Uncertainty Modelling in Deep Learning and Decision Trees for Medical Image Processing

Ryutaro Tanno

Department of Computer Science
University College London
Gower Street, London, United Kingdom

THESIS

Submitted for the degree of
Doctor of Philosophy, University College London

July 24, 2019

I, Ryutaro Tanno, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Impact Statement

Acknowledgements

Aknowledgement.

MPhil-to-PhD Transfer

Problem statement

Deep learning and decisions trees are now ubiquitous in the field of medical image processing. However, the current methods disproportionately rely on deterministic algorithms, which lack a mechanism to represent and manipulate uncertainty about models and predictions. In safety-critical applications such as medical imaging, quantifying what the model does not know is critical for constructing a reliable decision making system. The aim of this thesis is to explore probabilitic modelling as a framework to integrate uncertainty information in deep learning and decision tree models, and demonstrate utility in various medical image processing applications.

Literature Review

I plan to survey the following topics in the given order to motivate the thesis.

1. **"Classics" on uncertainty quantification for medical imaging**: there is a large body of prior work on uncertainty quantification based on traditional probabilistic modelling techniques (e.g. grpahical models) in a variety of medical image analysis applications such as registration, classification, segmentation and image synthesis. I would like to use this section to motivate the importance of uncertainty quantification in medical imaging applications.

2. **Surge of black-box models in medical imaging**: In the last few years, with increasing availability of labelled data, hardware and user-friendly software, black-box models such as deep learning and decision trees have permeated every corner of medical image processing research, often surpassing the performance of more traditional probabilistic techniques.

3. **Uncertainty in black-box models**: This section will review both theoretical and application-driven previous research on uncertainty modelling in deep learning and decision trees. We will discuss why such research is important for designing safer and interpretable systems for medical applications.

Summary of the contribution

In this thesis I contributed to the three key aspects mentioned above.

Scope of the thesis

Research/thesis progress

1. Relevant publications have been formated into different chapters, but still require editing to improve the flow.

2. Need to write an introduction chapter to set the context with a literature review, explain the importance of uncertainty modelling in medical imaging, and outline my thesis and contributions.

3. Need to write a chapter on future research.

4. (Optional): If time permits, I plan to add another subsection in Chapter 4 by extending the presented method for modelling human uncertainty to the segmentation task.

Timeline

I plan to submit the thesis by mid September (My post-doc fellow position is due to start in the same month). The first rought draft should be completed before 16 August 2019.

List of publications

1. **R. Tanno**, A. Ghosh, F. Grussu, E. Kaden, A. Criminisi, and D. C. Alexander, "Bayesian image quality transfer". (2016) **MICCAI**

2. **R. Tanno**, D. E. Worrall, A. Ghosh, E. Kaden, S. N. Sotiropoulos, A. Criminisi, and D. C. Alexander, "Bayesian image quality transfer with cnns: Exploring uncertainty in dmri super-resolution". (2017) **MICCAI**

3. D. C. Alexander, D. Zikic, A. Ghosh, **R. Tanno**, V. Wottschel, J. Zhang, E. Kaden, T. B. Dyrby, S. N. Sotiropoulos et al., "Image quality transfer and applications in diffusion MRI". (2017) **Neuroimage**

4. **R. Tanno**, A. Makropoulos, S. Arslan, O. Oktay, S. Mischkewitz, F. Al-Noor1, J. Oppenheimer, R. Mandegaran, B. Kainz, M. Heinrich. "AutoDVT: Joint Real-time Classification for Vein Compressibility Analysis in Deep Vein Thrombosis Ultrasound Diagnostics". (2018) **MICCAI**

5. F.J.S. Bragman, **R. Tanno**, Z. Eaton-Rosen, W. Li, D. J. Hawkes, S. Ourselin, D. C. Alexander, J. R. McClelland, M. J. Cardoso, "Uncertainty in multitask learning: joint representations for probabilistic MR-only radiotherapy planning". (2018) **MICCAI**

