Improving Deep Ensembles for Better Deep Uncertainty Quantification

Ginte Petrulionyte

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Churchill College August 2021

Declaration

I, Ginte Petrulionyte of Churchill College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

I further declare the software and data used for this thesis. The experimentation was carried out using Python and its standard scientific and machine learning libraries: Numpy, Scipy, PyTorch. Experiments using the Laplace approximation used a further open-source library (Daxberger et al., 2021). Weights and Biases experiment tracking software (Biewald, 2020) was used to monitor experimentation, produce intermediate visualisations and reports. Plots and visualisations included in this report were generated using Matplotlib. None of the aforementioned software was modified. Additionally, the metric computation and sparse gating for mixtures of experts implementations were based on the code provided by Ovadia et al. (2019) and Shazeer et al. (2017) respectively. No other third-party software was used. Freely publicly available data was used for the project, including the MNIST, CIFAR10, and CIFAR100 datasets, as well as corrupted versions of the latter two.

Word count (including appendices and captions): 14955

Ginte Petrulionyte
August 2021

Acknowledgements

I would like to thank my supervisor, Jose Miguel Hernandez Lobato, for the insight and critique provided throughout the course of the project. Also, my co-supervisor, Erik Daxberger, for the many insights, suggestions, and even more hours spent in video conferences. The guidance was always appreciated, and the project would not be nearly as complete without the encouragement and support of my supervisors.

Lastly, a special thanks to my flatmates who helped me get through the year full of lockdowns, self-isolation and study stress, and listened to many frustrated monologues. Also, for performing the vital task of proofreading this dissertation and helping me discover (hopefully all) the countless typos!

Abstract

Deep neural networks have become increasingly accurate and applicable to decision-making in high-risk real-world settings, such as medicine or autonomous driving. In these situations, the ability to detect that a network might be wrong – due to the input type changing, the domain shifting, or any other cause – is crucial and allows requesting timely intervention. While there are ways to produce uncertainty estimates for modern neural networks, they tend to be overconfident and are rarely able to detect unfamiliar situations.

Deep ensembles have been shown to provide remarkable improvements in prediction calibration – the correspondence of estimated uncertainty and empirical accuracy – both for data drawn from the same distribution as the training set and in situations where dataset shift is observed. The diversity of ensemble member predictions has been shown to be a key factor differentiating deep ensembles from other alternatives, such as Bayesian methods.

Despite the inherent diversity, recent research has explored strategies to further diversify deep ensembles, which we categorise as explicit and implicit. We explore in detail two explicit methods for classification ensembles: negative correlation learning and regularising via pairwise subnetwork cross-entropy. We show both methods can improve over deep ensemble calibration under ideal conditions, but depend heavily on the choice of a scaling hyperparameter value, which is difficult to tune with only in-distribution data available.

As a complementary method, we present a novel view of mixtures of experts (implemented with neural networks for all component predictors) as a form of implicit deep ensemble diversification. We show this strategy produces highly diverse and localised member networks but has poor out-of-the-box calibration. This can be improved by using Bayesian gating networks, with localisation and increased diversity maintained but confidence level adjusted. Such strategies achieve calibration similar to that of deep ensembles for shifted data. However, significant challenges are encountered when training mixtures of experts and the diversifying effect is not fully utilised by these strategies, suggesting further work is needed to make the method competitive.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

DE	Deep ensemble
DNN	Deep neural network
ECE	Expected Calibration Error
ID	In-distribution
LA	Laplace Approximation
MC	Monte Carlo
MoE	Mixture of experts
NLL	Negative Log-Likelihood
OOD	Out-of-distribution

Chapter 1

Introduction

1.1 Motivation

Modern deep neural networks (DNNs) can achieve remarkable feats in their predictive performance. Their capabilities have long expanded beyond somewhat isolated tasks like recognising handwritten digits where the impact of trusting incorrect predictions is limited. DNNs are increasingly applicable in scenarios with significantly higher inherent risk, such as medical diagnostics (Esteva et al., 2017) and autonomous driving. In both of these settings, an incorrect prediction made by the network can have far-reaching consequences – making patients undergo unnecessary invasive procedures or causing crashes, e.g. the 2016 Tesla incident, partially influenced by a segmentation error (NHTSA, 2017).

DNNs are capable of producing uncertainty estimates, indicating how confident they are in the prediction's correctness. A natural mitigation strategy for these issues might thus be flagging up uncertain predictions as requiring further human attention. However, this relies on the networks' calibration – the confidence estimates corresponding to the true likelihood of predictions being correct. Unfortunately, DNNs are often significantly over-confident (Guo et al., 2017), reporting extremely high certainty in all predictions even when the overall accuracy is relatively low. This is further exaggerated under dataset shift, e.g. when images different from ones seen during the supervised training process are encountered, with plenty of humorous examples – like models confident that a child is a balance beam (Shafaei et al., 2019). Similarly, images falling into the training categories, but obstructed or shifted cause the accuracy to drop, while confidence often remains high (Ovadia et al., 2019).

Deep ensembles (DEs) – sets of identical (except for initialisation) independently trained DNNs used to produce a single prediction – provide remarkable improvements in uncertainty estimation, as shown by Ovadia et al. (2019). It is a fundamentally intuitive concept – as long as the predictors are individually accurate but not identical, we might expect them to make different mistakes. In particular, for images different from ones then networks were trained on, the individual ensemble members might still produce over-confident predictions, but they can now be diverse. As the final prediction is produced by averaging, the final confidence estimate is generally lower (Rahaman and Thiery, 2020).

In this dissertation, we study DEs and methods which can be used to ensure the diversity of their members' predictions. We investigate implicit and explicit strategies used to achieve this and their impact on the calibration of uncertainty estimates, both for in-distribution data and under dataset shift.

1.2 Contributions

The contributions of this dissertation are as follows:

- We present a structured view of popular DE diversification methods, bringing focus to the distinction between explicit and implicit diversification;
- We analyse explicit DE diversity regularisation methods, originally proposed to improve in-distribution calibration and predictive performance, in the context of calibration under distribution shift, providing a more thorough understanding of their impact;
- We identify the selection of diversification-specific hyperparameters as a key challenge to applying explicit DE diversity regularisation in practice. We show that the sensitivity to their values is high, and an in-distribution validation set cannot be reliably used to choose their values while maintaining calibration improvements under dataset shift;
- Lastly, we provide novel analysis of the inherently diverse ensembles constructed as mixture of experts models in the context of calibration. We show the calibration of a mixture of experts tends to mimic that of a single predictor but can be improved by using variations of Bayesian DNNs for the gating model.

Additionally, we provide the codebase created throughout this project as an open-source repository[1], containing a flexible and highly customisable framework for experimentation with DEs and their variations.

1.3 Dissertation Outline

The dissertation follows the structure outlined here. In Chapter 2 we present an overview of relevant background material on uncertainty quantification and DEs followed by a structured

[1] The codebase can be found at https://github.com/gintepe/DeepEnsembleUncertainty.

review of their diversification methods. Chapter 3 describes the experimental setup used throughout the work, including a discussion of evaluation metrics and baseline methods.

The following two chapters contain the results, our analysis and discussion. Chapter 4 presents two methods of explicit deep ensemble diversification, NCL-DE and CE-DE, chosen as promising approaches not previously examined in terms of impact on calibration under dataset shift. We provide an overview of the potential these strategies offer under ideal conditions and discuss the challenges encountered when choosing the values of diversification-specific hyperparameters. Chapter 5 instead studies the implicitly induced diversity of mixture of experts models. We analyse the classic model's calibration and explore strategies to improve it via using Bayesian approaches for the gating network. Lastly, in Chapter 6 we summarise the overall findings and discuss aspects of potential future work.

Chapter 2

Background

In this chapter, we introduce methods for uncertainty estimation in deep neural networks (DNNs), the concept of calibration and the importance of examining calibration under distribution shift. This is followed by a brief literature review on deep ensembles (DEs) where we provide a structured outlook on existing strategies aiming to improve both their calibration and predictive performance via diversification. Lastly, we introduce the Mixture of Experts (MoE) paradigm and its connection to deep ensembles.

2.1 Uncertainty Estimation in Neural Networks

It is often useful to interpret neural networks not simply as black-boxes capable of providing predictions, but as a probabilistic model taking into account observed data and producing a predictive distribution. It is common to take a Bayesian view, framing the goal distribution as a Bayesian model average. It can be expressed as

$$p(y \mid \boldsymbol{x}, \mathscr{D}) = \int p(y \mid \boldsymbol{x}, \theta) p(\theta \mid \mathscr{D}) d\theta, \tag{2.1}$$

given a dataset $\mathscr{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$ and parameters θ specifying the model.

The probability over outputs given a set of parameters, $p(y|\boldsymbol{x}, \theta)$, is often referred to as the likelihood when seen as a function of θ. The equation combines these functions across different parameter settings, as weighted by the parameter posterior $p(\theta|\mathscr{D})$. The latter can be formally obtained using a prior belief on the weights via Bayes theorem as $p(\theta|\mathscr{D}) = \frac{p(\mathscr{D}|\theta)p(\theta)}{p(\mathscr{D})}$.

The approach is robust in modelling epistemic uncertainty – one arising from the variation in possible models fitting the data. However, the integral is usually intractable, and has to be approximated, as the distribution over models is often difficult to estimate.

Traditional neural network training can be seen as a crude approximation. They provide a point estimate of the weights – $\hat{\theta}$ – and are equivalent to solving this integral with $p(\theta|\mathscr{D}) = \delta(\theta - \hat{\theta})$, where δ is the Dirac delta function (a probability density function zero everywhere except the origin). For point-estimate networks, uncertainty values can be derived by a set of well-established methods. In a k-class classification problem, a final layer with k units and a softmax activation function is typically employed. Given the raw final layer output $\boldsymbol{h}$, the activations $\boldsymbol{o}$ are found as

$$o_i = \frac{\exp(h_i)}{\sum_{j=1}^{k} \exp(h_j)}. \tag{2.2}$$

The transformation ensures final activations sum to 1, and output can be interpreted as a categorical probability distribution over the possible classes. For an input $\boldsymbol{x}$, ground truth label y and model parameters θ, we consider the output to be $o_i = p(y = i|\boldsymbol{x}, \theta)$. Predictions are then made by selecting the class with the highest corresponding probability, as $\hat{y} = \text{argmax}_i(o_i)$. The output values serve directly as confidence estimates.

Regression networks require a slight adjustment to provide interpretable confidence estimates. The models typically only output a predictive estimate, without any indication of confidence. A common approach is to instead train a network with two outputs – the predictive mean and variance (Jain et al., 2020; Lakshminarayanan et al., 2016). These can be further interpreted to specify a Gaussian distribution approximating the predictive posterior, and a training criterion chosen appropriately.

Due to the collapsing of the integral in Equation 2.1 to a single point, the uncertainty estimates above are more reflective of aleatoric – inherent to the data – uncertainty.

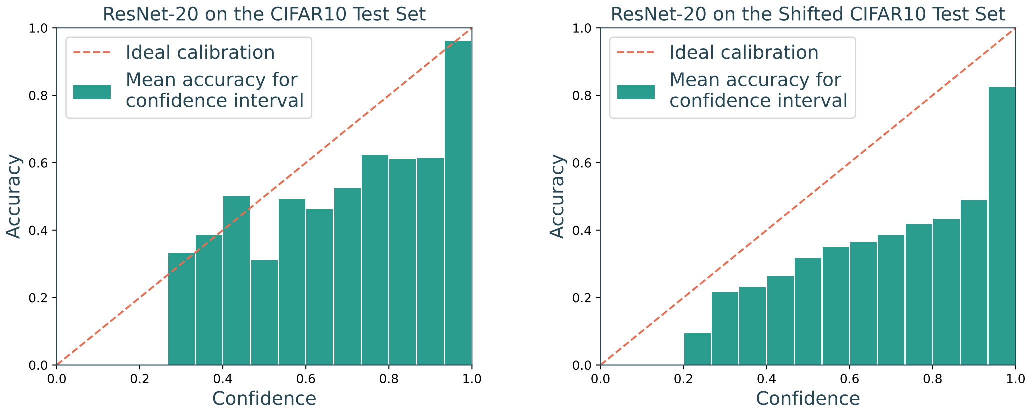
Bayesian deep learning aims to approximate the Bayesian model average more explicitly. If it is possible to sample from $p(\theta|\mathscr{D})$, a Monte Carlo (MC) approximation can be used to approximate the overall integral:

$$p(y \mid \boldsymbol{x}, \mathscr{D}) \approx \frac{1}{M} \sum_{m=1}^{M} p(y \mid \boldsymbol{x}, \theta_m), \theta_m \sim p(\theta|\mathscr{D}). \tag{2.3}$$

The distribution $p(\theta|\mathscr{D})$ is often difficult to directly express and sample from. It is commonly approximated with an easy to sample from proxy distribution $q(\theta, \mathscr{D})$, parametrized to make q as close to p as possible. For example, Laplace approximation (MacKay, 1992) can be used to approximate the distribution by a Gaussian centred at a point estimate. Another common approach is to fit the parameters of q by variational inference (Blundell et al., 2015). Lastly, it has been shown by Gal and Ghahramani (2016) that training and testing using Dropout (Hinton et al., 2012) or DropConnect (Mobiny et al., 2021; Wan et al., 2013) can be seen as approximating such a distribution as well. Test-time predictions are obtained by averaging the outputs of multiple forward passes through the network, each with a different random sample of nodes or connections deactivated. Ultimately, this yields a Monte Carlo estimate of the predictive distribution and the method is often referred to as MC Dropout.

2.2 The Uncertainty Calibration Problem

Having a range of methods for estimating the uncertainty of DNN predictions enables us to study their quality. One of the problems often examined is uncertainty calibration. In the context of machine learning, calibration refers to the correspondence between predictor accuracy and the uncertainty estimates it produces. In other words, a well-calibrated network can detect when its predictions are likely to be incorrect, often referred to as "knowing what it does not know". While early neural networks were reasonably well-calibrated (Niculescu-Mizil and Caruana, 2005), in their work exploring the calibration of modern DNNs Guo et al. (2017) show this is no longer true. They note a correlation between recent increases in model capacity, depth, reduced use of weight regularisation, and significant overconfidence – causing dangerously poor prediction calibration.



(a) Original in-distribution test set

(b) Shifted test set

Fig. 2.1 Reliability diagram illustrating the calibration of ResNet-20 (He et al., 2016) on an image recognition dataset. Network's predictions are binned along the x-axis by confidence, with bar height indicating the mean accuracy in the bin. A well-calibrated network would have accuracy falling within bin boundary values. Both figures show model overconfidence, with empirical accuracy lying significantly below the corresponding confidence estimate (red line).

Reliability diagrams allow us to visualise over and under-confidence, showing accuracy for predictions made with confidence within a given range. Fig. 2.1a provides an example typical of modern DNNs, using a popular architecture on an in-distribution (ID) test set. The network exhibits significant overconfidence in predictions with low uncertainty, and slight under-confidence on the other end of the spectrum. These behaviours indicate that confidence estimates produced are largely unreliable.

2.2.1 Calibration Under Dataset Shift

A considerable body of work exists studying DNN uncertainty estimates on out of distribution (OOD) inputs (Hendrycks and Gimpel, 2016; Lakshminarayanan et al., 2016), relating the calibration problem to the wider field of outlier (or OOD) detection (Hendrycks et al., 2018; Shafaei et al., 2019). Similarly, calibration for ID inputs is also crucial, and studied in depth (Guo et al., 2017). However, Ovadia et al. (2019) argue that distribution shift provides the most appropriate context for studying DNN calibration and quality of uncertainty estimates.

Dataset shift can occur naturally in a variety of settings (e.g. different camera settings or light conditions for images). It is often gradual, with inputs resembling the ID samples – making methods from other fields, like outlier detection, not applicable directly.

The test distribution not matching that of training data exactly can often further exaggerate miscalibration. As illustrated by Fig. 2.1b, it can cause the network to be even more overconfident as the prediction accuracy inevitably drops, but confidence remains high.

Ovadia et al. (2019) find that strategies which lead to excellent calibration on ID samples are often not robust to dataset shift. In particular, post-hoc temperature scaling on a validation set – found to be most beneficial by Guo et al. (2017) – is used as an example. To comprehensively evaluate a method's calibration it is thus important to study it not only on ID data but also on shifted samples.

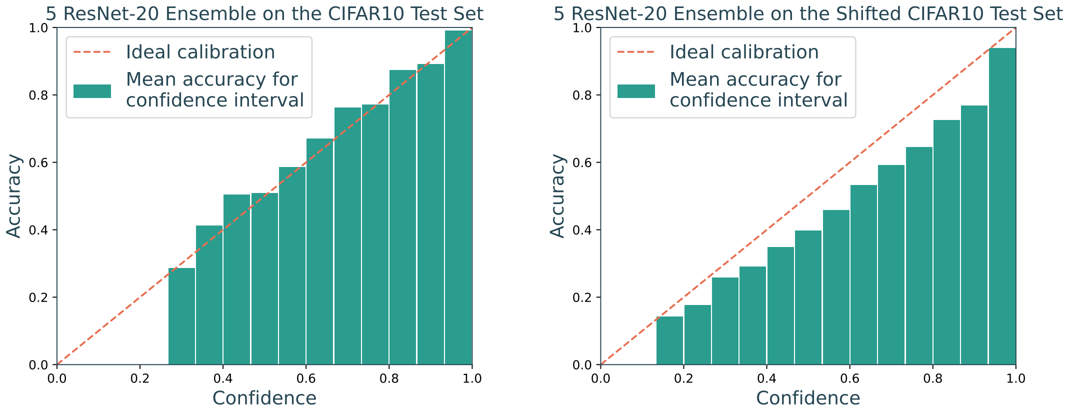
2.3 Deep Ensembles

A variety of recent works (Gustafsson et al., 2020; Lakshminarayanan et al., 2016; Ovadia et al., 2019) find that a fundamentally simple method – deep ensembles (DEs) – outperforms a wide variety of other methods for producing robust uncertainty estimates in terms of calibration. In this section, we present detailed background on the method and provide a review of strategies used to further improve their performance.

Replacing a single predictor by an ensemble has long been a popular strategy for improving overall predictions. The combined output, produced by (potentially weighted) averaging or voting, have proven to be more robust, reducing overall model variance, and often more accurate than a single model (Dietterich, 2000).

Deep ensembles, formally introduced by Lakshminarayanan et al. (2016), provide a simple ensemble construction method. It only requires several identical DNNs trained on the same dataset, using the same training procedure. The only difference lies in their random initialisation of the parameters. Ensemble predictions are constructed by averaging the individual outputs. The strategy is loosely motivated by the classic ensembling technique called Bagging (bootstrap aggregating), where identical predictors are trained on datasets created by repeatedly sampling from the original one with replacement. However, DEs consistently outperform Bagging (Nixon et al., 2020), showing the complexity added by dataset re-sampling is redundant in this case, and the random initialisation is sufficient to ensure good performance.

Relating back to the predictive distribution in Equation 2.1, DEs can be seen as an extension of the point estimate interpretation, sampling several distinct parameter settings via maximum likelihood training. Wilson and Izmailov (2020) and Gustafsson et al. (2020) note this can be interpreted as a Monte Carlo approximation, with the random initialisation inducing approximate sampling from the distribution over model parameters.



(a) Original in-distribution test set

(b) Shifted test set

Fig. 2.2 Reliability diagram illustrating calibration of an ensemble of 5 ResNet-20's on an image recognition dataset. Ensemble predictions are binned along the x-axis by confidence, with bar height indicating the mean accuracy in the bin. Figure on the left displays good calibration, with empirical accuracy similar to corresponding confidence estimates (red line). Figure on the right displays slight but consistent overconfidence.

Due to their simplicity and ease of training parallelisation, DEs have become a popular staple for producing state-of-the-art results in machine learning. As noted earlier and illustrated by Fig. 2.2, they also significantly improve prediction calibration over that of a single network (Fig. 2.1), both for ID and shifted data.

