = t(\theta, \epsilon)$ where $t(\theta, \epsilon)$ is a deterministic function. Suppose further that the marginal probability density of $\mathbf{w}$, $q(\mathbf{w}|\theta))$, is such that $q(\epsilon)d\epsilon = q(\mathbf{w}|\theta)d\mathbf{w}$. Then for a function f with derivatives in $\mathbf{w}$:*

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w},\theta) = \mathbb{E}_{q(\ )}\left[\frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \theta} + \frac{\partial f(\mathbf{w},\theta)}{\partial \theta}\right]$$

Proof.

$$\begin{aligned}
\frac{\partial}{\partial\theta}\mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w},\theta)] &= \frac{\partial}{\partial\theta}\int f(\mathbf{w},\theta)q(\mathbf{w}|\theta)\mathrm{d}\mathbf{w} \\
&= \frac{\partial}{\partial\theta}\int f(\mathbf{w},\theta)q(\epsilon)\mathrm{d}\epsilon \\
&= \mathbb{E}_{q(\ )}\left[\frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \theta} + \frac{\partial f(\mathbf{w},\theta)}{\partial \theta}\right]
\end{aligned}$$

□

Appendix D: Showing that optimal variance is equal to MSE

Using the variance as a measure of uncertainty can be justified by showing that the probability density of a prediction is maximised when the variance σ^2 is equal to the squared error $(y - \mu)^2 = \text{SE}$. Starting from Gaussian probability density function

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{\text{SE}}{\sigma^2}\right\}$$

We take the derivative with respect to σ and set it equal to 0 to obtain:

$$\frac{\mathrm{d}f(y)}{\mathrm{d}\sigma} = -\frac{\sqrt{2}\exp\{-\frac{\text{SE}}{\sigma^2}\}}{2\sigma^2\sqrt{\pi}} + \frac{\sqrt{2}\text{SE}\exp\{-\frac{\text{SE}}{2\sigma^2}\}}{2\sigma^4\sqrt{\pi}} = 0$$

Solving this expression yields:

$$\sigma = \sqrt{\text{SE}}$$
$$\sigma^2 = \text{SE}$$

the optimal variance for maximising the probability density is equal to the squared error. When averaging over a bin this means that the average variance is equal to the mean squared error.

Appendix E: Hyperparameter sweep tables

Regression models

We have three regression datasets, the one-dimensional and multidimensional toy datasets and the Communities and crime dataset. We performed hyperparameter sweeps over different values of σ which controls regularisation strength, with a smaller value of σ indicating stronger regularisation.

Model	Subnetworks	Learning Rate	σ
Baseline	{1}	$\{1 \times 10^{-3}\}$	$\{1, 3, 5, 10, 30, 50, 5000\}$
Naive	{2,3,4,5}	$\{1 \times 10^{-3}\}$	$\{1, 3, 5, 10, 30, 50, 5000\}$
MIMO	{2,3,4,5}	$\{1 \times 10^{-3}\}$	$\{1, 3, 5, 10, 30, 50, 5000\}$
BNN	{1}	$\{3 \times 10^{-4}\}$	$\{1, 3, 5, 10, 30, 50, 5000\}$
MIMBO	{2,3,4,5}	$\{3 \times 10^{-4}\}$	$\{1, 3, 5, 10, 30, 50, 5000\}$

Table 1. Hyperparameter table for the regression models. σ is the standard deviation of the prior for Bayesian models, or controls the L2 regularisation strength in non-Bayesian models, $\lambda = \frac{1}{2\sigma^2}$.

The results of the hyperparameter sweeps can be found in table 2.

Model	Subnetworks	σ_{1D}	$\sigma_{MultiDim}$	σ_{Crime}
Baseline	{1}	{100}	{10}	{10}
Naive	{2,3,4,5}	$\{3, 5000, 50, 3\}$	$\{50, 5000, 10, 10\}$	$\{100, 5000, 3, 100\}$
MIMO	{2,3,4,5}	$\{3, 50, 5, 5000\}$	$\{100, 10, 30, 50\}$	$\{1, 5000, 3, 30\}$
BNN	{1}	{1}	{1}	{1}
MIMBO	{2,3,4,5}	$\{1, 1, 1, 1\}$	$\{1, 1, 1, 1\}$	$\{1, 1, 1, 1\}$

Table 2. Hyperparameter sweep results for the regression models. For each dataset, the hyperparameter's position in the list matches the number of subnetworks it has, e.g. the result for MIMO with 2 subnetworks is the leftmost number in the list, 3 is the next number to the right and so on.