6. S. B. Blumberg, **R. Tanno**, I. Kokkinos, D. C Alexander. "Deeper Image Quality Transfer: Training Low-Memory Neural Networks for 3D Images". (2018) **MICCAI**

7. K. Kamnitsas, D. Castro, L. Folgoc, **R. Tanno**, D. Rueckert, B. Glocker, A. Criminisi, A. Nori. "Semi-Supervised Learning via Compact Latent Space Clustering". (2018) **ICML**

8. **R. Tanno**, A. Saheedi, S. Sankaranarayanan, D. C. Alexander, N. Silberman, "Learning From Noisy Labels By Regularized Estimation Of Annotator Confusion". (2019) **CVPR**

9. **R. Tanno**, K. Arulkumaran, D. C. Alexander, A. Criminisi and A. Nori, "Adaptive Neural Trees". (2019) **ICML**

10. F.J.S. Bragman*, **R. Tanno***, S. Ourselin, D. C. Alexander, M. J. Cardoso, "Stochastic Filter Groups for Multi-Task CNNs: Learning Specialist and Generalist Convolution Kernels". (2019) **ICCV** (* equal contributions)

11. **R. Tanno**, D. E. Worrall, E. Kaden, A. Ghosh, F. Grussu, A. Bizzi, S. N. Sotiropoulos, A. Criminisi, and D. C. Alexander, "Uncertainty Quantification in Deep Learning for Safer Neuroimage Enhancement". (2019) **Neuroimage** (Under Submission)

12. F.J.S. Bragman*, **R. Tanno***, S. Ourselin, D. C. Alexander, M. J. Cardoso, "Learning task-specific and shared representations in medical imaging". (2019) **MICCAI** (*equal contributions)

13. C. Sudre, B.G. Anson, S. Ingala, D. Jimenez, C. Lane, L. Haider, T. Varsavsky, **R. Tanno**, L. Smith, S. Ourselin, R. Jager, M. J. Cardoso, "Let's agree to disagree: learning highly debatable multirater labelling". (2019) **MICCAI**

14. S. B. Blumberg, M. Palombo, C. S. Khoo, C. Tax, **R. Tanno**, D. C Alexander. "Multi-Stage Prediction Networks for Data Harmonization". (2019) **MICCAI**

Contents

Chapter 1

Introduction

Chapter 2

Quantifying Predictive Uncertainty in Decision Trees

This chapter is largely based on our paper accepted in MICCAI 2016 [1].

A key limitation of the current IQT implementation is the lack of a mechanism to communicate confidence in the predicted target image. High quality training data typically come from healthy volunteers. Thus, performance in the presence of pathology or other effects not observed in the training data is questionable. We expect methods to have high confidence in regions where the method has seen lots of similar examples during training, and lower confidence on previously unseen structures. However, current methods implicitly have equal confidence in all areas. Such an uncertainty characterisation is particularly important in medical applications where ultimately images can inform life-and-death decisions. It is also highly beneficial to downstream image processing algorithms, such as registration or tractography, which can propagate the uncertainty into the output.

In this chapter, we introduce an extension of IQT framework which can simultaneously perform reconstruction and uncertainty quantification over its prediction. We propose an efficient way to incorporate Bayesian inference into the framework and name the new method Bayesian IQT (BIQT). To our knowledge, none of the existing super-resolution [2, 3, 4, 5] and image synthesis methods [6, 7, 8, 9] address the problem of uncertainty estimation. Although many can be cast as *maximum a posteriori* (MAP) optimisation problems, the prohibitive dimensionality or complexity of the posterior distribution (due to non-standard regularisation prior) make the computation of uncertainty intractable or expensive. In contrast, the random forest implementation of the original IQT is amenable to uncertainty estimation thanks to the simple linear model at each leaf node, but the current approach computes *maximum likelihood* (ML) solution. BIQT replaces this ML based inference with Bayesian inference (rather than just MAP) and this allows the uncertainty estimate to reflect unfamiliarity of input data (see Fig. **??**).