2.4 Diversity in Deep Ensembles

Predictor diversity is often cited as one of the reasons for the effectiveness of ensembles (Buschjäger et al., 2020; Fort et al., 2019; Melville and Mooney, 2004). In the case of tasks evaluated by the mean squared error, it is theoretically motivated by the bias-variance-covariance decomposition of the expected error (we refer to Sammut and Webb (2010) for further detail). For an M network ensemble output $\bar{o}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} o_m(\boldsymbol{x})$ it is given by

$$\mathbb{E}_{\mathscr{D}}\left[(\bar{o}(\mathbf{x}) - y)^2\right] = \overline{\text{bias}}^2 + \frac{1}{M}\overline{\text{var}} + \left(1 - \frac{1}{M}\right)\overline{\text{covar}}. \tag{2.4}$$

Here $\overline{\text{bias}}$, $\overline{\text{var}}$ and $\overline{\text{covar}}$ refer to the mean bias, mean variance and mean pairwise covariance of the individual models. Diverse ensembles explicitly reduce the covariance term by producing less correlated predictions, in turn reducing the overall expected error.

A recent study by Fort et al. (2019) investigated diversity in deep ensembles from the perspective of solution positioning within the overall loss landscape. The authors show DEs typically achieve what many other methods struggle with – exploring distinct modes of the loss and inducing diversity in the functions learned, as illustrated in Fig. 2.3. With finite training data, many DNN parameter settings can explain the observations equally well. However, the extrapolations away from training samples, produced by models with these parameter sets can differ greatly. Random initialisation used for training deep ensembles is typically sufficient to allow individual predictors to find distinct solutions. Bayesian methods based on subspace sampling (such as MC Dropout or weight averaging) tend to explore uncertainty within a single mode. These findings indicate the diversity is a key factor allowing DEs to not only achieve impressive predictive performance but also better calibration than other strategies.

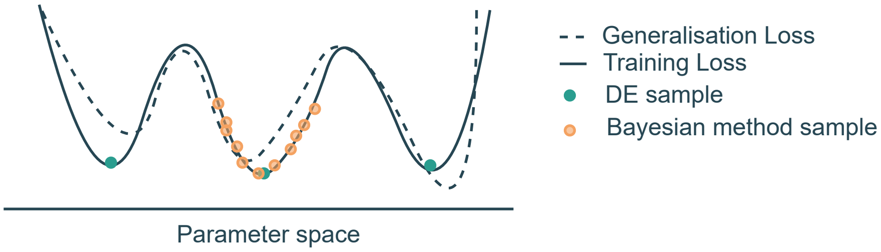


Fig. 2.3 Illustration (based on visualisation in Fort et al. (2019)) of optima found by DEs as compared to common Bayesian methods. Individual networks in DEs fit distinct modes, but ignore local uncertainty and may not pick the solution which generalises best, while Bayesian methods explore local uncertainty.

Under ideal conditions, we may not only want to model different modes in prediction space but also ensure they are picked to be significantly diverse. A variety of recent research has focused on further diversifying DEs. We believe a structured perspective is crucial to understanding the methodologies explored, and propose the classification of diversification methods shown in Fig. 2.4.

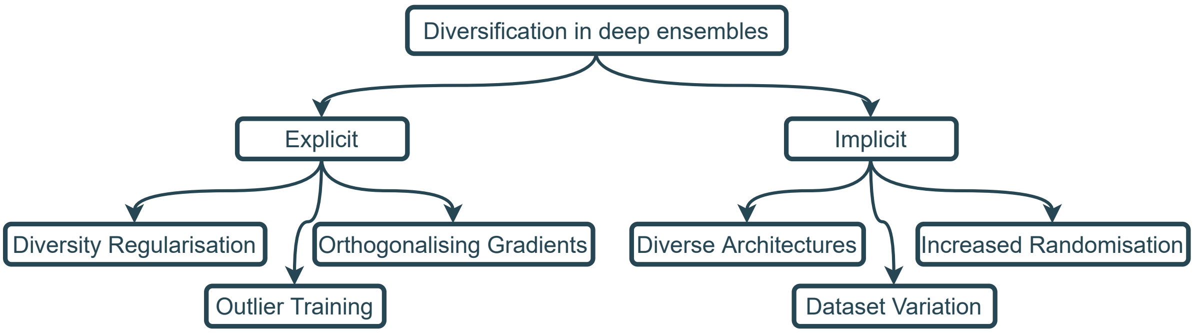


Fig. 2.4 Proposed taxonomy for existing deep ensemble diversification methods. Leaf nodes provide classes of common illustrative examples and are not intended as an exhaustive list.

The main feature differentiating the methods is the approach to inducing diversity. Some methods are *explicit*, defining a measure of diversity, or a diversifying factor to optimise for. Others, like DEs in their original formulation, are *implicit*. They often rely on randomness, sampling, and decisions made before training the predictors, with the overall framework remaining the same – independent network optimisation and post-hoc prediction combination. We explore the two diversification method families in detail in the following sections.

2.4.1 Implicit Deep Ensemble Diversification

There are several key features to standard DE training – the DNNs used are identical, use the same hyperparameters, and are trained on the same data. Implicit diversification typically focuses on changing one of these aspects.

Constructing ensembles with predictors differing in terms of architecture is a classic concept (Hansen and Salamon, 1990) explored in the context of DEs by Zaidi et al. (2020). The authors suggest an automated strategy – using architecture search to construct a large pool of trained predictors, from which an ensemble is selected via the procedure outlined by Caruana et al. (2004), with validation performance as a heuristic. The method improves over basic DE performance (where DE predictor architecture is chosen to be the same as the best individual predictor found) both on ID test images, and shifted datasets. However, the effect of architecture diversification is not isolated. When an equivalent ensemble selection procedure is run for an equally sized set of trained networks with the same architecture, but different parameter initialisations, baseline performance is also improved, although to a lesser extent.

Following a similar final model selection strategy, Wenzel et al. (2020) suggest utilising random search over hyperparameters to construct diverse DEs. A hyperparameter-diverse ensemble is initially selected from the set of networks trained during a search for a network with a fixed initialisation. For every selected setup additional models stratified over random initialisations are trained, creating a new pool of possible predictors out of which a final ensemble is selected. The authors conclude deep ensembles benefit from both initialisation and hyperparameter diversity. Additionally, they note an ensemble selected from models using a fixed initialisation and varied hyperparameters, can improve over standard DE calibration.

Both the strategies discussed rely on constructing a pool of hundreds of trained predictors to select from. Although an argument can be made for the use of both architecture and hyperparameter search as procedures one might run regardless of ensembling – simply to tune a network appropriately – the baseline ensemble training cost increases dramatically when compared with DEs. Wenzel et al. (2020) additionally propose a computationally conscious alternative based on batch ensembles (Wen et al., 2020), outperforming the baseline while only increasing the training cost by a factor of two, making the method more accessible.

Lastly, deep ensemble diversity can be implicitly encouraged by dataset variation. Although, as mentioned in Section 2.4, sub-sampling the dataset (as in Bagging) is not generally beneficial in DE training, other methods have been proposed, training each predictor on differently augmented data. Stickland and Murray (2020) show this strategy tends to improve calibration, particularly for shifted data. Although the original work exclusively used efficient ensembling, the proposed diversification can be directly applied to traditional DEs.

While we identify these as basic implicit diversification methods, it is not an exhaustive list. In an earlier paper, Lee et al. (2016) propose using multiple-choice learning – choosing k ensemble members with the lowest individual losses per sample to backpropagate through during training. This results in ensemble members being trained on subsets of the original dataset, specialising to achieve a state where some network is correct for almost any sample, although ensemble performance may be limited. Similarly, a classic ensembling method which we explore in-depth (see Section 2.5 and Chapter 5) – mixtures of experts – achieves implicit dataset subsampling and network specialisation by using a gating model.

Despite the promising results described, implicit diversification methods do not provide any guarantees – the diversity is not optimized for, and the final solution is not produced under requirements to maximize it. This is directly contrasted by explicit diversification strategies discussed in the following section.

2.4.2 Explicit Diversification Approaches

The most intuitive explicit DE diversification method is diversity regularisation. It involves deriving a quantifiable measure of overall diversity for the set of classifiers and using its scaled version as an additive loss term. In training, this forces optimisation not only for low error, but also for high diversity within the ensemble. A drawback of such methods is a reduction in computational efficiency. To compute the regulariser we often need predictions from all ensemble members, making parallelisation less trivial.

A classic strategy of diversity regularisation is negative correlation learning (Buschjäger et al., 2020; Liu and Yao, 1999; Shui et al., 2018). Here a term rewarding negatively correlated predictions is used, explicitly inducing diversity beyond the point we might expect if the networks were simply uncorrelated. The method has been shown to be effective in diversifying deep ensembles in ID testing, however further effects and trade-offs induced by this regularisation term remain to be studied. We do so in Chapter 4.

Other ways to quantify diversity yield different regularisers. Opitz et al. (2016), in their study of efficient ensembling for classification, suggest regularising via the average negative cross-entropy between predictor pairs. Meanwhile, Dvornik et al. (2019) focus on the categorical distributions over non-ground-truth classes produced by the classification predictors and regularise for negative cosine similarity between the vectors specifying these.

Jain et al. (2020) note that maximising diversity on the ID training data may be insufficient, especially for improving calibration under dataset shift. They suggest regularising by an approximation of OOD uncertainty estimates – an approach closely related to some OOD detection methods (Hendrycks et al., 2018). Authors choose to draw from a uniform distribution over the input space to produce this approximation, rather than use a specific dataset. This creates a relatively unbiased approximation of possible data and yields remarkable results for the regression tasks studied. However, this approach may not be as beneficial in, for example, image classification as sampling noise from a random distribution does not reflect any realistic OOD inputs – natural images are fundamentally highly structured.

Rather than optimising for function space diversity or OOD calibration, Kariyappa and Qureshi (2019) seek to improve DE performance on adversarial examples. The authors argue their goal can be achieved by creating non-overlapping adversarial subspaces via diversification. They show explicitly minimising gradient alignment throughout training can make ensembles significantly more robust to traditional adversarial examples. At first glance, the connection between this form of diversification and functional diversity seems indirect and may be considered implicit. However, more direct diversification methods have been shown to improve adversarial robustness (e.g. Pang et al. (2019) optimising for diversity in predictions of non-ground-truth classes), showing the connection between the tasks.

In general, explicit DE diversification allows us to directly optimize for a diversity metric most relevant to a task at hand – be it uncertainty on an outlier dataset when training for calibration under dataset shift, or adversarial sub-space overlap when the goal is adversarial robustness. The training procedure can then be expected to find a local optimum that trades-off between individual networks' predictive performance and the ensemble diversity.

2.5 Mixtures of Experts

While DEs rely on combining predictions from trained subnetworks with equal contribution, this is not the only option. An alternative is provided by mixtures of experts (MoE) (Jacobs et al., 1991). It is a classic ensembling method, using a divide-and-conquer approach – dividing the prediction task, explicitly or implicitly, into subtasks, handled by individual predictors.

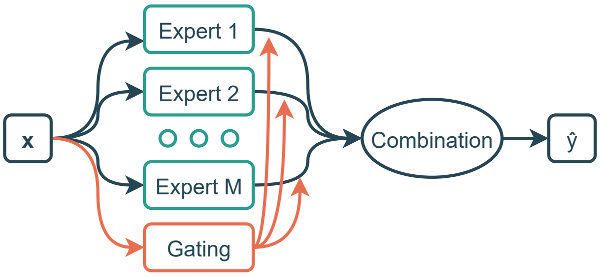


Fig. 2.5 General structure of a mixture of experts model.

A set of M predictors, typically of the same type, are used as "experts", and an auxiliary one provides gating. The standard setup is illustrated in Fig. 2.5. The gating output can typically be interpreted as a categorical distribution over the experts. These values are used in combining the individual outputs into a single ensemble prediction, effectively producing a weighted average. This enables expert specialisation. As long as the gating predictor selects an appropriate combination of outputs for a given input, the individual experts can be only suitable for a subset of possible values.

All MoE components can be implemented as DNNs – in this case, the overall structure can be highly similar to that of a DE, motivating our suggestion it might serve as an alternative. The expert models can be trained independently, using pre-defined subsets of data – referred to as explicit localisation by Masoudnia and Ebrahimpour (2014) – or jointly, by optimising a single loss, with implicit localisation provided by the gating network which is trained alongside.

Expert localisation enables the individual predictors to be extremely diverse. As the training focuses on only a subset of data, the predictions produced by models for sub-tasks they are not specialised to are near-random. Thus MoE can be seen as a potential way to build intrinsically diverse versions of DEs, with the added requirement to train a gating network.

Chapter 3

Experimental Setup

In this chapter we describe the experimental setup used throughout the dissertation. In particular, we cover the main focus of experimentation, the key metrics for method comparison and their relevance, the datasets we use for evaluation and the network architectures utilised.

3.1 Experimental Focus

In Section 2.1 we provided an overview of uncertainty estimation in DNNs. In particular, we noted that while classification networks lend themselves to direct probabilistic interpretation, the approach used for regression is less intuitive. This may have led to emphasis on the former in literature, with papers such as those by Ovadia et al. (2019) and Guo et al. (2017) exclusively examining classification tasks.

To take advantage of pre-existing knowledge, we likewise examine DEs and their variants in the context of classification. In particular, we focus on image recognition tasks – they offer interpretable inputs and established strategies to simulate semi-natural distribution shifts.

3.2 Metrics

Calibration, while an intuitive concept, is non-trivial to quantify. There is no consensus on a single best way to measure DNN calibration. We choose to follow the loose convention (as employed by Ovadia et al. (2019), Guo et al. (2017) and others) of compiling a set of metrics providing complementary insights. The metrics chosen are the following:

- **Expected calibration error (ECE)**. It quantifies the discrepancy between prediction confidence and the empirical accuracy of the model (Naeini et al., 2015). To compute

ECE, the model's predictions on dataset $\mathscr{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$ are partitioned into S bins. With $\rho_0, \rho_1 \ldots \rho_S$ denoting the bin edges we define

$$B_s = \{1 \leq n \leq N | \max_i (p(y = i|\boldsymbol{x}^{(n)}, \theta) \in [\rho_{s-1}, \rho_s)\}. \tag{3.1}$$

ECE is then

$$\text{ECE} = \sum_{s=1}^{S} \frac{|B_s|}{N} \left(\text{Acc}(B_s) - \text{Conf}(B_s)\right). \tag{3.2}$$

Here we use $\text{Acc}(B_s)$ to denote the accuracy of predictions in B_s

$$\text{Acc}(B_s) = \frac{1}{|B_s|} \cdot \sum_{n \in B_s} \mathbb{1}(\hat{y}^{(n)} = y^{(n)}), \tag{3.3}$$

and $\text{Conf}(B_s)$ for mean confidence,

$$\text{Conf}(B_s) = \frac{1}{|B_s|} \cdot \sum_{n \in B_s} p(y = \hat{y}^{(n)}|\boldsymbol{x}^{(n)}, \theta). \tag{3.4}$$

with $\hat{y}^{(n)} = \text{argmax}_i(p(y = i|\boldsymbol{x}^{(n)})$ (predicted class), and $\mathbb{1}(\cdot)$ – the indicator function, 1 when the argument is true and 0 otherwise. Low ECE values indicate good calibration.

Measuring calibration via ECE is intuitive and closely corresponds to visual methods, such as reliability diagrams. However, it has significant drawbacks. It requires choosing the number of bins (we use $S = 20$), potentially affecting the values. The metric also disregards the accuracy achieved and has trivial minimisers. For example, a model outputting the marginal class probabilities in the data distribution would, in expectation, achieve an ECE value of 0.

- **Negative log-likelihood (NLL)**. It is a common measure for assessing a model's performance on a held-out dataset (Friedman et al., 2001). In the form of cross-entropy loss for one-hot ground truth labels, NLL is often used as a training criterion for classification in deep learning. NLL is computed as the negative log-likelihood of the labels, using probabilities assigned to them by the model. In particular, we have

$$NLL = -\sum_{n=1}^{N} \log\left(p(y = y^{(n)}|\boldsymbol{x}^{(n)}, \theta)\right). \tag{3.5}$$

Low NLL indicates the model recovers the data distribution well, while large values indicate a poor fit. Unlike ECE, NLL does not have trivial minimisers and is only zero when the ground-truth labels are recovered with high confidence.

- **Brier score.** This metric provides another way of evaluating both network accuracy and the associated confidence estimates. It is defined as the squared difference between the model's output and a one-hot encoded ground-truth label, $\boldsymbol{y}^{(n)}$ (Brier et al., 1950). Over a dataset $\mathscr{D}$, it is found as

$$\begin{aligned} BS &= \frac{1}{N} \cdot \sum_{n=1}^{N} \left(\frac{1}{K} \cdot \sum_{i=1}^{K} (p(y=i|\boldsymbol{x}^{(n)}, \theta) - y_i^{(n)})^2 \right) \\ &= \frac{1}{N} \cdot \sum_{n=1}^{N} \left(\frac{1}{K} \cdot \sum_{i=1}^{K} (p(y=i|\boldsymbol{x}^{(n)}, \theta) - \mathbb{1}(y^{(n)}=i))^2 \right) \\ &= \frac{1}{N} \cdot \sum_{n=1}^{N} \left(\frac{1}{K} \cdot \left(\sum_{i=1}^{K} \left(p(y=i|\boldsymbol{x}^{(n)}, \theta)^2 \right) - 2p(y=y^{(n)}|\boldsymbol{x}^{(n)}, \theta) + 1 \right) \right). \end{aligned} \tag{3.6}$$

Like NLL, Brier score is minimised if and only if ground truth labels are recovered with high confidence. NLL is highly sensitive to tail probabilities – the penalty for confident, but incorrect predictions is unbounded. Meanwhile, the Brier score is less affected by rare events – the maximum penalty for a single sample is bounded at 1 – making the metrics complementary.

It is also crucial to maintain predictive performance. Thus we additionally report **accuracy**, as a metric focused primarily on correctness, and disregarding confidence estimates.

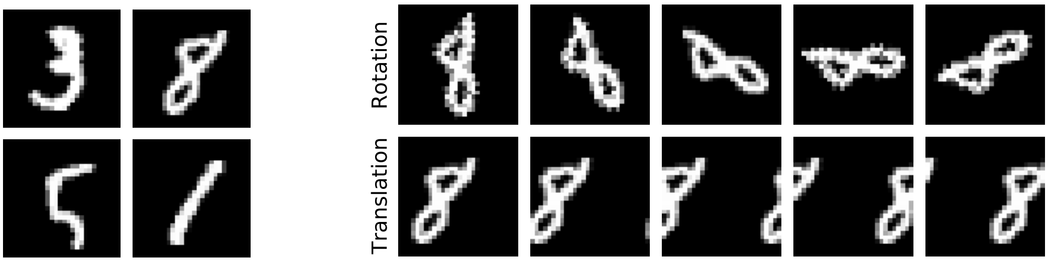
Lastly, in Section 2.4, we noted the importance of diversity among the sub-networks in a DE. To quantify this we use **disagreement** between network pairs, consistent with work by Fort et al. (2019). It is defined as the fraction of samples networks provide different predictions (disagree) on. The disagreement between two classification models with parameters θ_1 and θ_2 on a dataset $\mathscr{D}$ is

$$DA = \frac{1}{N} \cdot \sum_{n=1}^{N} \mathbb{1}(\text{argmax}_i(p(y=i|\boldsymbol{x}^{(n)}, \theta_1)) = \text{argmax}_i(p(y=i|\boldsymbol{x}^{(n)}, \theta_2))). \tag{3.7}$$

We often use the **mean disagreement** between all network pairs in an ensemble to provide a global diversity metric.