Classification models

The classification datasets, CIFAR-10 and CIFAR-100 are very similar, so we would perform the same sweeps for both datasets. This includes the sweep over the dropout rate as described in the main text.

Model	Subnetworks	σ	Dropout†
Baseline	{1}	$\{1, 3, 5, 10, 30, 50, 5000\}$	-
Naive	{2,3,4,5}	$\{1, 3, 5, 10, 30, 50, 5000\}$	{0, 0.15, 0.3}
MIMO	{2,3,4,5}	$\{1, 3, 5, 10, 30, 50, 5000\}$	{0, 0.15, 0.3}
BNN	{1}	$\{1, 3, 5, 10, 30, 50, 5000\}$	{0, 0.15, 0.3}
MIMBO	{2,3,4,5}	$\{1, 3, 5, 10, 30, 50, 5000\}$	{0, 0.15, 0.3}

Table 3. Hyperparameter table for the classification models. σ is the standard deviation of the prior for Bayesian models, or controls the L2 regularisation strength in non-Bayesian models, $\lambda = \frac{1}{2\sigma^2}$. The † denotes the hyperparameter is only used in Wide ResNet.

The results of the hyperparameter sweeps are in the tables below. Because we did not have the time to perform hyperparameter sweeps for CIFAR-100, we simply use

the values determined for the equivalent CIFAR-10 model. While the best values for CIFAR-10 are unlikely to be the best for CIFAR-100, we decided it was better to use the CIFAR-10 values over randomly guessing hyperparameters.

Model	**Subnetworks**	$\sigma_{MediumCNN}$	$\sigma_{WideResNet}$	**Dropout**†
Baseline*	{1}	{3}	{-}	{-}
Naive	{2,3,4,5}	$\{50, 1, 30, 30\}$	$\{5, 1, 1, 5\}$	{0, 0, 0.15, 0}
MIMO	{2,3,4,5}	$\{50, 5, 3, 3\}$	$\{30, 10, 10, 10\}$	{0, 0, 0, 0}
BNN	{1}	{1}	{50}	{0}
MIMBO	{2,3,4,5}	$\{5000, 30, 50, 3\}$	$\{5000, 10, 5, 3\}$	{0, 0, 0, 0}

Table 4. The table for the results of the hyperparameter sweeps on CIFAR-10. The hyperparameter's position in the list matches the number of subnetworks it has, e.g. the result for MIMO with 2 subnetworks is the leftmost number in the list, 3 is the next number to the right and so on.

Appendix F: Additional examples of the effect of adding subnetworks for the predictive probability distribution

The following plots supplement the plot in the end of results for CIFAR-10.