We demonstrate BIQT through super-resolution of DTI on HCP dataset [10], which has sufficient size and resolution to provide training data and a testbed to gauge the baseline performance. We then use clinical data sets from multiple sclerosis (MS) and tumour studies to show the efficacy of the uncertainty estimation in the presence of pathology, not represented in the HCP training data.

2.1 Introduction

2.2 Method

Here we first review the original IQT framework based on a regression forest. We then introduce our Bayesian extension, BIQT, highlighting the proposed efficient hyperparameter optimisation method and the robust uncertainty measure.

2.2.1 Background: Image Quality Transfer

IQT splits a LR image into small patches and performs quality enhancement on them independently. This patch-wise reconstruction is formulated as a regression problem of learning a mapping from each patch $\mathbf{x}$ of N_l voxels in the LR image to a corresponding patch $\mathbf{y}(\mathbf{x})$ of N_h voxels in the HR image. Input and output voxels are vector-valued containing p_l and p_h values, and thus the mapping is $\mathbf{x} \in \mathbf{R}^{N_l p_l} \rightarrow \mathbf{y}(\mathbf{x}) \in \mathbf{R}^{N_h p_h}$. Training data comes from high quality data sets, which are artificially downsampled to provide matched pairs of LR and HR patches. For application, each patch of a LR image is passed through the learned mapping to obtain a HR patch and those patches combine to estimate a HR image.

2.2.2 Background: Decisions Trees and Random Forest

To solve the above regression problem, IQT employs a variant of random forests [11]. The method proceeds in two stages: training and prediction. During training, we grow a number of trees on different sets of training data. Each tree implements a piecewise linear regression; it partitions the input space $\mathbf{R}^{N_l p_l}$ and performs regressions in respective subsets. Learning the structure of a tree on dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_i^{|\mathcal{D}|}$ aims to find an 'optimal' sequence of the following form of binary partitioning. At the initial node (root), $\mathcal{D}$ is split into two sets $\mathcal{D}_\mathrm{R}$ and $\mathcal{D}_\mathrm{L}$ by thresholding one of J scalar functions of $\mathbf{x}$, or *features*, $f_1, ..., f_J$. The optimal pair of a feature f_m and a threshold τ with the most effective splitting is selected by maximising the *information gain* [12], $\mathrm{IG}(f_m, \tau, \mathcal{D}) \triangleq |\mathcal{D}| \cdot H(\mathcal{D}) - |\mathcal{D}_\mathrm{R}| \cdot H(\mathcal{D}_\mathrm{R}) - |\mathcal{D}_\mathrm{L}| \cdot H(\mathcal{D}_\mathrm{L})$ where $|\mathcal{D}|$ denotes the size of set $\mathcal{D}$ and $H(\mathcal{D})$ is the average differential entropy of the predictive distribution $\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H})$ given by

$$H(\mathcal{D}) \triangleq -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \int \mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H}) \cdot \log \mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H}) \, \mathbf{dy}. \tag{2.1}$$

Maximising the information gain helps selecting the splitting with highest confidence in predictive distributions. This optimization problem is solved by performing golden search on the threshold for all features. The hypothesis space $\mathcal{H}$ specifies the class of statistical models and governs the form of predictive distribution. In particular, IQT fits the ML estimation of a linear model with a Gaussian noise. To control over-fitting, a validation set $\mathcal{D}^\mathrm{V}$ with similar size to $\mathcal{D}$ is used and the root node is only split if the residual error is reduced. This process is repeated in all new nodes until no more splits pass the validation test.

At the time of prediction, every LR patch $\mathbf{x}$ is routed to one of the leaf nodes (nodes with no children) in each tree through a series of binary splitting learned during training, and the corresponding HR patch is estimated by the mode of the predictive distribution. The forest output is computed as the average of predictions from all trees weighted by the inverted variance of predictive distributions.