3.3 Datasets

As mentioned in Section 3.1, we focus on image recognition datasets and their shifted variations, allowing thorough calibration analysis. The first dataset examined, chosen for popularity and fast experimentation, is MNIST (LeCun et al., 2010). The dataset contains 70,000 single-channel, 28 × 28 pixel images of handwritten digits (samples shown in Fig. 3.1a). The images are partitioned into training and test sets of sizes 60,000 and 10,000 respectively. We additionally split off a set of 6,000 randomly selected training samples to be used for validation.



(a) Original samples

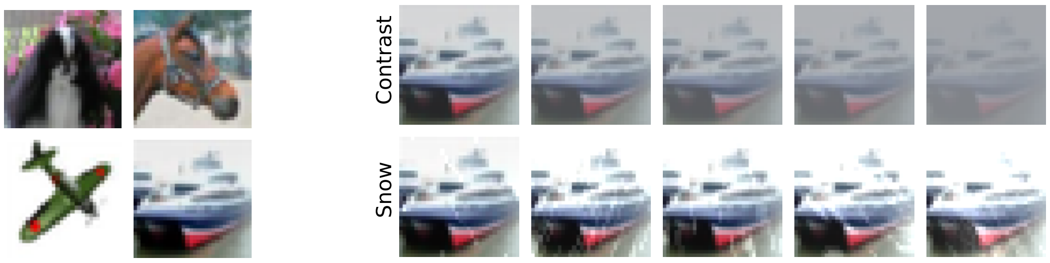
(b) Sample shifted by increasing shift values

Fig. 3.1 Visualisation of MNIST samples, as used throughout the project. On the left a random set of images from the original test set is shown, on the right – single sample's increasingly rotated and translated versions.

To model dataset shift, we introduce distortions via rotation and translation. In particular, to test on rotated data with rotation level r, we rotate the original test set images by r° clockwise or anticlockwise (direction chosen at random). To evaluate on translated data with translation level t, we shift the original image left by t pixels, wrapping the excess back around to the right. We evaluate using rotations $r = 15k, 1 \leq k \leq \frac{180}{15}$ and translations $t = 2l, 1 \leq l \leq \frac{26}{2}$, with $k, l \in \mathbb{N}$. A visualisation for a subset of these is provided in Fig. 3.1b.

We also use CIFAR10 and CIFAR100 (Krizhevsky et al., 2009) as examples of more complex image recognition datasets. Both datasets consist of 60,000 RGB 32×32 pixel images, with 50,000 used for a training set and 10,000 – for a test set. CIFAR10 contains images from 10 different classes consisting of highly distinct objects, such as ships or dogs, with a total of 6,000 samples per class – some of these are visualised in Fig. 3.2a. CIFAR100 offers 600 samples per class instead, with images from a total of 100 classes. As for MNIST, we randomly split off 10% of each training set to form a validation set for tracking generalization throughout training and hyperparameter selection.

To simulate shift for these datasets we employ the corruptions suggested by Hendrycks and Dietterich (2019). While synthetic, they aim to emulate data shifts that can be encountered in natural data. There are 19 (15 main and 4 supplementary) types of algorithmically generated corruptions, ranging from various types of blur, brightness and contrast changes, to simulated natural obstructions, like rain or fog. Each corruption can be expressed at intensity levels ranging from 1 to 5. When evaluating model performance on shifted data, we use unseen images (the test set) with corruptions of a given intensity applied. We refer to Fig. 3.2b for a visualisation of a subset of the corruption types.



(a) Original samples

(b) Single sample corrupted at increasing intensity levels

Fig. 3.2 Visualisation of CIFAR10 samples. On the right a random set of images from the original test set is shown, on the left – a single sample's corrupted variations with corruption intensity increasing left to right for two corruptions: contrast change and snow obstruction. These are also representative of CIFAR100 samples as it has equivalent image and shift types.

3.4 Predictor Networks

As our goal is to eamine the calibration of DEs and their variations, we do not require state of the art performance. To avoid excessive computational overhead we choose common and well-established network architectures to serve as baseline predictors. These are then either evaluated individually or combined, before or after training, via the various forms of ensembling.

The baseline predictor model used for the MNIST dataset is LeNet5 (LeCun et al., 1998) – a relatively small and simple convolutional network designed for this dataset. It employs a combination of convolutional and fully-connected layers, as well as average pooling (illustrated in Fig. 3.3). The network is commonly used as a baseline for predictive performance, and training can be carried out quickly. The architecture has also been applied in the context of studying calibration by Ovadia et al. (2019); Wenzel et al. (2020) and others.

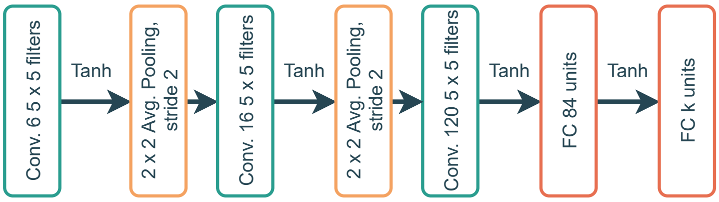


Fig. 3.3 Architecture of LeNet5.

We briefly trialled a custom architecture – a multi-layer perceptron consisting of three hidden layers with ReLU activations, 200 units each and batch normalisation after every fully-connected layer. Initial experimentation indicated the performance and calibration trends were equivalent for the two architectures (see Appendix A). To minimise the computational load, further experiments use LeNet5 predictors.

We use ResNet-20 models as base predictors for the CIFAR datasets. This architecture was introduced by He et al. (2016), alongside the more well-known versions ResNet-18 and ResNet-50. The latter two, however, are adapted to ImageNet (Deng et al., 2009). We instead use a version suggested by the authors specifically for use with the CIFAR10 dataset. It consists of an initial convolutional layer, 3 sets of 3 residual blocks (as shown in Fig. 3.4), using values of $f = 32, 16, 8$ for each set respectively, a global average pooling layer, and a fully-connected prediction layer. Each convolution is followed by batch normalisation. This is another common choice for studying network calibration on image data and DEs, employed by Fort et al. (2019); Ovadia et al. (2019) and others.

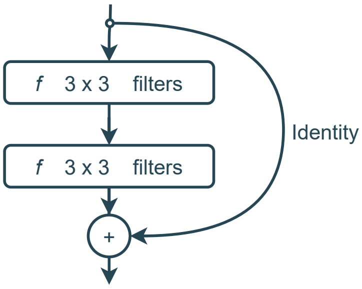


Fig. 3.4 Residual block, as used in ResNet-20. Figure based on He et al. (2016).

3.5 Hyperparameter Selection and Training

Where fully or semi-independent training is possible the predictors are trained using a fixed set of hyperparameters. This is done to compare model performance and calibration in maximally equivalent settings. Where this is not reasonable – for example, when baseline predictors cannot be trained separately in an end-to-end mixture of experts models, an additional random search is performed. All experiments are run with 3 different random seeds (results are non-deterministic due to randomness in initialisation and dataset shuffling). Fully-connected and convolutional layers are initialised using the Kaiming uniform (He et al., 2015) distribution, unless stated otherwise.

LeNet predictors are trained using the Adam optimizer (Kingma and Ba, 2014), with learning and weight decay rates chosen by random search, optimising for validation set performance. The batch size is kept constant at 128, and models are trained for 40 epochs.

For ResNet-20 models, we use the hyperparameters suggested by He et al. (2016): training via stochastic gradient descent, a weight decay rate of 0.0001, momentum of 0.9, batch size 128 and an initial learning rate of 0.1, decayed by a factor of 10 after 90 and 135 epochs. Training is carried out for 180 epochs in total. Parameters of convolutional and fully-connected layers are initialised using the Kaiming normal (He et al., 2015) distribution.

All ensemble models studied have a constant size ($M = 5$). The decision to fix this ensemble size was made in line with literature indicating that the calibration results are consistent even for relatively small DEs. In particular, Ovadia et al. (2019) note that the improvement obtained by increasing ensemble size quickly diminishes, and a five-network ensemble tends to have calibration similar to that of a much larger DE.

3.6 Baselines

To assess the relative performance of the methods examined, we compare it to a set of baseline methods. We include the following:

- A single neural network. We train a single baseline predictor and evaluate its performance on the in-distribution test set, as well as shifted data. This provides the most basic standard we aim to improve over, both in terms of predictive performance and calibration.

- A traditional DE (Lakshminarayanan et al., 2016). We train a set of baseline predictors independently and identically, with random parameter initialisations drawn from the same distributions. The predictions of the individual networks are then aggregated by averaging to produce an ensemble prediction.

- Monte Carlo Dropout (Gal and Ghahramani, 2016). We insert dropout layers after every non-final fully-connected or convolutional layer in the baseline predictors (with a dropout rate of 0.5 for LeNet5 and 0.1 for ResNet-20, consistent with Ovadia et al. (2019)). Dropout remains enabled in testing – predictions from 50 forward passes are averaged to produce the final prediction. As noted in Section 2.1, it provides a simple Bayesian approach to uncertainty estimation.

The networks are trained using the cross-entropy loss (equivalent to the negative log-likelihood) and the procedure specified in Section 3.5.

These baselines do not cover the full variety of methods for improving calibration and uncertainty estimation of deep neural networks. Some notable omissions include other Bayesian neural networks, and post-hoc calibration methods, such as temperature scaling. It has been shown by Ovadia et al. (2019) that the baselines selected outperform – in terms of predictive performance and calibration under dataset shift – a wide variety of methods, including the aforementioned ones. We thus claim comparison to the chosen methods throughout the study provides a comprehensive perspective while maintaining focused analysis.

Chapter 4

Diversity Regularised Deep Ensembles Under Dataset Shift

In this chapter, we take a closer look at two methods for explicit DE diversification via regularisation – negative correlation learning (Liu and Yao, 1999; Shui et al., 2018) and pairwise cross-entropy between ensemble members (Opitz et al., 2016). We introduce the methods and the analysis framework, as well as modifications to the general experimental setup outlined in Chapter 3. We then provide experimental results, analysing the impact of such regularisation and its scaling. We find that while the proposed methods are capable of improving the ensemble calibration and diversity level, their application in practice might be limited. In particular, the regularisation weight is difficult to tune, with performance on an in-distribution validation set not entirely indicative of calibration under distribution shift.

4.1 Methods for Diversity Regularisation

As discussed in Section 2.4.2, employing regularisation terms is the most common strategy for explicit DE diversification. In this dissertation we focus on methods that allow for semi-independent training of ensemble members, maintaining an overall procedure that is nearly identical to DE training. In particular, training can still be easily parallelised, with independent forward and backward passes – data sharing is only needed for loss computation.

As established in Chapter 3, we train k-class classification ensembles using the cross-entropy loss with one-hot targets. Thus, with the k-dimensional output vector for network i and input $\boldsymbol{x}$, denoted $\boldsymbol{o}_i(\boldsymbol{x})$, it's j'th element denoted $\boldsymbol{o}_i(\boldsymbol{x})_j = p(y = j|\boldsymbol{x}, \boldsymbol{\theta}_i)$ and ground truth vector $\boldsymbol{y}$, the loss is

$$L_i(\boldsymbol{x},\boldsymbol{y}) = \mathrm{CE}(\boldsymbol{y},\boldsymbol{o}_i(\boldsymbol{x})) = -\sum_{j=1}^{k} y_j \log\left(o_i(\boldsymbol{x})_j\right). \tag{4.1}$$

We can then define and add a regularising term $R_i(\boldsymbol{x})$:

$$L_i^*(\boldsymbol{x},\boldsymbol{y}) = L_i(\boldsymbol{x},\boldsymbol{y}) + \lambda_i \cdot R_i(\boldsymbol{x}). \tag{4.2}$$

Here λ_i denotes a hyperparameter controlling the relative impact of $R_i(\boldsymbol{x})$. To simplify and minimise the number of hyperparameters, we hold $\lambda = \lambda_i$ constant for all ensemble members. While $R_i(\boldsymbol{x})$ may depend on network outputs other than $\boldsymbol{o}_i$, these values are treated as constant during backpropagation, and only the relevant term is used to update the weights of network i.

We hypothesise that regularisation may have a stronger impact in the early stages of training. This is motivated by the traditional DE setup, with diversity induced purely by random initialisation. Strongly emphasising the diversity inducing term early in the training, when network predictions are still mostly uninformed, can be seen as further diversifying these starting points. However, using a large scaling value λ_i throughout training might cause over-regularisation and worsen predictive performance. To avoid this, we propose gradually reducing the scaling factor via exponential annealing based on the current epoch n_e. We investigate this approach by extending the framework to involve a decay term d:

$$L_i^*(\boldsymbol{x},\boldsymbol{y}) = L_i(\boldsymbol{x},\boldsymbol{y}) + \lambda_i \cdot d^{n_e} \cdot R_i(\boldsymbol{x}). \tag{4.3}$$

Both methods we study in detail follow this framework but present distinct interpretations of the regularisation function R_i. These are described in the following two sections.

4.1.1 Negative Correlation Regularisation

Negative correlation learning has been present in ensembling literature for a long time (Liu and Yao, 1999). The standard form uses a formulation specific to one-dimensional outputs o_i and labels y, and is typically seen in a regression context as

$$R_i(\boldsymbol{x}) = (o_i(\boldsymbol{x}) - \bar{o}(\boldsymbol{x}))\left(\sum_{j\neq i}(o_j(\boldsymbol{x}) - \bar{o}(\boldsymbol{x}))\right). \tag{4.4}$$

Letting M denote the number of networks in the ensemble, $\bar{o}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} o_m(\boldsymbol{x})$.

This term, while not directly corresponding to the formal definition of correlation, provides a measure of how different the predictions of the subnetworks are relative to the ensemble mean. Temporarily disregarding the dependence of $\bar{o}(\boldsymbol{x})$ on the individual predictions, we note that for a particular subnetwork i, $R_i(\boldsymbol{x})$ is minimised when all the other predictions are on the "other side" of $\bar{o}(\boldsymbol{x})$, encouraging them to be low when $o_i(\boldsymbol{x})$ is high and vice versa – making them negatively correlated. The term is non-positive and can be rewritten:

$$\begin{aligned} R_i(\boldsymbol{x}) &= (o_i(\boldsymbol{x}) - \bar{o}(\boldsymbol{x})) \left(\sum_{j \neq i} (o_j(\boldsymbol{x}) - \bar{o}(\boldsymbol{x})) \right) \\ &= (o_i(\boldsymbol{x}) - \bar{o}(\boldsymbol{x})) \left((M\bar{o}(\boldsymbol{x}) - o_i(\boldsymbol{x})) - (M-1)\bar{o}(\boldsymbol{x}) \right) \qquad (4.5) \\ &= (o_i(\boldsymbol{x}) - \bar{o}(\boldsymbol{x}))(\bar{o}(\boldsymbol{x}) - o_i(\boldsymbol{x})) \\ &= -(o_i(\boldsymbol{x}) - \bar{o}(\boldsymbol{x}))^2. \end{aligned}$$

This gives an alternative intuitive interpretation for a minimising solution – ensemble member predictions should be far from the mean, thus inducing diversity.

Shui et al. (2018) proposed to use the formulation directly in the context of classification ensembles, claiming it can improve DE accuracy and calibration on ID testing data in terms of ECE. However, the exact form of the penalty, as adapted for multi-element output is not given, and the calibration is not studied in a context of distribution shift. To address the first point, we suggest the most direct adaptation to the context of vector outputs $\boldsymbol{o}_i(\boldsymbol{x})$ replaces multiplication with dot products as follows:

$$\begin{aligned} R_i(\boldsymbol{x}) &= (\boldsymbol{o}_i(\boldsymbol{x}) - \bar{\boldsymbol{o}}(\boldsymbol{x}))^\top \left(\sum_{j \neq i} (\boldsymbol{o}_j(\boldsymbol{x}) - \bar{\boldsymbol{o}}(\boldsymbol{x})) \right) \qquad (4.6) \\ &= (\boldsymbol{o}_i(\boldsymbol{x}) - \bar{\boldsymbol{o}}(\boldsymbol{x}))^\top (\bar{\boldsymbol{o}}(\boldsymbol{x}) - \boldsymbol{o}_i(\boldsymbol{x})) \\ &= -||\boldsymbol{o}_i(\boldsymbol{x}) - \bar{\boldsymbol{o}}(\boldsymbol{x}))||^2. \qquad (4.7) \end{aligned}$$

The replacement is well-founded – dot product (for vectors of a fixed length) is minimised when the vectors denote opposite directions so this will encourage diversity in the deviations of network predictions from their mean.

An alternative formulation can be derived as before and is given in Equation 4.7. The term can again be seen as encouraging the probability vectors produced by individual networks to be far from the mean vector in terms of Euclidean distance.

We use the vector-specific formulation for training negative correlation regularised deep ensembles (NCL-DE). When training a specific network with output $\boldsymbol{o}_i(\cdot)$, its contribution to the mean prediction is ignored for backpropagation, in line with Liu and Yao (1999), with both $\bar{\boldsymbol{o}}(\cdot)$ and the predictions of other networks treated as constants.

4.1.2 Pairwise Cross-Entropy Regularisation

The second method we examine is instead specific to ensembles of classification networks and relies on the output interpretation as a categorical distribution. In their work on efficient ensembling, Opitz et al. (2016) proposed regularising the training loss using the sum of pairwise cross-entropies between the individual predictions made. This was done to improve generalisation in terms of accuracy on an ID testing set. However, we hypothesise it should have an effect comparable to that of negative correlation learning in terms of uncertainty calibration.

In particular, we propose adapting the regularisation term used by (Opitz et al., 2016) to fit the framework for diversity regularisation (Equation 4.2) by setting

$$R_i(\boldsymbol{x}) = \frac{1}{M-1} \sum_{j \neq i} -\mathrm{CE}(\boldsymbol{o}_i(\boldsymbol{x}), \boldsymbol{o}_j(\boldsymbol{x})). \tag{4.8}$$

The key adjustment we make is inducing separation, to allow regularisers specific to different subnetworks to be used individually. DEs trained using this form of regularisation are referred to as CE-DE.

This is a compelling diversity regularisation method due to how widespread the cross-entropy function is in machine learning. It is the go-to loss for training classification networks and a popular indicator of similarity between categorical distributions. Minimising the negative cross-entropy between a given predictor's output and the predictions made by other subnetworks should thus have a diversifying effect in the probability distributions produced.

4.2 Improvement Achievable Under Ideal Conditions

To assess the strategies, it is important to understand the potential performance in terms of calibration achievable using NCL-DE and CE-DE under ideal conditions. In this section, we compare the results for the best hyperparameter settings of the annealed and constant-regularised methods and baselines outlined in Section 3.6. To illustrate ideal performance, diversification hyperparameters are chosen by *actual performance on shifted test data for each dataset*, with the additional requirement of competitive ID accuracy.

4.2.1 Results for MNIST Data

Figures 4.1, 4.2 and 4.3 summarise NCL-DE and CE-DE performance on the variations of MNIST testing data. Figures 4.1 and 4.2 show the metric values under progressive image rotation and translation respectively. Two corresponding tables are provided in Appendix B (Table B.3 and Table B.4), providing numerical results for select shift levels. Notably, all the methods tested consistently outperform a single network in terms of calibration metrics for shifted data. Furthermore, all included regularised methods have lower ECE, NLL and Brier score values than traditional DEs. While the improvements observed are more significant under shift by translation, the trend is consistent across the two dataset shift types.