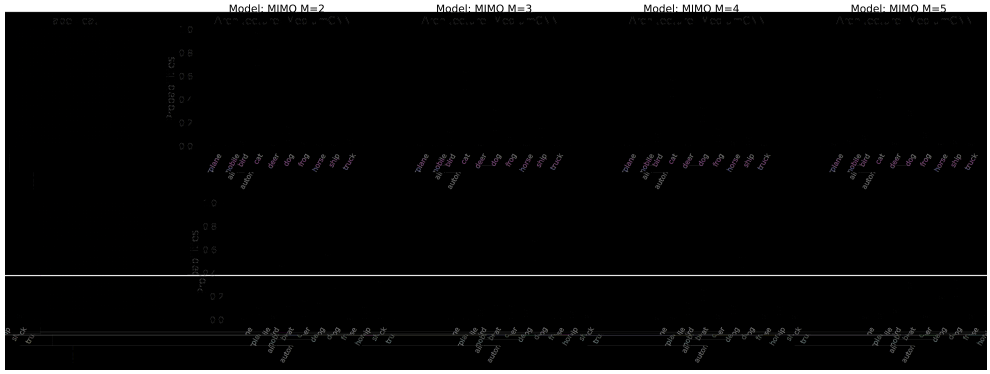


Figure 2. CIFAR-10 MIMO with Medium CNN architecture

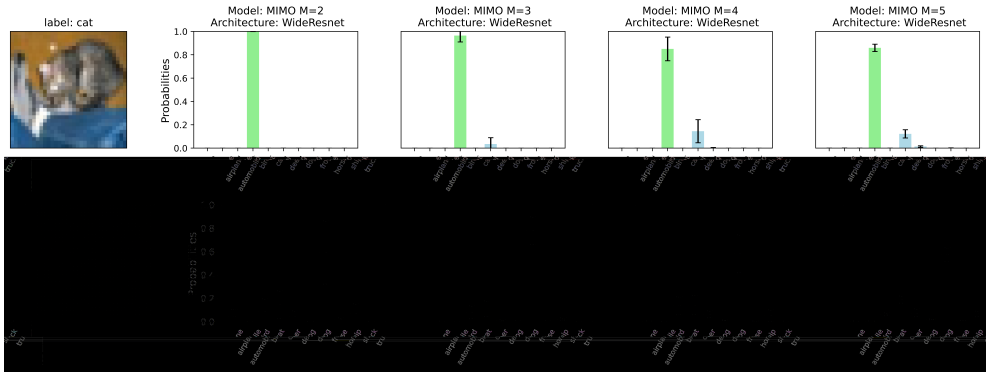


Figure 3. CIFAR-10 MIMO with Wide ResNet architecture

Figure 4. CIFAR-10 MIMO with Wide ResNet architecture

All the above images all show how additional subnetworks smoothen out the predictive probability distribution.

Appendix G: Training accuracy of a subnetwork as M increases

(a) **(b)**

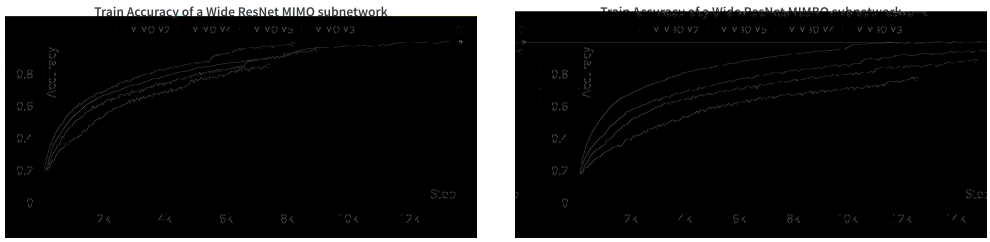


Figure 5. This figure shows the training accuracy of a single subnetwork from a MIMO model (left) and a MIMBO model (right) with Wide ResNet architecture on CIFAR-10 for $M = 2, 3, 4, 5$ subnetworks.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *CoRR*, vol. abs/1708.02709, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[4] L. Liu, W. Ouyang, X. Wang, P. W. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *CoRR*, vol. abs/1809.02165, 2018.

[5] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[6] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.

[7] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, (New York, NY, USA), p. 1721–1730, Association for Computing Machinery, 2015.

[8] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. M. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *CoRR*, vol. abs/2107.03342, 2021.

[9] S. Dirmeier, Y. Hong, Y. Xin, and F. Perez-Cruz, "Uncertainty quantification and out-of-distribution detection using surjective normalizing flows," 2023.

[10] B. Lakshminarayanan, D. Tran, and J. snoek, "Introduction to uncertainty in deep learning," 2021.

[11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *CoRR*, vol. abs/1706.04599, 2017.

[12] Y. Bai, S. Mei, H. Wang, and C. Xiong, "Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification," *CoRR*, vol. abs/2102.07856, 2021.

[13] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *CoRR*, vol. abs/1610.02136, 2016.

[14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2017.

[15] M. W. Dusenberry, G. Jerfel, Y. Wen, Y. Ma, J. Snoek, K. A. Heller, B. Lakshminarayanan, and D. Tran, "Efficient and scalable bayesian neural nets with rank-1 factors," *CoRR*, vol. abs/2005.07186, 2020.

[16] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015.

[17] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," 2020.

[18] Y. Wen, D. Tran, and J. Ba, "Batchensemble: An alternative approach to efficient ensemble and lifelong learning," *CoRR*, vol. abs/2002.06715, 2020.

[19] M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, and D. Tran, "Training independent subnetworks for robust prediction," 2021.

[20] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Training pruned neural networks," *CoRR*, vol. abs/1803.03635, 2018.

[21] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *CoRR*, vol. abs/1506.02626, 2015.

[22] M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic, "Revisiting the calibration of modern neural networks," 2021.

[23] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," *ICML*, vol. 1, 05 2001.

[24] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classif.*, vol. 10, 06 2000.