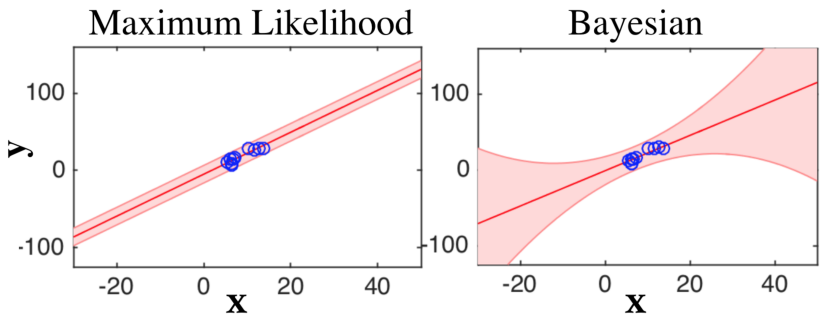


Figure 2.1: 1D illustration (i.e. both $\mathbf{x}, \mathbf{y} \in \mathbf{R}$) of maximum likelihood and Bayesian linear models fitted to the data (blue circles). The red line and shaded areas show the mode and variance (uncertainty) of $\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H})$ at respective $\mathbf{x}$ values. Bayesian method assigns high uncertainty to an input distant from the training data whilst the ML's uncertainty is fixed.

2.2.3 BIQT: Locally Bayesian Decision Trees

Our method, BIQT follows the IQT framework described in chapter 2 and performs a patch-wise reconstruction using a regression forest. The key difference lies in the choice of $\mathcal{H}$ (eq. (2.1)); we fit a Bayesian linear model [13] instead of standard linear regression. We will again assume that each (clean) output is generated from the linear model $\mathbf{W}\mathbf{x}$ and the observed output is generated by the addition of a Gaussian noise. The previously employed *maximum likelihood* (ML) approach just yields a point-estimate for the model parameters $\mathbf{W}$ and we accept it as the 'true' parameter of the model although limited available training data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{D}$ means that parameter estimation is inherently uncertain. For instance, there might well be several candidates for $\mathbf{W}$ whose likelihood are very close to that of the ML estimator $\mathbf{W}_{ML}$, implying that the data at our disposal are not sufficient to specify the value of $\mathbf{W}$; the model would perform almost equally well even if you chose one of these equally good options for $\mathbf{W}$. Therefore, instead of resorting to a single estimate, it would be more sensible to account for the presence of such uncertainty. To this end, we explore possible extensions of the current method in the Bayesian paradigm. By viewing the linear map $\mathbf{W}$ as a random variable, we can compute the estimated predictive distribution $\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H})$ by integrating over all possible values of $\mathbf{W}$ according to their corresponding strength of belief given the data. One of the limitations of the previous approach is the fact that the covariance of the predictive distribution $\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \mathcal{H})$ is solely determined by the training data $\mathcal{D}$ and fixed for all new test input $\mathbf{x}$. This is not realistic – for example if the test input is very far from the cloud of training data, you should expect higher uncertainty. We shall see that this Bayesian extension yields an adaptive covariance which varies with the new test input $\mathbf{x}$ in an intuitive manner (Figure **??**).

More formally, given an arbitrary subset of training data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ at one of the internal nodes of a tree, Bayesian linear regression fits the model $\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\eta}$ where the additive noise $\boldsymbol{\eta}$ and the linear transform $\mathbf{W} \in \mathbf{R}^{N_h p_h \times N_l p_l}$ follow isotropic Gaussian distributions $\mathrm{P}(\boldsymbol{\eta}|\beta) = \mathcal{N}(\boldsymbol{\eta}|\mathbf{0}, \beta^{-1}\mathbf{I})$ and $\mathrm{P}(\mathbf{W}_{|}|\alpha) = \mathcal{N}(\mathbf{W}_{|}|\mathbf{0}, \alpha^{-1}\mathbf{I})$, with $\mathbf{W}_{|}$ denoting the row-wise vectorised version of $\mathbf{W}$. The hyperparameters α and β are positive scalars which will be specified in a data-driven manner (see section 2.2.4), and $\mathbf{I}$ denotes an identity matrix. The key difference with the previous model is the prior distribution defined on the model parameters $\mathbf{W}$ (see Table 2.1 for a comparison), which allows us to integrate over all possible candidates of $\mathbf{W}$ in computing the predictive distribution $\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \alpha, \beta)$ in lieu of resorting to a 'best' point estimate. Assuming the noise isotropy is equivalent to assuming all the output dimensions are statistically independent with the same variance, which is very unrealistic, and failing to capture correlational structures would be detrimental to the final forest estimate (eq. **??**). However,