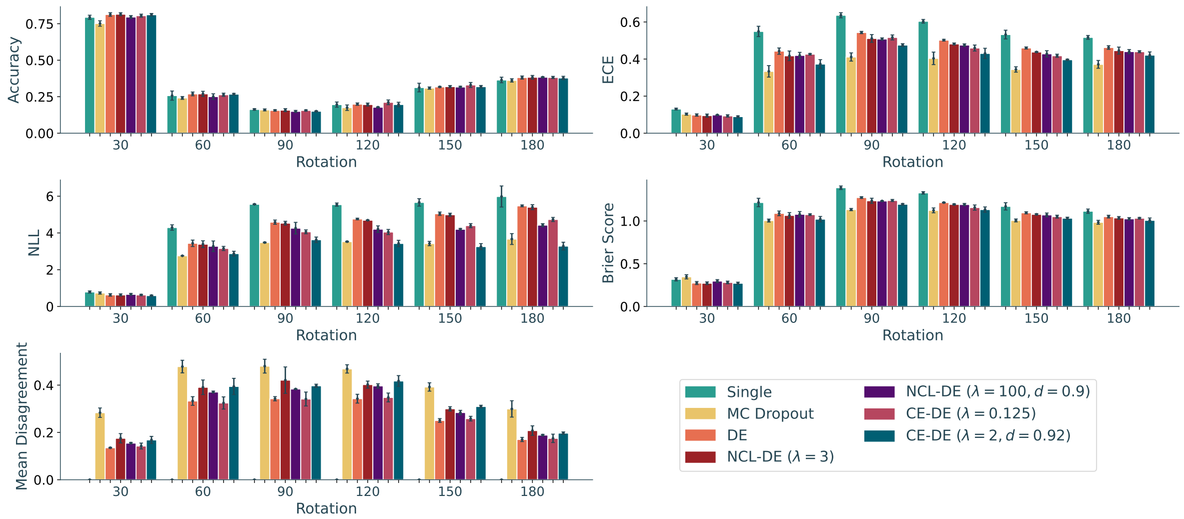


Fig. 4.1 A comparison of baseline methods – a single DNN ("Single" in legend), MC Dropout, and DE – with best-performing regularised alternatives (notation as per Equation 4.3) on rotated MNIST test images. The x-axis labels denote the rotation level in degrees, bar height – the metric value. Dark lines show the 95% confidence interval obtained from 3 independent runs.

The methods using decayed regularisation terms outperform the alternatives with fixed values of λ, supporting the earlier hypothesis on the importance of early diversification. The heavy initial regularisation does not have a significant adverse effect on the ensemble's accuracy, with the values being close and falling within error bars for most methods.

Additionally, the regularisation terms increase the diversity of network predictions in function space. This is evidenced by the mean disagreement, which is consistently higher for the regularised DE variations. Thus the metrics used to quantify diversity in regulariser design have the intended effect.

As noted by Ovadia et al. (2019), on this particular shifted dataset MC Dropout is often more effective in improving calibration than other methods. The level of disagreement between

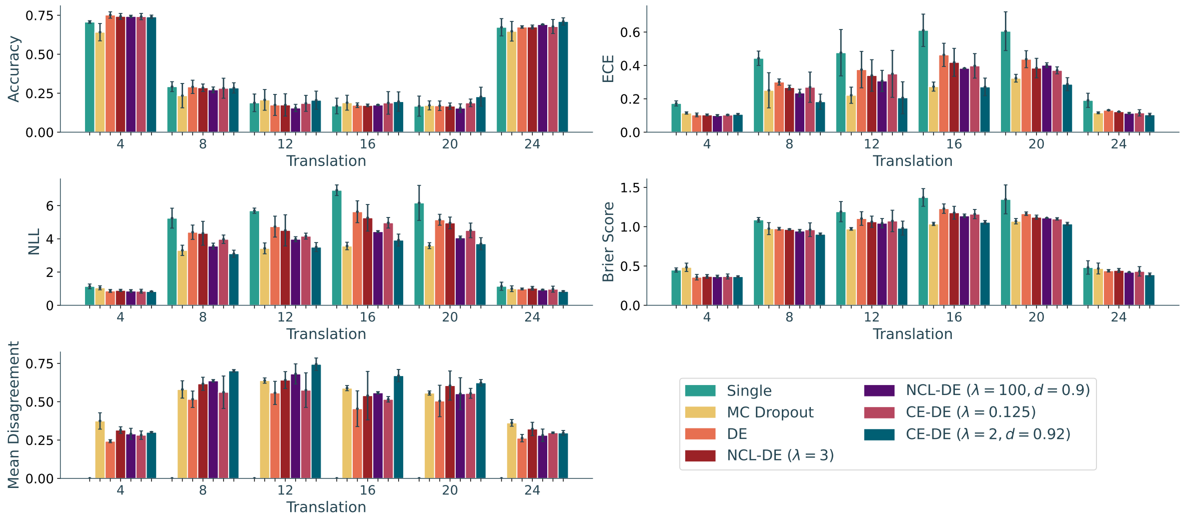


Fig. 4.2 A comparison of baseline methods with best-performing regularised alternatives on translated MNIST test images. The x-axis labels denote shift in pixels, bar heights – the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

predictions illustrates that the large dropout rate ($p = 0.5$) we can use here induces highly diverse subnetwork predictions. The disagreement is consistently higher than for DEs and, for rotated data, their regularised variants. However, the randomness induced causes a slight accuracy drop.

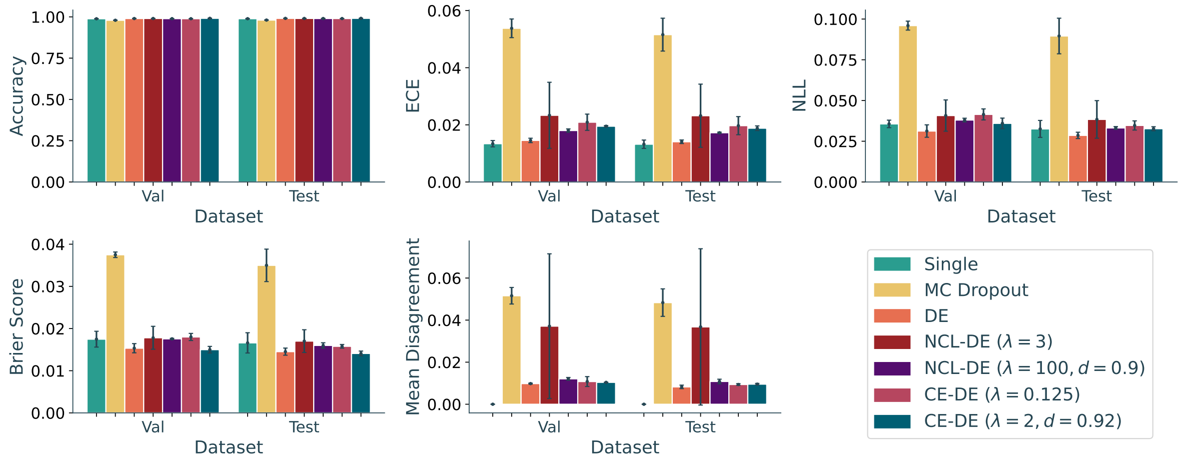


Fig. 4.3 A comparison of baseline methods with best-performing regularised alternatives on ID MNIST test images. Bar height equals the relevant metric value, while dark lines show the 95% confidence interval as obtained from 3 independent runs.

In-distribution results, shown in Figure 4.3, with numerical values in Table B.2, provide a somewhat different perspective. We observe that regularised methods chosen by their performance on shifted data tend to perform slightly worse than DEs, in at least one calibration metric on ID sets (validation and testing). This can be explained by the slight increase in network disagreement on ID data. The dataset considered is simple, with very high accuracy achieved ($\approx 99\%$ test accuracy), so any drop in confidence induced by diversification can cause slight miscalibration. As on shifted data, regularised methods utilising scaling term decay perform slightly better than those using constant λ, providing a better trade-off between over and under-regularising.

Notably, the base network used here, LeNet5, is not very deep and, as noted by Guo et al. (2017), such classic networks were better calibrated on ID data than modern alternatives. This is reflected in our results, with the lowest ECE values achieved by a single predictor. However, we have shown this does not hold under dataset shift and both the use of DEs and their regularised variations are beneficial for calibration in these conditions.

4.2.2 Results for CIFAR Data

The CIFAR10 and CIFAR100 datasets are both larger, with shifts emulating natural ones more closely than those used for MNIST. Furthermore, Ovadia et al. (2019) found trends observed here to be representative of even more complex datasets, such as ImageNet (Deng et al., 2009).

Fig. 4.4 illustrates the result for the CIFAR10 dataset. As before, NCL-DE and CE-DE tend to have slightly lower median ECE, NLL and Brier score values than traditional DEs, particularly for severe dataset shift intensities. In these cases, accuracy is also slightly increased. Unlike for MNIST, these approaches typically obtain the best scores (as further illustrated by Table B.5 in Appendix B) – MC Dropout no longer outperforms DEs. This is likely due to the significantly lower dropout probability ($p = 0.1$) used to ensure competitive predictive performance. As a result, the diversity of MC Dropout predictions is lower than that of DEs and their variations, producing poorer calibration.

Fig. 4.5 illustrates equivalent trends on the CIFAR100 dataset, although we no longer observe a consistent improvement in the accuracy for severely shifted data. The differences in calibration metric values are also smaller, but still present as seen in numerical results in Appendix B.

The regularised methods are more robust to the type of dataset shift applied on both datasets – the range of values obtained for ECE, NLL or Brier score is typically smaller than that of DEs. This is a desirable property and suggests the NCL-DE and CE-DE might generalise better to unseen dataset shift. The effect is stronger on CIFAR10, but still present on CIFAR100.

A notable difference from the trends on MNIST lies in the comparison of constant and annealed regularisation. Here the differences between the best results for the two strategies are much smaller. While for instance the annealed version of NCL-DE consistently outperforms the constant-regularised alternative in terms of the mean metric values across all shifts of a certain intensity, the difference is less clear for the medians. This is partially due to the narrower range of values obtained under annealing impacting the mean more directly.

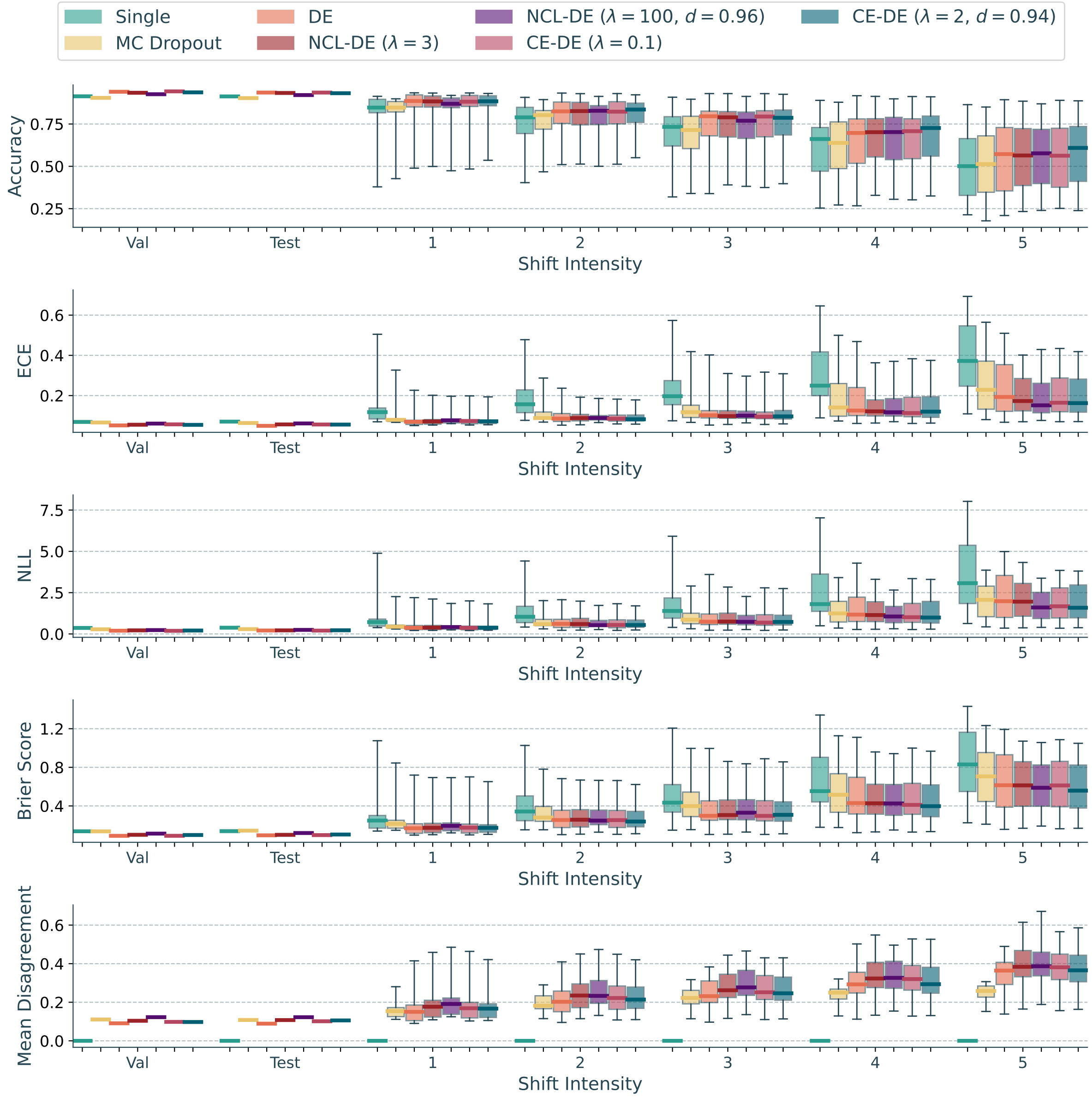


Fig. 4.4 A comparison of baseline methods with best-performing regularised alternatives on CIFAR10 data. Box plots summarise the distributions of metric values across different data shift types. Box boundaries indicate 25th and 75th percentiles, thick coloured lines – the median, and whiskers show the full range of values.

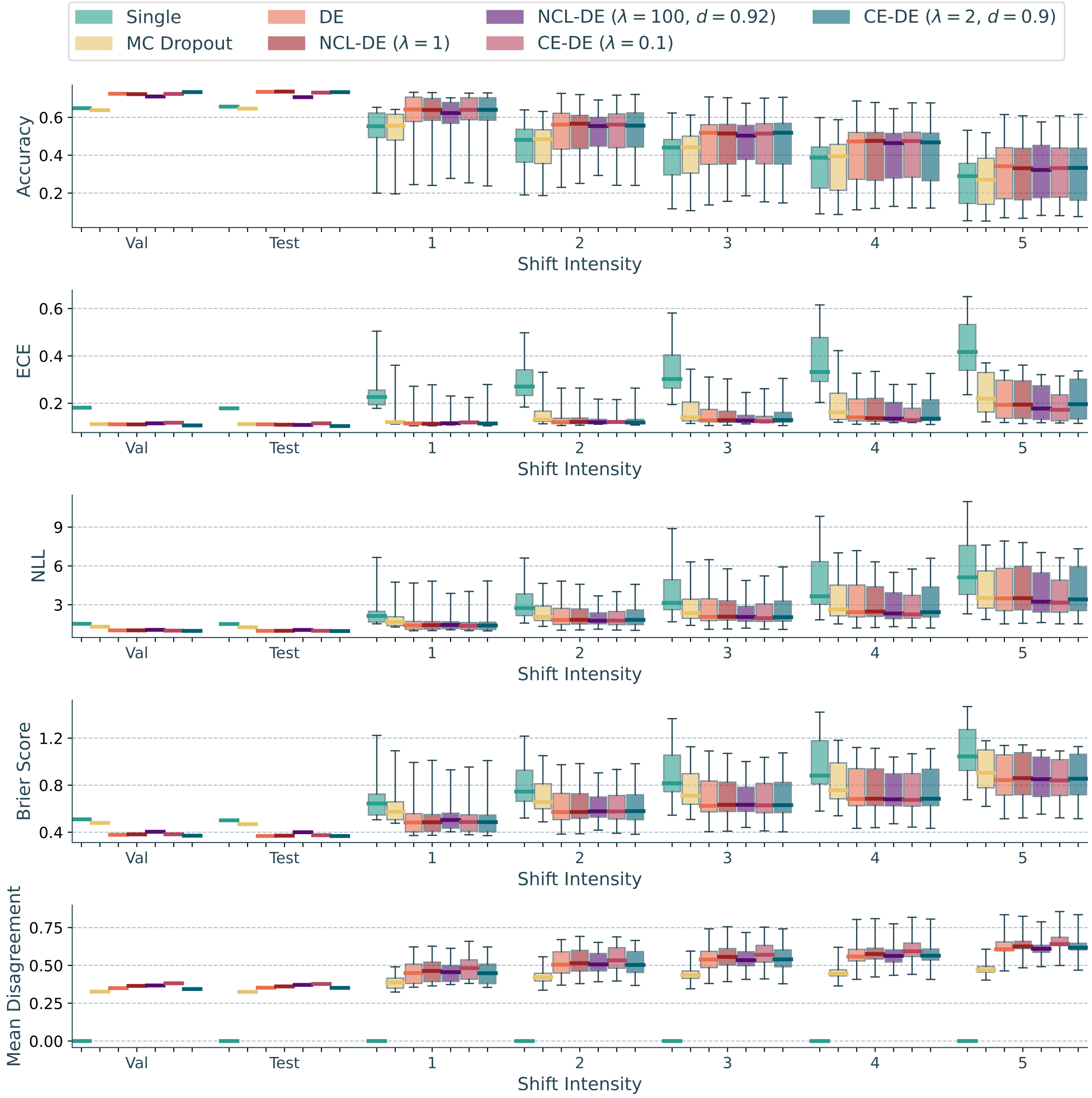


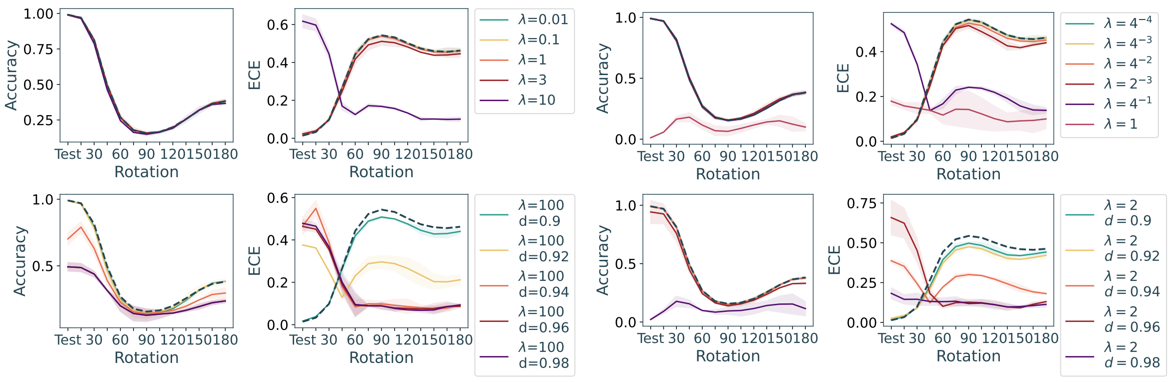
Fig. 4.5 A comparison of baseline methods with best-performing regularised alternatives on CIFAR100 data. Box plots summarise the distributions of metric values across different data shift types. Box boundaries indicate 25th and 75th percentiles, thick coloured lines – the median, and whiskers show the full range of values.

4.3 Sensitivity to Regularisation Weight

While the preceding section illustrates some desirable properties of NCL-DE and CE-DE, the hyperparameters have been carefully selected to display the best possible use case. In practice, this is not always possible. It is particularly challenging to select hyperparameters to guarantee good performance on shifted data. Such samples are typically not available during training and even the type of dataset shift that might be encountered in practice is largely unknown.

Both of the methods studied in this chapter heavily rely on the parameters λ and d. To ensure they can be easily applied in practice, with improvement under the unknown conditions as described above, the sensitivity to particular hyperparameter values should be low. To study this, we investigate how the performance of NCL-DE and CE-DE on both ID and shifted data is affected as the hyperparameter values are varied.

We sweep over logarithmic search spaces of 10^a with $a \in [-2,-1,0,1]$ for NCL-DE and 4^a with $a \in [-4,-3,-2,-1,0]$ for CE-DE in their constant-regularized versions, and investigate $d \in [0.9, 0.92, 0.94, 0.96, 0.98]$ with a fixed λ for the annealed ones. Well-performing ad-hoc values are included for illustrative purposes. We focus on evaluating the performance in terms of the accuracy and ECE – summarising both the predictive performance and calibration. The results are visualised in Fig. 4.6 and Fig. 4.7 for the MNIST and CIFAR datasets respectively.