[25] M. Kull, T. S. Filho, and P. Flach, "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, pp. 623–631, PMLR, 20–22 Apr 2017.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 ed., 2007.

[27] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT 1993, Santa Cruz, CA, USA, July 26-28, 1993* (L. Pitt, ed.), pp. 5–13, ACM, 1993.

[28] D. Barber and C. Bishop, "Ensemble learning for multi-layer networks," in *Advances in Neural Information Processing Systems* (M. Jordan, M. Kearns, and S. Solla, eds.), vol. 10, MIT Press, 1997.

[29] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems* (J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds.), vol. 24, Curran Associates, Inc., 2011.

[30] J. Heek, "Well-calibrated bayesian neural networks: On the empirical assessment of calibration and construction of well-calibrated neural networks," *Master's Thesis*, 2018.

[31] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, p. 859–877, Apr. 2017.

[32] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," 2018.

[33] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Verlag, 2004.

[34] R. M. Neal, *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag, 1996.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[36] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[37] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.

[38] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical stochastic gradient mcmc for bayesian deep learning," 2020.

[39] J. Heek and N. Kalchbrenner, "Bayesian inference for large scale image classification," 2019.

[40] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," 2014.

[41] Z. Deng, F. Zhou, and J. Zhu, "Accelerated linearized laplace approximation for bayesian deep learning," 2022.

[42] A. Y. K. Foong, Y. Li, J. M. Hernández-Lobato, and R. E. Turner, "'in-between' uncertainty in bayesian neural networks," 2019.

[43] A. Kristiadi, M. Hein, and P. Hennig, "Being bayesian, even just a bit, fixes overconfidence in relu networks," 2020.

[44] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[45] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2016.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," 2014.

[47] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," 2016.

[48] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton, "Hyperparameter ensembles for robustness and uncertainty quantification," 2021.

[49] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[50] G. Nam, J. Yoon, Y. Lee, and J. Lee, "Diversity matters when learning from ensembles," *CoRR*, vol. abs/2110.14149, 2021.

[51] Y. Lecun, J. Denker, and S. Solla, "Optimal brain damage," 01 1989.

[52] A. Baumann, T. Roßberg, and M. Schmitt, "Probabilistic mimo u-net: Efficient and accurate uncertainty estimation for pixel-wise regression," 2023.

[53] S. Cygert and A. Czyżewski, "Robust object detection with multi-input multi-output faster r-cnn," in *Image Analysis and Processing – ICIAP 2022* (S. Sclaroff, C. Distante, M. Leo, G. M. Farinella, and F. Tombari, eds.), (Cham), pp. 572–583, Springer International Publishing, 2022.

[54] M. Pitropov, C. Huang, V. Abdelzad, K. Czarnecki, and S. Waslander, "Lidar-mimo: Efficient uncertainty estimation for lidar-based 3d object detection," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 813–820, 2022.

[55] A. Ramé, R. Sun, and M. Cord, "Mixmo: Mixing multiple inputs for multiple outputs via deep subnetworks," *CoRR*, vol. abs/2103.06132, 2021.

[56] M. Ferianc and M. Rodrigues, "Mimmo: Multi-input massive multi-output neural network," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4564–4569, 2023.

[57] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. W. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *CoRR*, vol. abs/2011.06225, 2020.

[58] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction," *CoRR*, vol. abs/1910.09457, 2019.

[59] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[60] T. Salimans, D. P. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," 2015.

[61] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[62] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," 1995.

[63] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *CoRR*, vol. abs/1901.02731, 2019.

[64] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[67] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2017.

[68] J. Shlens, "A tutorial on principal component analysis," *CoRR*, vol. abs/1404.1100, 2014.

[69] A. Sgarro, "Informational divergence and the dissimilarity of probability distributions," *CALCOLO*, vol. 18, pp. 293–302, 1981.

[70] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, pp. 1–3, 1950.

[71] K. N. Markelle Kelly, Rachel Longjohn, "The uci machine learning repository."

[72] M. Redmond, "Communities and Crime." UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C53W3X.

[73] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[74] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *CoRR*, vol. abs/1903.12261, 2019.

[75] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017.

[76] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," 2022.

[77] M. Valdenegro-Toro and D. Saromo, "A deeper look into aleatoric and epistemic uncertainty disentanglement," 2022.

{2.7182818284}ϑφερτυθιοπσδφγηξκλ