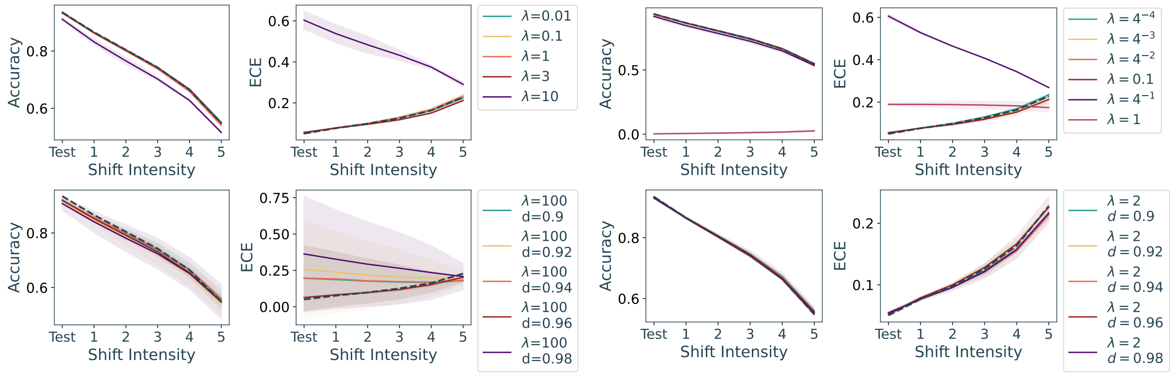
(a) NCL-DE

(b) CE-DE

Fig. 4.6 Performance of regularised methods with different hyperparameters on rotated MNIST data. A row in each subfigure displays accuracy, ECE and legend for the two plots. Solid lines show mean values across 3 runs and all shift types of a given intensity, shaded areas – 95% confidence interval, dashed dark line – mean baseline (DE) performance.

Across both datasets, annealed and constant-weighted versions of NCL-DE and CE-DE, we observe relative large sensitivity to hyperparameter choice. In particular, only a narrow range of values leads to performance improvement over standard DEs on shifted data without causing over-regularisation. The latter manifests either in significantly higher ID ECE values ($\lambda = 10$

for const. NCL-DE in Fig. 4.6a; $\lambda = 4^{-1}$ for const. CE-DE in Fig 4.7b), or a combination of rising ID ECE and a drop in accuracy ($\lambda = 1$ in Fig. 4.6b; $\lambda = 10$ in Fig. 4.7a). On the opposite side of the spectrum, small regularisation terms or quick decay makes the method's performance equivalent to that of a DE.



(a) NCL-DE

(b) CE-DE

Fig. 4.7 Performance of NCL-DE and CE-DE for different hyperparameters on shifted CIFAR10 data. A row in each subfigure displays accuracy, ECE and the legend for the two plots. Solid lines show mean values across 3 runs and all shift types of a given intensity, shaded areas – 95% confidence interval, dashed dark line – mean baseline (DE) performance.

Additionally, annealed NCL-DE tends to exhibit more variability across runs than other strategies, as evidenced by large confidence intervals for most decay values. This may in part be caused by the large initial λ value allowing more randomness to be introduced throughout the training. However, it further emphasises NCL-DE is highly sensitive to hyperparameter choice. While results indicate that, in theory, it is possible to improve over standard DE performance and increase predictor diversity as measured by disagreement using this method, it may be hard to apply in practice.

4.4 Discussion

The methods examined in this chapter, NCL-DE and CE-DE, were originally suggested for improving ID calibration (as measured by ECE) and ID accuracy by Shui et al. (2018) and Opitz et al. (2016) respectively. While the approaches are different, both methods are constructed to explicitly increase the diversity of individual predictors in the ensemble.

We have shown that under appropriate conditions and well-selected hyperparameters diversification can be achieved. This is illustrated by the consistent increase in mean disagreement between predictor pairs – a metric measuring function space diversity. However, the goal is not to diversify blindly – high, but not useful diversity could be achieved by simply randomising predictions. As discussed in Section 4.3, excess diversification can be detrimental in terms of both accuracy and calibration. Particularly, explicit diversity regularisation like NCL-DE and CE-DE optimises for diversity on the ID training data, matching the goal of diversity for OOD samples imperfectly. Thus over-regularisation affects ID data the most, sometimes causing no accuracy drop but inducing high ECE values on the standard test set (e.g. $\lambda = 4^{-1}$ in Fig. 4.6b), often due to the overall confidence level being reduced. While it is desirable for shifted data – and leads to shifted ECE improvement in the example mentioned – ID miscalibration is introduced, particularly for highly accurate predictors, such as models for MNIST and CIFAR10. Evaluating ID data is often models' primary use case and thus such behaviour is undesirable.

This further emphasises the need to select appropriate hyperparameters when using NCL-DE or CE-DE. When the ideal hyperparameter settings for each problem are used (as per Section 4.2) each regularisation strategy tested improves over the traditional DE in terms of calibration under dataset shift. This effect is summarised in Table 4.1, where we report the mean ranks (1 – best, 7 – worst) each method achieves across all levels of dataset shift and the three calibration metrics (ECE, NLL and Brier score). The mean rank of both methods, using either constant or annealed weighting, is lower than that of DE for all datasets examined.

Dataset	Single	MC Dropout	DE	NCL-DE (const.)	NCL-DE (anneal.)	CE-DE (const.)	CE-DE (anneal.)
MNIST	6.24	3.18	4.79	4.63	3.52	3.73	1.92
CIFAR10	7	5.8	4.4	3.4	3.13	1.73	2.53
CIFAR100	7	6	4.87	3.87	1.27	2	3

Table 4.1 Mean method ranks across calibration metrics (ECE, Brier score and NLL), all data shift types and intensities. Values compared as means for 3 runs. NCL-DE and CE-DE use the best hyperparameter settings, as per Section 4.2. "const." indicates constant regularisation scaling, "anneal." – annealed scaling.

Notably, the best hyperparameter settings are somewhat consistent across the datasets, particularly for constant-weighted regularisation. For CE-DE the best results are achieved for $\lambda = 0.1$ on both the CIFAR datasets and $\lambda = 0.125$ for MNIST. NCL-DE achieves the best performance for $\lambda = 3$ on MNIST and CIFAR10 and $\lambda = 1$ for CIFAR100. This suggests that despite the sensitivity to hyperparameters, the optimal range of values is relatively consistent.

However, as shown by Fig. 4.3, these parameter settings are not always the best for ID data. Indeed, if the hyperparameters for NCL-DE and CE-DE are instead selected by the performance on a validation set, the outlook is different. Table 4.2 summarises the mean ranks in this case, with ECE used as a heuristic for selection. DEs are no longer consistently outperformed by their variations with constant regularisation scaling. This is due to the lowest regularisation terms tested being selected in both cases, making the performance on shifted data near-indistinguishable from that of DEs. The annealed versions of NCL-DE and CE-DE still consistently perform better than DEs, however, this can be partially attributed to the limited range of values tested. As in Section 4.3, we select from $d \in [0.9, 0.92, 0.94, 0.96, 0.98]$ for a fixed initial λ, constraining the choices to only methods with strong initial regularisation. When NLL or Brier score is used to select methods via their validation performance, the overall results are improved, however, no strategy can recover the best settings.

Dataset	Single	MC Dropout	DE	NCL-DE (const.)	NCL-DE (anneal.)	CE-DE (const.)	CE-DE (anneal.)
MNIST	6.39	3.06	4.18	4.86	3.38	4.08	2.05
CIFAR10	7	5.67	3.33	3.6	2.53	4	1.87
CIFAR100	7	6	4.6	3.4	1.07	3	2.93

Table 4.2 Mean method ranks across calibration metrics (ECE, Brier score and NLL), all data shift types and intensities when NCL-DE and CE-DE use hyperparameters selected by validation ECE. Values compared as means for 3 runs. "const." indicates constant regularisation scaling, "anneal." – annealed scaling.

Despite poor performance for shifted data, selecting hyperparameters via their validation ECE allows us to confirm results reported by the original works proposing the respective methods. Under this selection strategy slight but consistent improvements in the ID test set accuracy and ECE values can be observed (detailed values provided in Appendix B, Table B.1).

In addition to exploring NCL-DE and CE-DE performance under dataset shift, we also started with the hypothesis that using a large regularisation weight in the initial stages of training and gradually annealing it throughout training can improve final calibration. This is corroborated by the results for the MNIST dataset – we note a significant improvement when annealing is used. It is further illustrated by the ranks reported in Table 4.1. However, here we also see that the trend does not hold in the larger datasets, where more natural corruptions are applied. While NCL appears to be improved by the annealing, constant-weight CE-DE performs better than the alternative. Despite this, we have also shown NCL-DE is quite sensitive to the annealing term and the overall result can vary significantly across runs. We can thus neither confirm nor reject our hypothesis. We suggest it might be a promising strategy, particularly for NCL-DE, although it can also increase the variability.

Furthermore, Table 4.1 suggests CE-DE is the preferred strategy in most cases. The cross-entropy based diversity definition did not require any adaptation to the multi-output setting and may be more suitable for diversifying classification ensembles. However, we have shown that both methods have the potential to improve DE calibration, although more robust strategies for selecting hyperparameter values are needed.

Chapter 5

Calibration in Mixtures of Experts

The MoE paradigm has fallen somewhat out of use within the machine learning community as deep learning's popularity has increased. However, some recent work has utilised related methods. In particular, MoE-inspired layers using sparse gating to select a subset of experts to pass the input through, are used to boost model capacity and performance. Shazeer et al. (2017) and Riquelme et al. (2021) report state-of-the-art accuracy alongside reduced computational cost in models employing such strategies, illustrating their potential. However, the impact of employing MoEs has not been, to our knowledge, extensively studied in the context of calibration. In this chapter, we conduct such analysis for ID and shifted data, using the evaluation framework established in earlier chapters. We also investigate strategies to improve MoE model calibration by using Bayesian gating networks.

5.1 Why Study Mixtures of Experts?

We believe the study of MoE calibration is important and highly related to the improvements achievable via DE diversification, as examined in Chapter 4. Due to (explicit or implicit) expert localisation, MoE training typically produces highly diverse ensembles. Thus benefits of varied predictions are fully available in this setting, without the need for explicit diversity quantification and regularisation. However, it also emphasises the role of the gating network, as simply averaging network predictions has the potential to negatively affect the ensemble's accuracy. On the other hand, if the gating collapses to selecting a single expert for each sample we might run into miscalibration due to the over-confidence of individual experts.

Employing a gating network can in theory improve an ensemble's performance. This can be seen in Table 5.1 which illustrates the difference between ensemble and oracle accuracy (Lee et al., 2016). The latter corresponds to accuracy when the network prediction with the lowest loss is picked as the ensemble output, illustrating how often at least one predictor in the ensemble is correct. The results are given for both DEs and an equivalently sized set of explicitly localised networks (trained on subsets of data selected by class).

	MNIST	CIFAR10	CIFAR100
DE Ensemble Accuracy	98.97	93.6	72.99
DE Oracle Accuracy	99.51	97.73	85.24
Localised Ensemble Accuracy	40.58	54.25	57
Localised Oracle Accuracy	99.76	98.72	86.18

Table 5.1 Ensemble (predictions – means of individual outputs) and oracle (predictions – best individual outputs) accuracy for DEs and an equivalent number of explicitly localised experts.

DE oracle accuracy being higher than ensemble accuracy is expected. Although a DE can classify a sample correctly without individual predictors being right (e.g. when the highest probability is assigned to a different class by each predictor, but the second-highest always corresponds to the ground-truth class), it is not common. However, the difference between ensemble and oracle accuracy on the CIFAR datasets is quite large, suggesting that even DEs might benefit from appropriate gating, at least in terms of predictive performance.

In addition to this, we note that using localised experts consistently increases the oracle accuracy, although mean prediction accuracy falls with specialisation. It was also observed by Lee et al. (2016), who propose using multiple-choice learning to achieve implicit localisation. This, however, does not provide a reliable way of producing accurate ensemble predictions. We propose MoEs might serve as effective alternatives by training the gating network alongside localised experts.

5.2 Mixture of Experts Model Architecture

We focus on MoE models where both the experts and the gating predictor are DNNs. The base predictors described in Section 3.4 are used as experts, together with a selection of gating network architectures.

The first option for gating network design is simply re-using the same type of network as the one chosen for experts (referred to as Exp. gating). It limits the design decisions required, does not introduce any additional variables and ensures the network is well-suited to the input data. In addition to this, we examine the effect of using a relatively small gating network, with approx. 15 000 parameters (compared to nearly 300 000 parameters in a ResNet-20). The network has a simple convolutional architecture (Fig. 5.1, referred to as Conv. gating), making it well-suited for the image data used. We also include a simple multi-layer perceptron gating network. Referred to as MLP gating, it takes the flattened pixel values as input and has a single

ReLU activated hidden layer of 100 units followed by batch normalisation. While this network has a large number of parameters, it offers the most generic architecture option.

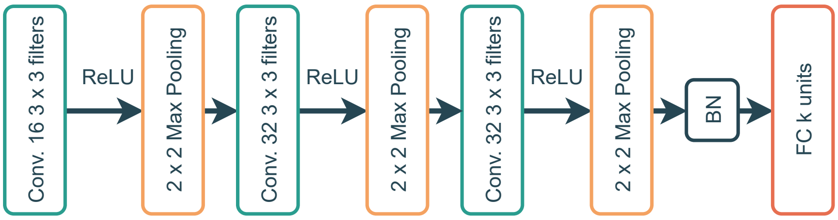


Fig. 5.1 Custom gating network architecture used in MoE experimentation. Conv. refers to a convolutional layer, BN – to batch normalisation, FC – a fully connected layer.

As per the general MoE setup, for a sample $\boldsymbol{x}$ and a mixture of M experts with individual outputs $\boldsymbol{o}_m(\boldsymbol{x})$ and gating weights $\boldsymbol{g}(\boldsymbol{x})$, the ensemble prediction is

$$\bar{\boldsymbol{o}}(\boldsymbol{x}) = \sum_{m=1}^{M} g(\boldsymbol{x})_m \cdot \boldsymbol{o}_m(\boldsymbol{x}). \tag{5.1}$$

As mentioned in Section 2.5, the gating network is typically implemented with a final softmax layer, ensuring the entries sum to one. The ensemble prediction is thus a weighted mean of individual outputs.

5.3 Training Mixtures of Experts

To experiment with MoEs, we need to establish a training procedure. While the typical early strategies for MoE use the expectation-maximization (EM) algorithm (Masoudnia and Ebrahimpour, 2014) for training, recent advances in deep learning allow backpropagation based training to be used efficiently.

We aim to utilise the implicit expert localisation offered by training the gating network alongside the experts, rather than explicitly dividing the dataset. This allows for degenerate gating to develop – the gating network always giving a high weight to the same network, emulating training a single predictor, or always weighting all networks equally, emulating a DE. However, it does not require domain knowledge and presents the most universal approach.

We conduct end-to-end training, using a single optimiser for the experts and the gating network, with all parameters updated together. We consider two loss functions: the overall ensemble loss

$$L_{ENS}(\boldsymbol{x},\boldsymbol{y}) = \mathrm{CE}(\boldsymbol{y},\bar{\boldsymbol{o}}(\boldsymbol{x})) \tag{5.2}$$

and the weighted sum of individual network losses,

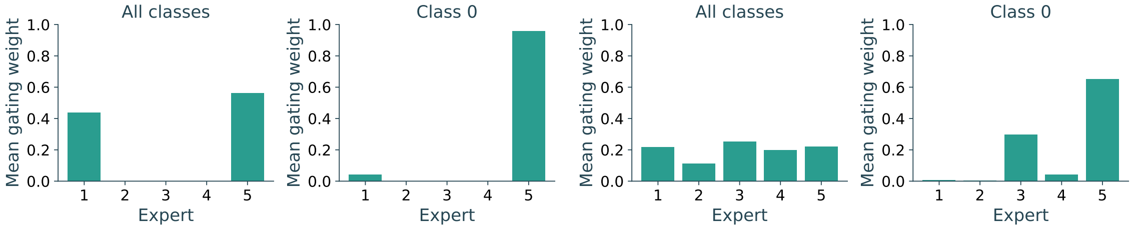
$$L_{SUM}(\boldsymbol{x},\boldsymbol{y}) = \sum_{m=1}^{M} g(\boldsymbol{x})_m \cdot \mathrm{CE}(\boldsymbol{y},\boldsymbol{o}_m(\boldsymbol{x})). \tag{5.3}$$

While L_{ENS} optimises the overall objective directly, L_{SUM} brings the training procedure closer to that of DEs, with expert training being independent aside from the scaling effect of the gating outputs. The latter has also been more widely applied for MoE training, with Jacobs et al. (1991) pointing out it might serve better than the ensemble loss in encouraging localisation.

Load balancing of the gating network is a known issue that arises in such MoE setups (Eigen et al., 2013; Shazeer et al., 2017). As mentioned before, end-to-end training can sometimes induce gating behaviour where all samples are allocated to a small subset of experts. This causes ineffective utilisation of the model, as some of the predictors are not trained. To avoid this we adopt the strategy suggested by Shazeer et al. (2017) and add a batch-wise importance loss term L_I to the overall loss during training. For a batch of samples X, it is computed as

$$L_I = w_I \cdot \mathrm{CV}(Importance(X))^2, \text{ where } Importance(X) = \sum_{\boldsymbol{x}\in X} \boldsymbol{g}(\boldsymbol{x}). \tag{5.4}$$

Here w_I is a tunable hyperparameter (set to 0.1 after a grid search) and CV refers to the coefficient of variation. Minimising L_I encourages similar cumulative weights to be assigned to all experts for each batch. The effect of applying it during end-to-end training of a MoE model with 5 experts on the CIFAR10 dataset is illustrated in Fig. 5.2.



(a) No importance loss

(b) Importance loss with $w_I = 0.1$

Fig. 5.2 Mean gating weight assigned to experts across the CIFAR10 test set in a trained MoE with 5 ResNet-20 models and Conv. gating. For the figure on the left, the model was trained using L_{ENS} only; scaled importance loss was included to generate the results on the right.

The gating network of the model not utilising L_I learns to use only two experts (Fig. 5.2a), with localisation shown by one being favoured for a given class. When L_I is used, the overall allocation is much more even (Fig. 5.2b), however, localisation is maintained, with the weight distribution for samples from a single class differing significantly from the overall result.

We also briefly investigated the hypothesis that balanced gating can be induced by ensuring appropriate network initialisation, bringing the initial gating distribution closer to uniform. However, the load balancing problem is self-reinforcing – networks with training emphasised by large gating weights are trained faster and perform better and the gating network is trained to favour them. Thus even small random deviations eventually lead to imbalanced gating and using an explicit balancing term is more effective.

5.4 Baseline Mixture of Experts Calibration

Throughout this chapter, we focus on two of the three datasets used earlier, MNIST (under rotation shift only) and CIFAR10. The decision is made due to computational constraints and to allow for a more thorough analysis. Additionally, in Chapter 4 we saw the majority of trends are consistent across datasets, thus we deem the results sufficiently illustrative. Methods are also no longer compared to MC Dropout, as we primarily investigate how MoE models compare to DEs.

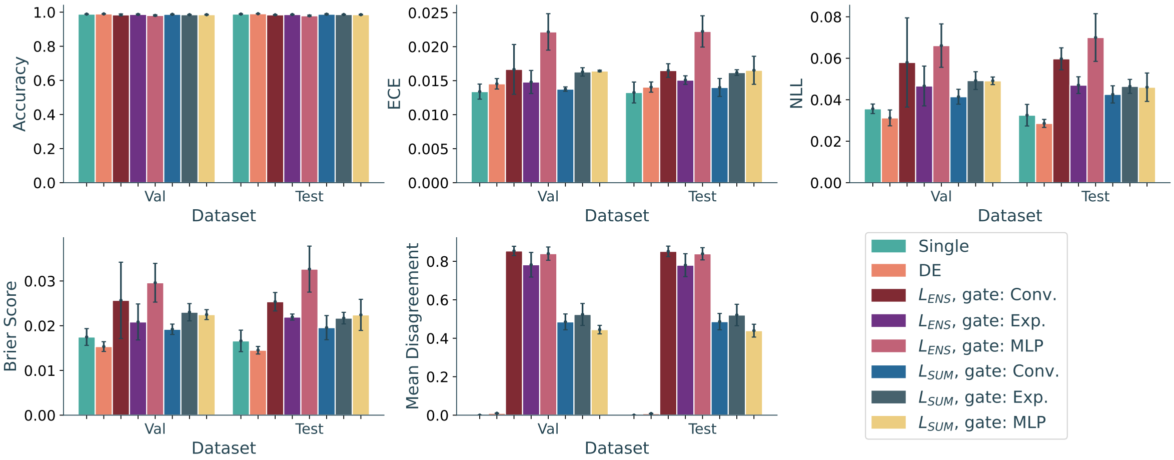


Fig. 5.3 A comparison of baseline methods with MoEs trained with L_{ENS} or L_{SUM} and using different gating networks on ID MNIST test images. Bar heights indicate the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

On ID MNIST data (Fig. 5.3) for all MoE methods the accuracy level remains similar to that of both DEs and a single network – likely due to the simplicity of the dataset investigated. However, the diversity of the MoE ensembles, as measured by disagreement, is very high. This illustrates the extensive localisation induced throughout training, and the importance of the gating network – correct overall predictions are given even when network pairs disagree on 80% of the samples. A similar increase in diversity due to localisation can be seen for ID CIFAR10 data, as illustrated by Fig. 5.5. However, here we also observe a somewhat concerning drop in ID accuracy, with all MoE models under-performing even when compared to a single model – we discuss this in more detail in Section 5.4.1

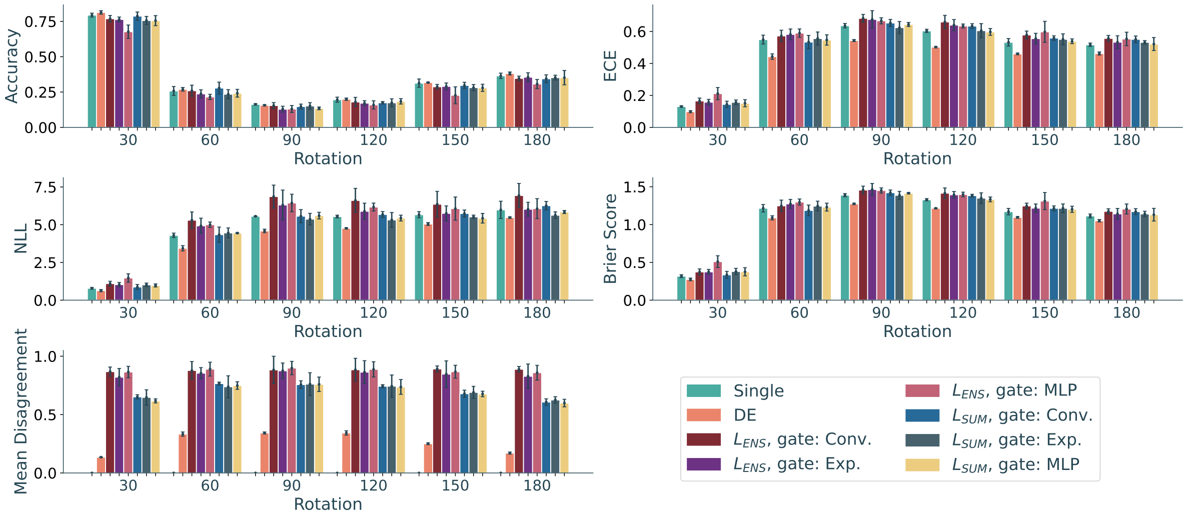


Fig. 5.4 A comparison of baseline methods with MoEs trained with L_{ENS} or L_{SUM} and using different gating networks on rotated MNIST test images. The x-axis labels denote rotation in degrees, bar heights – the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

A general trend, persistent across datasets (seen for MNIST in Fig. 5.3 and Fig. 5.4 and for CIFAR10 in Fig. 5.5) and both ID and shifted data, is poor calibration of MoE models. In terms of NLL, ECE and Brier score the models tend to be equivalent to – or worse than – a single predictor. The models produced are often more over-confident than a single network. This is in part explained by localisation. Due to the use of gating throughout training, the networks are effectively trained on an implicitly selected subset of data, which stabilises as the experts and the gating network improve. This leads to exaggerated overconfidence in their predictions as the effect of other samples is reduced by low gating weights. However, the gating network is also a DNN prone to overconfidence (with mean confidence over the ID MNIST test set of 95.5% and 90.6% over the 60° rotated set, as averaged across the different MoE settings). Typically a single expert is assigned a high score at prediction time for a given sample, allowing the overconfidence to propagate.

Lastly, we compare the performance of MoE models trained using different loss function and gating network combinations. Across both datasets, the ensemble loss L_{ENS} induces a higher level of disagreement than L_{SUM}. On MNIST models trained using L_{SUM} tend to have slightly better calibration than equivalent ones trained using L_{ENS}. However, this trend is not present on CIFAR10. In Fig. 5.5 no consistent differences can be seen for ID or slightly shifted

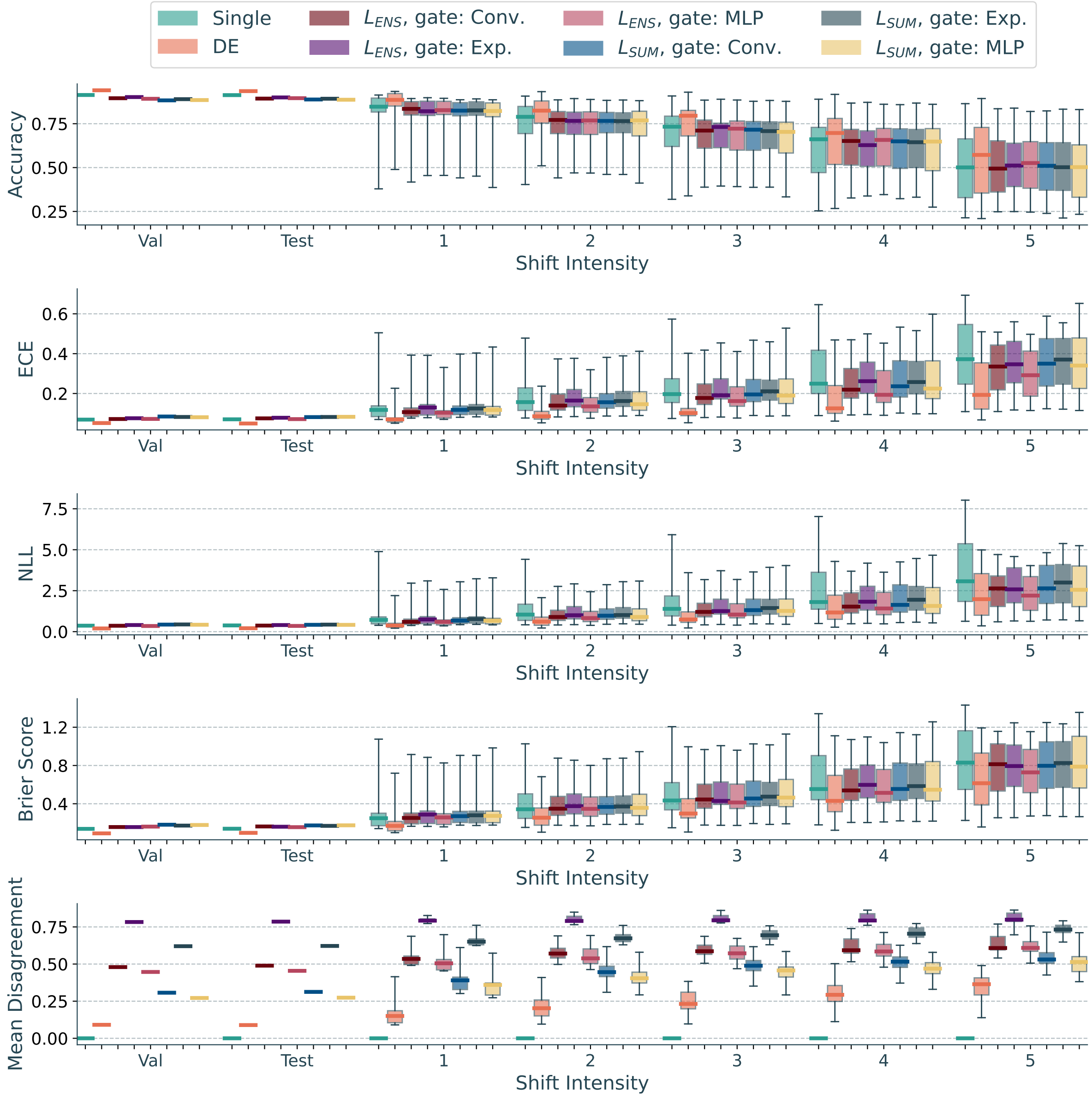


Fig. 5.5 A comparison of baseline methods with MoEs trained with L_{ENS} or L_{SUM} and using different gating networks on ID and shifted CIFAR10 data. Box plots summarise the distributions of metric values across different data shift types. Box boundaries indicate 25th and 75th percentiles, thick coloured lines – the median, and whiskers show the full range of values.

data, and the trend is reversed under more intense shifts. This indicates the loss functions are largely equivalent. However, the difference in diversity is notable. It would be reasonable to expect L_{SUM} – which emphasises individual expert loss values – to encourage localisation more than L_{ENS} which only uses the ensemble prediction, but the opposite is observed. This might also be caused by the high gating network confidence levels – the ensemble prediction largely represents a single expert's output in later stages of training.

For the CIFAR10 dataset, the MLP gating achieves the best calibration, under both training losses. It also has a slightly narrower distribution across corruption types (Fig. 5.5 whisker spread). This becomes more pronounced as dataset shift increases and might be due to the MLP being less suited to image data and thus exhibiting less overconfidence than the alternatives. This is further corroborated by the Exp. gating – most suited for the dataset – consistently resulting in the worst calibration. However, the MLP performs worse on MNIST, causing some accuracy loss on shifted data and often poorer calibration. This suggests no conclusive recommendation of a gating architecture can be made and the choice should depend on the target dataset. Additionally, the Conv. gating network does not consistently perform worse than alternatives, indicating that simple, low parameter count gating networks can be used with comparable results.

5.4.1 Challenges for Complex Datasets

As noted earlier, we observe consistent poor MoE accuracy on the CIFAR10 dataset. While for heavily shifted data the results are generally comparable to or slightly better than that of a single network (Fig. 5.5, shift intensities 4 and 5), this is not true for ID or slightly shifted data. The behaviour persists for different loss and gating network combinations. Additionally, an extensive hyperparameter search using early stopping has been conducted to rule out potential overfitting and other common problems. The results suggest that as long as a level of implicit specialisation among experts is maintained, the overall accuracy level remains lowered (with better results achieved when the gating network favours a single expert from the start, or the learned gating is uniform, emulating a single predictor and a DE respectively).

Different training strategies, such as alternating training experts and the gating network or re-training the gating network post-hoc were also trialled, however, no consistent improvement was observed. This indicates that training full MoE models using modern DNNs as experts is a challenging task and more advanced techniques might be needed to make the method beneficial in practice.

5.5 Bayesian Gating Approaches

While the results in Section 5.4 indicate MoEs are not well-calibrated out-of-the-box, they motivate further exploration. In particular, we confirm the implicit expert localisation via the end-to-end joint expert and gating network training is effective and produces highly diverse networks. The expert sets maintain high oracle accuracy – similar to or exceeding that of DEs. These qualities suggest that it is possible to construct a gating method leading to well-calibrated but correct predictions.

We thus hypothesise the calibration of a MoE model depends primarily on the gating network and if we can improve its calibration, the entire model would benefit. As Bayesian methods are often used to improve uncertainty estimates (Ovadia et al., 2019), we examine their use in the gating network. In particular, we study the use of MC Dropout and Laplace approximation (LA) as alternatives to classic DNNs.

For MC Dropout, as briefly introduced in Section 2.1, we add dropout layers after every non-final layer in a given gating network. At test time, the gating weights are computed as a mean of 50 forward pass outputs, consistent with the use of MC Dropout throughout this work.

LA uses a second-order Taylor expansion of the loss $\mathscr{L}(\mathscr{D};\boldsymbol{\theta})$ as a function of network parameters around a given estimate to approximate $p(\boldsymbol{\theta}|\mathscr{D})$ by a Gaussian distribution. In particular, it can be applied *post-hoc*, for a maximum-a-posteriori estimate θ_{MAP} found by standard DNN training minimizing the negative log-likelihood of the data (cross-entropy in classification). The approximation is computed as

$$p(\boldsymbol{\theta} \mid \mathscr{D}) \approx \mathscr{N}\left(\boldsymbol{\theta}; \boldsymbol{\theta}_{\text{MAP}}, \Sigma\right) \text{ where } \Sigma := -\left(\nabla^2_{\theta}\mathscr{L}(\mathscr{D};\boldsymbol{\theta})\big|_{\theta_{\text{MAP}}}\right)^{-1}, \tag{5.5}$$

and we refer to Daxberger et al. (2021) for a detailed derivation.

The Hessian, required for the covariance, is computed as follows:

$$\nabla^2_{\theta}\mathscr{L}(\mathscr{D};\boldsymbol{\theta})\big|_{\theta_{\text{MAP}}} = -\gamma^{-2}I - \sum_{n=1}^{N} \nabla^2_{\theta} \log p\left(y^{(n)} \mid \boldsymbol{x}^{(n)}, \boldsymbol{\theta}\right)\Bigg|_{\theta_{\text{MAP}}}, \tag{5.6}$$

assuming a zero-mean Gaussian prior over the weights with homoscedastic variance γ^2.

We use the library provided by Daxberger et al. (2021) to produce such post-hoc approximations for the gating network. In practice, the Hessian matrix is approximated as a generalized Gauss-Newton matrix (Schraudolph, 2002) and the predictive distribution uses the probit approximation (MacKay, 1992).

To examine and compare the impact of LA using several gating network architectures within reasonable computational constraints, we restrict the approximated posterior over weights to include only the parameters of the DNN's last layer, not the entire network. This allows for the full approximate Hessian matrix for the parameters of interest to be computed, as opposed to imposing additional assumptions on it (e.g. diagonal form).

5.5.1 Calibration Results

To analyse the impact of dropout rate for gating with MC Dropout and prior variance for gating with post-hoc LA we restrict the results reported to a single loss function and gating architecture combination – L_{ENS} and Conv. gating. This is done primarily for brevity – experiments were conducted for all settings, but the insights related to Bayesian gating strategies were found to be consistent.

Gating Using MC Dropout

The use of MC Dropout in gating networks results in a slight increase in MoE model accuracy, both for ID and shifted data. The effect is most pronounced for slight rotation on MNIST (Fig 5.6) and all variations of CIFAR10 (Fig. 5.7). However, for the latter, some training challenges remain and the ID accuracy achieved remains lower than that of DEs.

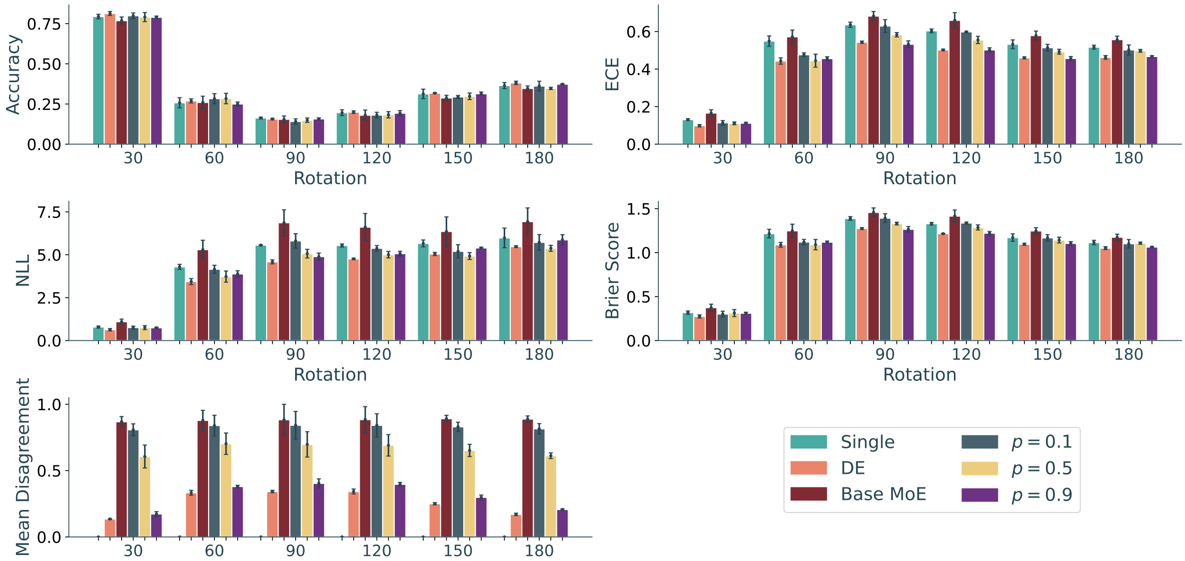


Fig. 5.6 Comparison, for rotated MNIST data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models trained with Conv. gating and MC Dropout with varied dropout probabilities p. The x-axis labels denote rotation in degrees, bar heights – the metric value. Lines show the 95% confidence interval as obtained from 3 independent runs.

In terms of calibration, the different dropout probability values p create a gradual shift from the poorly calibrated base MoE model using a regular DNN for gating to a calibration level similar to that of DEs when $p = 0.9$ is used. The trend can be seen on both MNIST and CIFAR10, with Fig. 5.6 and Fig. 5.7 showing the respective results. The trend is seen most clearly in the change in ECE for both datasets.

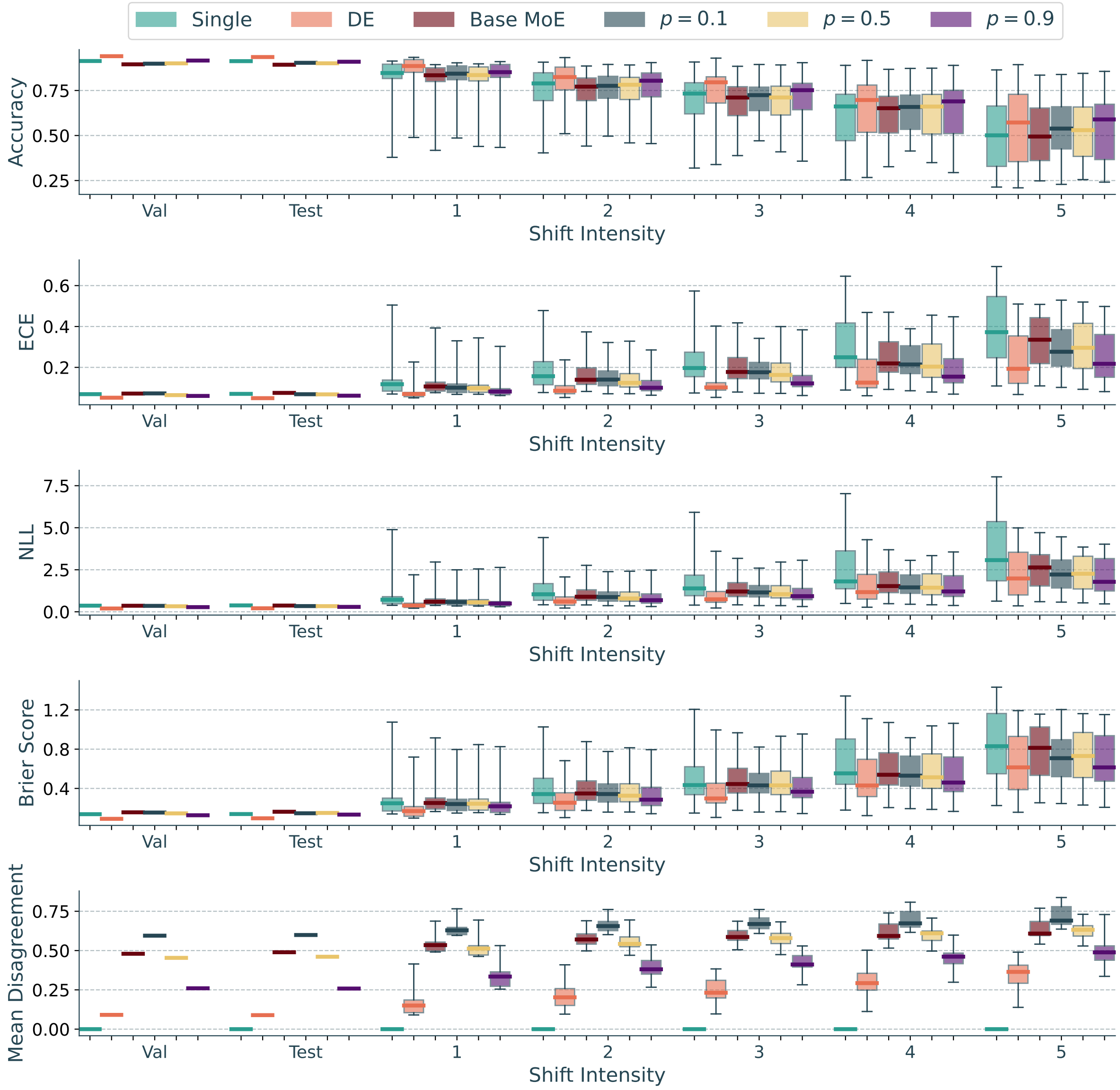


Fig. 5.7 Comparison, for CIFAR10 ID and shifted data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models trained with Conv. gating and MC Dropout with varied dropout probabilities p. Box plots summarise the distributions of metric values across different data shift types. Box boundaries indicate 25th and 75th percentiles, thick coloured lines – the median, and whiskers show the full range of values.

This is in part due to gating networks with high dropout rates inducing softer localisation. Due to the randomness in the training process, more diverse samples impact a particular expert. This is directly reflected in network disagreement. While using a dropout rate of $p = 0.1$ on the CIFAR dataset induces slightly higher expert diversity, using $p = 0.5$ and $p = 0.9$ causes the diversity to drop, although it remains significantly higher than for DEs. This trend is more prevalent on the MNIST dataset – we observe a consistent gradual drop in diversity as the dropout rate is increased, with models using $p = 0.9$ only slightly more diverse than DEs.

While the MoE models using MC Dropout for gating can offer calibration and accuracy comparable to that of DEs for shifted data (e.g. $p = 0.9$, shift intensity 5 in Fig. 5.7 or rotations 90-150 on MNIST), this is not reflected on ID sets. Fig. 5.8 shows the calibration of all MoE methods is slightly worse than that of DEs on the MNIST dataset, with a similar, although less distinct, trend seen for the CIFAR10 ID sets.

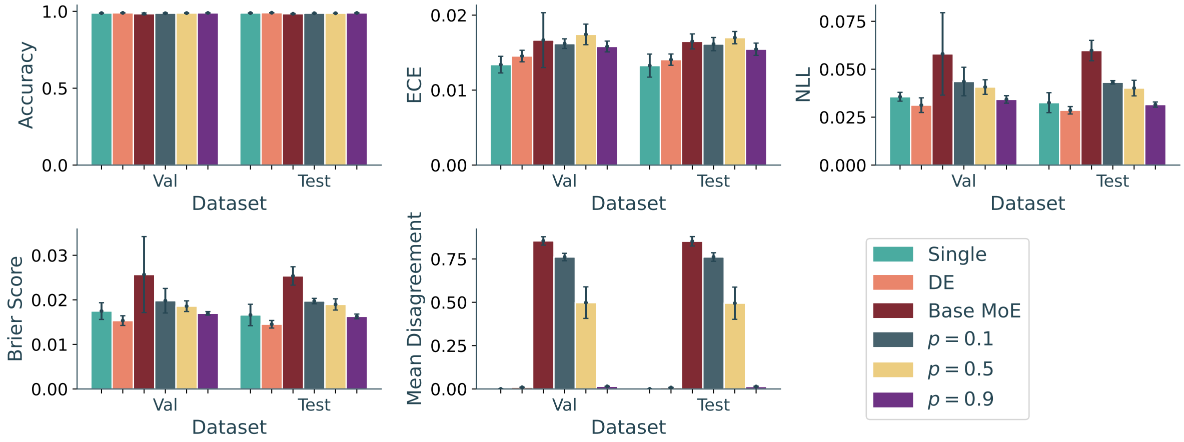


Fig. 5.8 Comparison, for ID MNIST test data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models trained with Conv. gating and MC Dropout with varied dropout probabilities p. Bar height indicates the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

Post-hoc Laplace Approximation for Gating

Employing LA with progressively higher prior variance assumed has a similar effect to using increasingly high dropout rates. The calibration level of the original MoE model is improved, reaching a level similar to that of DEs without any accuracy reductions. This is illustrated by the methods with γ^2 between 0.1 and 2 on shifted MNIST data (Fig 5.9).

An equivalent effect is observed for MoEs with LA γ^2 between 0.1 and 10 on CIFAR10 (Fig. 5.11). However, the method is fundamentally different. Applying LA does not change the experts or the way they are trained. Thus the disagreement (and diversity) level in the ensemble remains constant. The only change comes from the estimated distribution over the gating network's weights. As shown in Fig. 5.9 and Fig. 5.11, the method reduces the predictor's overconfidence, while maintaining accuracy. Notably, in this section the last metric shown in visualisations is ensemble prediction confidence, illustrating this statement.

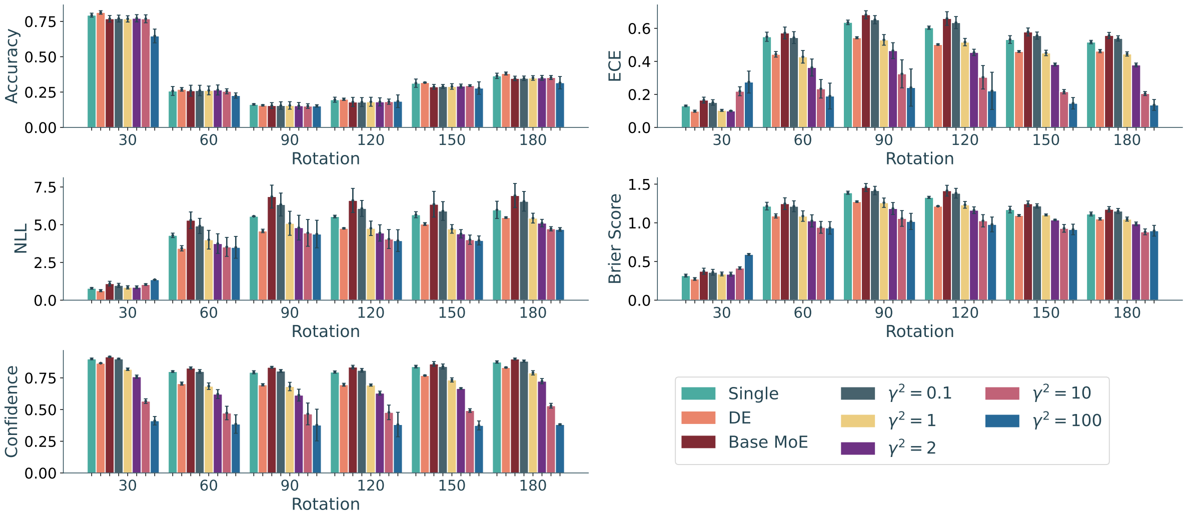


Fig. 5.9 Comparison, for rotated MNIST data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models using post-hoc Laplace approximation for the Conv. gating network with varied prior variances. The x-axis labels denote rotation in degrees, bar heights – the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

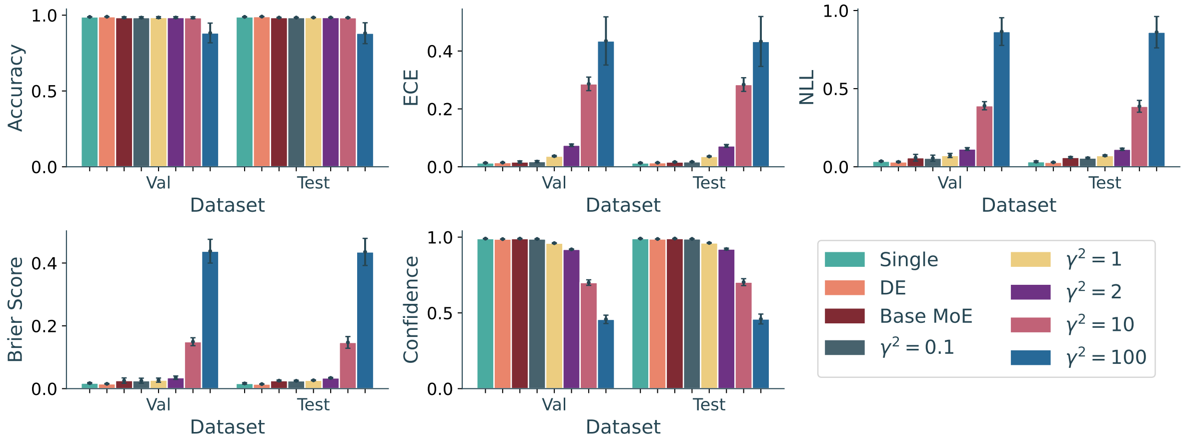


Fig. 5.10 Comparison, for ID MNIST data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models using post-hoc Laplace approximation for the Conv. gating network with varied prior variances. Bar height indicates the metric value. Dark lines show the 95% confidence interval as obtained from 3 independent runs.

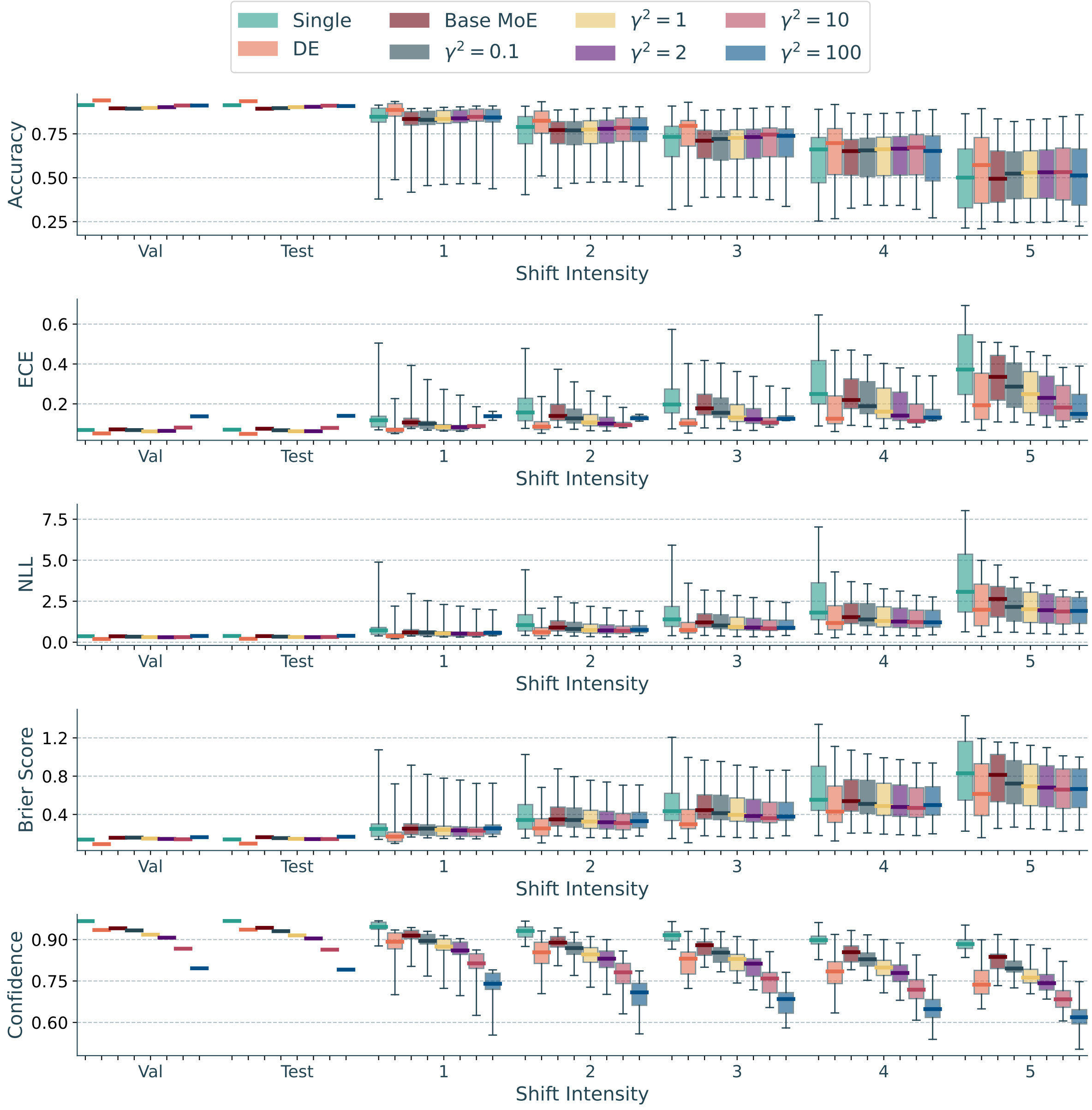


Fig. 5.11 Comparison, for CIFAR10 ID and shifted data, of baseline methods, a simple MoE using L_{ENS} and Conv. gating (Base MoE) and models using post-hoc Laplace approximation for the Conv. gating network with varied prior variances. Box plots summarise the distributions of metric values across different data shift types. Box boundaries indicate 25th and 75th percentiles, thick coloured lines – the median, and whiskers show the full range of values.

It is now possible to cause an effect similar to over-regularisation discussed in Chapter 4 – assuming very high prior variance artificially lowers the overall confidence too much. This is seen particularly clearly for ID data – on MNIST, as shown in Fig. 5.10, calibration metrics for methods with $\gamma^2 = 10$ or $\gamma^2 = 100$ indicate much poorer performance than for other methods.

This effect is less significant on the CIFAR10 dataset, although we do observe a significant worsening in the ID ECE value when $\gamma^2 = 100$ as seen in Fig. 5.11. The visualisation here also gives us an indication of performance variation across different dataset shift types. We note that using LA with $\gamma^2 \in [0.1, 1, 2, 10]$ not only improves the calibration metric median when compared to the original MoE but also significantly reduces the spread of values. This indicates utilising Bayesian gating networks allows the MoE to become more robust to different dataset shifts, a quality highly desirable for practical applications.

5.6 Discussion

The experimentation covered in this chapter has provided two main conclusions. First, MoE models with all components implemented as standard DNNs are inherently poorly calibrated out-of-the-box. Although end-to-end training of the experts can induce a high level of expert localisation and leads to very high diversity between the networks, both they and the gating network suffer from calibration issues. They combine to produce predictions that are sometimes even more overconfident than those of a single predictor, particularly on shifted data. This persists across a variety of loss function and gating network architecture combinations.

The second important conclusion is that MoE calibration can be improved to nearly match DEs by using Bayesian approaches to adjsut the gating weights produced. In particular, using MC Dropout with a high dropout rate trades off localisation with less extreme gating choices, leading to models which resemble DEs more closely in terms of the subnetwork accuracy, but maintain a higher level of diversity. Meanwhile using the LA allows us to control the confidence level of the gating network while making use of MoE components trained using standard methods. This allows for improved calibration without accuracy loss, as well as showing indications of increased robustness to the type of dataset shift. However, it is possible to choose a distribution that has a spread too wide for a given problem and cause underperformance for ID data. The prior variance hyperparameter in LA thus requires more careful tuning than the dropout rate, although it can provide slightly better results.

Despite being able to improve over the calibration of standard MoE models, these methods are not able to consistently take advantage of the additional diversity in the ensemble and provide better calibration than DEs under dataset shift. Additionally, we find MoEs are difficult to train to a high standard for complex datasets. In particular, we are unable to ensure MoEs achieve accuracy higher than that of a single predictor on the CIFAR10 dataset without the gating strategy collapsing to the extremes of always choosing the same expert or producing uniform weights, emulating single predictors and DEs respectively. Both of these findings indicate that while MoEs provide a promising source of diversity for improving DE calibration, further research is needed to determine appropriate training and gating strategies to take advantage of it.

Chapter 6

Conclusion

In this dissertation, we have thoroughly studied strategies for improving DEs by diversification, and their impact on calibration. Upon a literature review, we identified two main paradigms. The first, *implicit diversification*, relies on introducing additional randomness or certain behaviours without specifically optimising for them. The second, *explicit diversification*, involves quantifying a desirable behaviour and jointly training for it and predictive performance.

To study the latter, we analysed two explicit diversification strategies: DEs with negative correlation learning (NCL-DE) and using pairwise predictor cross-entropy as regularisation (CE-DE). The methods were proposed in earlier works by Shui et al. (2018) and Opitz et al. (2016) respectively, however, they had not been studied extensively in the context of induced uncertainty estimates. We investigated their effects on prediction calibration for both ID and shifted data - a setting recognised as particularly important for studying uncertainty calibration in recent research (Ovadia et al., 2019).

We found both NCL-DE and CE-DE can improve over baseline DE calibration. However, the improvements for ID test sets were slight and most benefits were observed in calibration under dataset shift. As both methods depend on the choice of a scaling parameter λ, this raised concern over their applicability in practice. We showed that when calibration on an ID validation set is used as a heuristic for hyperparameter selection, the values chosen are suboptimal in the context of dataset shift. This indicates more robust strategies for selecting λ are needed – expanded on in Section 6.1.2. We also tested a hypothesis that diversity regularisation introduced by NCL-DE and CE-DE is more significant in the early stages of training by using annealed regularisation term scaling. However, we were unable to draw definite conclusions as the empirical evidence obtained suggests it is a promising strategy for NCL-DE but less effective for CE-DE.

We further proposed traditional ensembling by mixtures of experts (MoE) with all components implemented as DNNs can be seen as a form of implicit DE diversification, resulting in highly diverse and localised subnetworks. We analysed the out-of-the-box calibration of such models, under a range of loss function and gating network architecture combinations. We concluded MoE calibration tends to resemble the performance of a single predictor with models sometimes exhibiting even more overconfidence on shifted data. We motivate this by miscalibration of the gating network – the weights often significantly favour a single expert, which in turn is trained on an implicitly chosen data subset and thus poorly calibrated.

We further proposed using Bayesian gating networks to improve the MoE calibration. In particular, we showed that using MC Dropout or a final-layer Laplace approximation can bring the calibration of MoE models to a level similar to that of DEs, especially for shifted data. However, even using these methods we are unable to fully take advantage of the subnetwork diversity and improve over DEs.

6.1 Further Work

Our work provides detailed analysis, both in terms of explicit DE diversification via NCL-DE and CE-DE, and the implicit localisation in MoEs. However, the results are fundamentally empirical and observational. While they provide valuable insights into the applicability of DE diversification methods and illustrate the challenges of utilising implicit training data selection as a source of diversity via MoEs, they offer limited clarity in terms of the underlying principles. Furthermore, the results are often inconclusive and further research is needed to determine the true potential of these strategies in improving uncertainty calibration.

6.1.1 Calibration Under Natural Dataset Shift

Our work serves primarily as a proof of concept, with evaluation benchmarks borrowed from literature, in particular the evaluation framework used by Ovadia et al. (2019). While this utilises a range of datasets and allows us to showcase the differences between ID calibration and the trends under dataset shift, all changes in the test data are artificial. Corruptions applied to CIFAR images mimic possible real-world scenarios, but are still algorithmically generated.

A similar trend persists in most literature studying DNN and DE calibration, with limited information available on the calibration strategy performance under real-world dataset shift. This is caused in part by limited dataset availability. Recently Koh et al. (2021) compiled a variety of datasets representing natural dataset changes, manifesting via domain or subpopulation shift. We believe studying diversified DE calibration under these conditions could provide valuable insights in terms of their applicability in practice.

6.1.2 Robust Hyperparameter Selection

In Chapter 4 we noted that calibration on an ID validation set does not generalise well to calibration under dataset shift. Similar observations were also made by Ovadia et al. (2019) when discussing methods such as temperature scaling, which lead to excellent ID calibration but are outperformed by DEs on shifted data.

The methods examined in this dissertation, in particular, NCL-DE and CE-DE, have been shown to have the potential to improve over DEs. However, their performance is highly sensitive to the values of introduced regularisation scaling terms.

To make the methods reliable in practice, further investigation into hyperparameter selection is required. A potential strategy is choosing by performance on an additional OOD validation set. Similar methods have been successfully applied when training networks for outlier detection (e.g. by Hendrycks et al. (2018), who show that using a sample from a general large-scale dataset works to stand in for OOD data works well for a range of tasks), and thus might be expected to perform well.

6.1.3 Universal Deep Ensemble Diversification Applicability

Our work focused exclusively on image classification. It is desirable to also explore other settings and data modalities, such as text or tabular data, as well as deriving more extensive theoretical motivation for the approaches studied.

While negative correlation learning has been widely applied to regression ensembles to improve predictive performance, it has not been thoroughly studied in terms of uncertainty calibration. Similarly, MoE strategies are rarely evaluated or analysed in this context. To verify if the methods are universally applicable, both further analysis on a wide variety of data types and tasks, and a theoretical study of the underlying principles are desirable.

6.1.4 Improvements in Mixture of Experts Training

We have pointed out a range of problems encountered when training MoE models for the relatively complex CIFAR datasets. Simple end-to-end training strategies struggle to realise the primary premise of ensembling and exceed the predictive performance of a single network.

This calls for a further study of MoE training strategies. We have primarily focused on implicit localisation, however, explicit localisation of experts coupled with post-hoc gating training might improve the performance. While we have explored adjusting the gating network after training the experts, due to time constraints no extensive tuning was performed, and only a small variety of strategies were tested.

6.1.5 Efficient Mixtures of Experts

We have chosen to focus on the classic setting for MoE, where the ensemble prediction is a weighted average of individual predictor outputs and weights are provided by the gating network. However, recent work, like that by Shazeer et al. (2017) and Riquelme et al. (2021), uses gating output to select a subset of experts to be queried. We expect the calibration of MoE models constructed this way to be roughly equivalent to that of a classic MoE, and observe such trends when examining small sparsely gated MoEs (see Appendix C for results). However, sparsity allows the number of experts to be increased without changing the computational cost. While we did not examine this in favour of fixed ensemble size, it is not clear if and how such expansion would affect calibration, and further study is required.

References

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Sebastian Buschjäger, Lukas Pfahler, and Katharina Morik. Generalized negative correlation learning for deep ensembling. *arXiv preprint arXiv:2011.02952*, 2020.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18, 2004.

Erik A. Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, M. Bauer, and Philipp Hennig. Laplace redux - effortless bayesian deep learning. *ArXiv*, abs/2106.14806, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45014-6.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3723–3731, 2019.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319, 2020.

L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. doi: 10.1109/34.58871.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Siddhartha Jain, Ge Liu, Jonas Mueller, and David Gifford. Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4264–4271, 2020.

Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pages 2119–2127, 2016.

Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural networks*, 12(10): 1399–1404, 1999.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74, 2004.

Aryan Mobiny, Pengyu Yuan, Supratik K Moulik, Naveen Garg, Carol C Wu, and Hien Van Nguyen. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):1–14, 2021.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

NHTSA. Tesla crash preliminary evaluation report. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2017.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

Jeremy Nixon, Balaji Lakshminarayanan, and Dustin Tran. Why are bootstrapped deep ensembles not better? In *"I Can't Believe It's Not Better!"NeurIPS 2020 workshop*, 2020.

Michael Opitz, Horst Possegger, and Horst Bischof. Efficient model averaging for deep neural networks. In *Asian Conference on Computer Vision*, pages 205–220. Springer, 2016.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

Rahul Rahaman and Alexandre H Thiery. Uncertainty quantification and deep ensembles. *arXiv preprint arXiv:2007.08792*, 2020.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *arXiv preprint arXiv:2106.05974*, 2021.

Claude Sammut and Geoffrey I. Webb, editors. *Bias-Variance-Covariance Decomposition*, pages 111–111. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_77. URL https://doi.org/10.1007/978-0-387-30164-8_77.

Nicol Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14:1723–38, 08 2002. doi: 10.1162/08997660260028683.

Alireza Shafaei, Mark W. Schmidt, and J. Little. A less biased evaluation of out-of-distribution sample detectors. In *BMVC*, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Changjian Shui, Azadeh Sadat Mozafari, Jonathan Marek, Ihsen Hedhli, and Christian Gagné. Diversity regularization in deep ensembles. *arXiv preprint arXiv:1802.07881*, 2018.

Asa Cooper Stickland and Iain Murray. Diverse ensembles improve calibration. *arXiv preprint arXiv:2007.04206*, 2020.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *arXiv preprint arXiv:2006.13570*, 2020.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for uncertainty estimation and dataset shift. *arXiv preprint arXiv:2006.08573*, 2020.

Appendix A

Trends for Different Base Predictor Architectures on MNIST

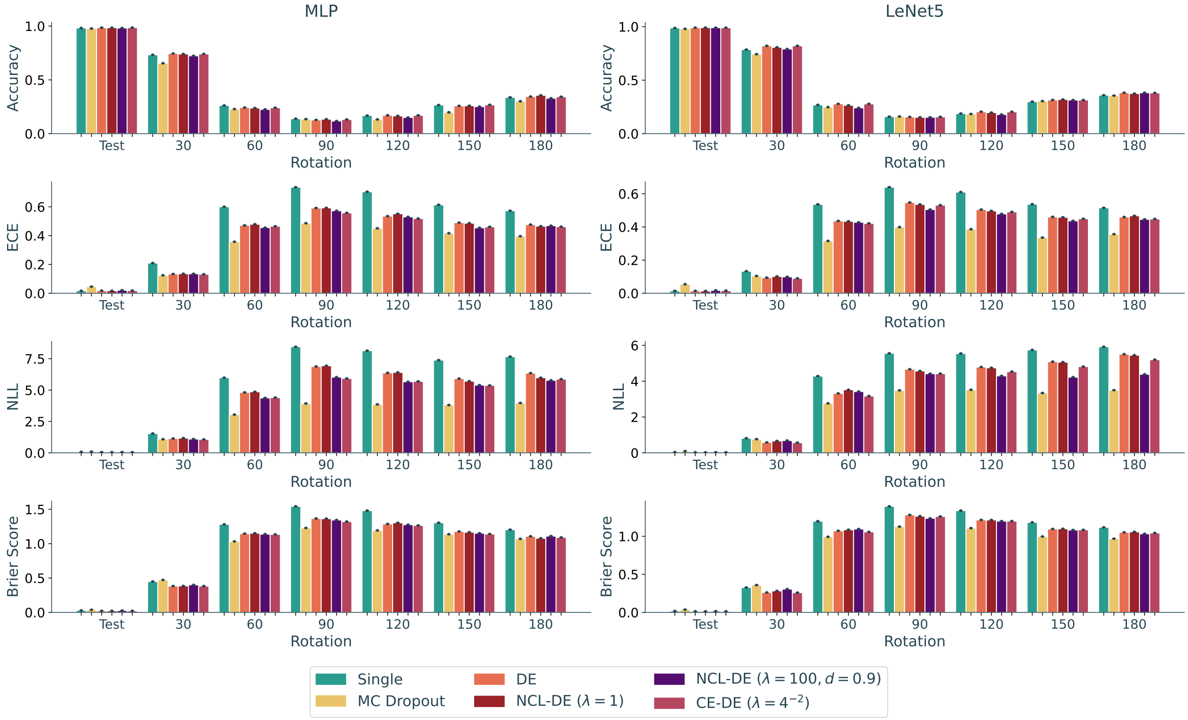


Fig. A.1 Comparison of using MLP and LeNet5 base predictors for MNIST data, covering ID and rotated test data. The left-hand-side column corresponds to ensembles with MLP base predictors, right-had-side one – LeNet5. Bar height indicates metric value, grouping along the x-axis – the dataset evaluated on. Trends observed are consistent across architectures.

Appendix B

Numeric Results for Regularisation Experiments

This appendix contains numerical results supporting plots and comments in Chapter 4. The tables cover all datasets used: MNIST, CIFAR10 and CIFAR100.

Arrows next to metric names indicate whether high or low values are desirable. For ID data, no indication is given for mean disagreement – the preference is ambiguous. Numbers in parenthesis indicate standard deviation across 3 experiments. Values in **bold** are best across different methods or fall within a standard deviation of the best one.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement
DE	0.935 (0.0016)	0.05 (0.0014)	**0.21** (0.0045)	**0.097** (0.0019)	0.091 (0.0012)
NCL-DE (λ=0.01)	**0.937** (0.0016)	**0.048** (0.001)	**0.208** (0.0024)	**0.096** (0.0013)	0.091 (0.0005)
NCL-DE (λ=100, d=0.96)	0.92 (0.0014)	0.064 (0.0019)	0.249 (0.0092)	0.12 (0.0039)	0.126 (0.0036)
CE-DE ($\lambda = 4^{-4}$)	**0.936** (0.0008)	0.05 (0.0005)	**0.21** (0.0049)	**0.097** (0.0011)	0.091 (0.0019)
CE-DE (λ=2, d=0.9)	0.932 (0.0013)	0.052 (0.0006)	0.215 (0.0029)	0.101 (0.0017)	0.097 (0.0046)

Table B.1 Results for DEs and regularised DE variations, selected by their ECE on the validation set, for CIFAR10 test data. Values aggregated across all corruption types.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement
Single	0.989 (0.0007)	**0.013** (0.0007)	0.033 (0.0026)	0.017 (0.0012)	-
MC Dropout	0.981 (0.0008)	0.052 (0.0029)	0.09 (0.0055)	0.035 (0.0019)	0.048 (0.0033)
DE	**0.991** (0.0002)	0.014 (0.0003)	**0.029** (0.001)	0.015 (0.0004)	0.008 (0.0004)
NCL-DE (λ=3)	0.99 (0.0002)	0.023 (0.0055)	0.038 (0.0058)	0.017 (0.0013)	0.037 (0.0186)
NCL-DE (λ=100, d=0.9)	0.99 (0.0006)	0.017 (0.0001)	0.033 (0.0004)	0.016 (0.0003)	0.011 (0.0005)
CE-DE (λ=0.125)	0.99 (0.0001)	0.02 (0.0016)	0.035 (0.0014)	0.016 (0.0002)	0.009 (0.0002)
CE-DE (λ=2, d=0.92)	**0.991** (0.0001)	0.019 (0.0004)	0.033 (0.0006)	**0.014** (0.0003)	0.01 (0.0002)

Table B.2 Results for baselines and regularised DE variations selected by their performance on shifted data, on the MNIST test dataset.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement ↑
Single	**0.212** (0.0265)	0.466 (0.0267)	5.675 (0.1974)	1.168 (0.0231)	-
MC Dropout	**0.193** (0.0244)	**0.241** (0.0404)	**3.468** (0.1919)	**0.992** (0.033)	0.626 (0.0234)
DE	**0.197** (0.0204)	0.364 (0.0264)	4.835 (0.2615)	1.074 (0.0287)	0.551 (0.0267)
NCL-DE (λ=3)	0.191 (0.0073)	0.333 (0.0227)	4.715 (0.4327)	1.052 (0.0147)	0.645 (0.0179)
NCL-DE (λ=100, d=0.9)	0.188 (0.0207)	0.268 (0.0327)	3.92 (0.1279)	**1.013** (0.0234)	0.705 (0.0179)
CE-DE (λ=0.125)	**0.2** (0.0274)	0.348 (0.0681)	4.324 (0.0117)	1.062 (0.0624)	0.555 (0.0614)
CE-DE (λ=2, d=0.92)	**0.202** (0.0144)	**0.231** (0.0472)	3.892 (0.0553)	**0.995** (0.0308)	**0.738** (0.0414)

Table B.3 Results for baselines and regularised DE variations selected by their performance on shifted data, for 10-pixel translated MNIST test data.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement ↑
Single	0.258 (0.0158)	0.549 (0.0139)	4.294 (0.0739)	1.216 (0.0247)	-
MC Dropout	0.242 (0.0049)	**0.334** (0.0155)	**2.757** (0.0024)	**1.005** (0.0088)	**0.478** (0.0133)
DE	**0.269** (0.0065)	0.443 (0.0086)	3.437 (0.0879)	1.089 (0.0138)	0.333 (0.0092)
NCL-DE (λ=3)	**0.27** (0.0087)	0.417 (0.013)	3.388 (0.0922)	1.065 (0.0167)	0.391 (0.0154)
NCL-DE (λ=100, d=0.9)	0.251 (0.0097)	0.42 (0.0082)	3.278 (0.1431)	1.078 (0.0161)	0.371 (0.002)
CE-DE (λ=0.125)	0.263 (0.0054)	0.426 (0.002)	3.156 (0.0562)	1.075 (0.0047)	0.324 (0.0129)
CE-DE (λ=2, d=0.92)	0.266 (0.0012)	0.388 (0.0106)	2.962 (0.0116)	1.035 (0.0132)	0.384 (0.0201)

Table B.4 Results for baselines and regularised DE variations selected by their performance on shifted data, for 60° rotated MNIST test data.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement ↑
Single	0.622 (0.0051)	0.29 (0.0081)	2.429 (0.1105)	0.632 (0.0131)	0.0 (0.0)
MC Dropout	0.615 (0.0089)	0.197 (0.0103)	1.435 (0.0873)	0.556 (0.0174)	0.233 (0.0029)
DE	0.664 (0.0072)	0.163 (0.0062)	1.455 (0.0553)	0.482 (0.0104)	0.292 (0.0034)
NCL-DE (λ=3)	0.666 (0.0026)	**0.151** (0.002)	1.396 (0.023)	**0.474** (0.0033)	0.327 (0.0019)
NCL-DE (λ=100, d=0.96)	0.654 (0.0049)	**0.152** (0.0065)	**1.262** (0.0302)	0.483 (0.0116)	**0.336** (0.0058)
CE-DE (λ=0.1)	0.666 (0.0015)	**0.153** (0.0067)	**1.275** (0.0261)	**0.472** (0.0059)	0.307 (0.009)
CE-DE (λ=2, d=0.94)	**0.672** (0.0045)	0.155 (0.0055)	1.343 (0.0411)	**0.467** (0.009)	0.302 (0.0061)

Table B.5 Results for baselines and regularised DE variations selected by their performance on shifted data, for CIFAR10 test data shifted with intensity 4. Values reported as aggregated across all shift types.

Method	Accuracy ↑	ECE ↓	NLL ↓	Brier Score ↓	Mean Disagreement ↑
Single	0.349 (0.0052)	0.371 (0.0062)	4.609 (0.0956)	0.961 (0.0102)	0.0 (0.0)
MC Dropout	0.344 (0.0011)	0.2 (0.0049)	3.425 (0.0526)	0.829 (0.0047)	0.458 (0.0025)
DE	0.406 (0.0007)	0.18 (0.0015)	3.241 (0.0651)	0.764 (0.0008)	0.576 (0.0008)
NCL-DE (λ=1)	0.406 (0.0014)	0.179 (0.002)	3.204 (0.0163)	0.764 (0.0015)	0.59 (0.0018)
NCL-DE (λ=100, d=0.92)	**0.412** (0.0039)	**0.158** (0.0057)	**2.842** (0.0684)	**0.742** (0.0064)	0.585 (0.0053)
CE-DE (λ=0.1)	0.405 (0.0033)	**0.161** (0.0017)	2.957 (0.0804)	0.751 (0.0031)	**0.61** (0.0029)
CE-DE (λ=2, d=0.9)	0.405 (0.002)	0.178 (0.0021)	3.194 (0.0285)	0.763 (0.0025)	0.581 (0.0043)

Table B.6 Results for baselines and regularised DE variations selected by their performance on shifted data, for CIFAR100 test data shifted with intensity 4. Values reported as aggregated across all shift types.

Appendix C

Sparse Mixtures of Experts

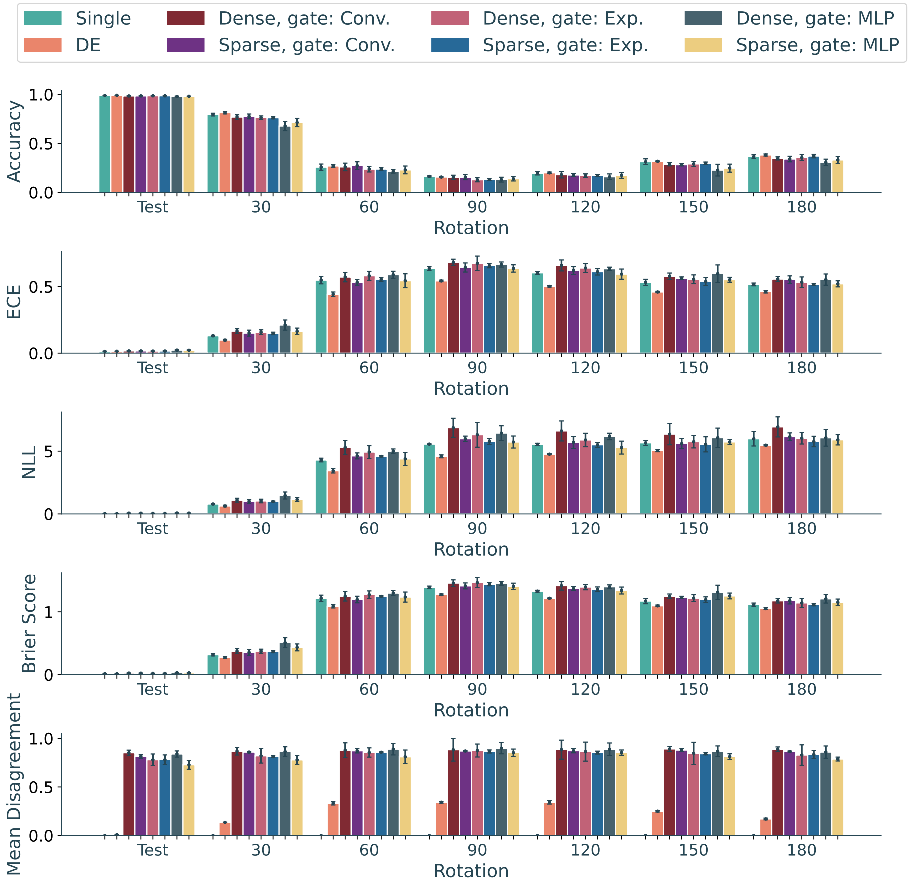


Fig. C.1 Comparison of trends for traditional (dense) and sparse MoE predictors using different gating networks and trained via ensemble loss L_{ENS} on MNIST data (ID and rotated test data). The overall trend of MoE calibration being similar or worse to that of a single network persists for sparse models. The relative ordering of gating network architecture performance is also mostly consistent for sparse and dense models.