applications in DT-SR and parameter mapping confirm that random-forest regression confers only a marginal improvement over decision trees, and so the isotropy assumption should not harm the prediction accuracy too much (although this claim needs attested). As a first attempt at this, we keep the implementation as simple as possible.

Assuming for now that hyperparameters α, β are known, the predictive distribution is computed by marginalising out the model parameters $\mathbf{W}$ as

$$\begin{aligned}
\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \alpha, \beta) &= \int \mathrm{P}(\mathbf{y}|\mathbf{x}, \alpha, \beta, \mathbf{W}) \cdot \mathrm{P}(\mathbf{W}_||\mathcal{D}, \alpha, \beta,) \, \mathbf{dW} && (2.2)\\
&= \int \mathrm{P}(\mathbf{y}|\mathbf{x}, \alpha, \beta, \mathbf{W}) \cdot \frac{\mathrm{P}(\mathbf{Y}|\mathbf{X}, \mathbf{W}, \alpha, \beta)\mathrm{P}(\mathbf{W}_|)}{\mathrm{P}(\mathbf{Y}|\mathbf{X}, \alpha, \beta)} \, \mathbf{dW} && (2.3)\\
&= \int \mathcal{N}(\mathbf{y}|\mathbf{W}\mathbf{x}, \beta^{-1}\mathbf{I}) \cdot \mathcal{N}(\mathbf{W}_|| \beta(\mathbf{Y}\mathbf{X}^T\mathbf{A}^{-T})_|, \, \oplus_{i=1}^{H}\mathbf{A}^{-1}) \, \mathbf{dW} && (2.4)\\
&= \mathcal{N}(\mathbf{y}| \, \mathbf{W}_{\mathrm{Pred}}\mathbf{x}, \, \sigma^2_{\mathrm{Pred}}(\mathbf{x}) \cdot \mathbf{I}) && (2.5)
\end{aligned}$$

where the i^{th} columns of matrices $\mathbf{X}$ and $\mathbf{Y}$ are given by $\mathbf{x}_i$ and $\mathbf{y}_i$, the mean linear map $\mathbf{W}_{\mathrm{Pred}} = \mathbf{Y}\mathbf{X}^T\big(\mathbf{X}\mathbf{X}^T + \frac{\alpha}{\beta}\mathbf{I}\big)^{-1}$ and the variance $\sigma^2_{\mathrm{Pred}}(\mathbf{x}) = \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} + \beta^{-1}$ with $\mathbf{A} = \alpha\mathbf{I} + \beta\mathbf{X}\mathbf{X}^T$. The mean differential entropy in equation (2.1) can be computed as $H(\mathcal{D}) = N_h p_h |\mathcal{D}|^{-1} \sum_{\mathbf{x}\in\mathcal{D}} \log(\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} + \beta^{-1})$ (up to additive constants) and so the information gain in eq **??** can be written as

$$\begin{aligned}
\mathrm{IG}(f_m, \tau, \mathcal{D}) = \sum_{\mathbf{x}\in\mathcal{D}} \log(\mathbf{x}^T\mathbf{A}(\mathcal{D})^{-1}\mathbf{x} + \beta^{-1}(\mathcal{D})) &- \sum_{\mathbf{x}\in\mathcal{D}_{\mathrm{L}}} \log(\mathbf{x}^T\mathbf{A}(\mathcal{D}_{\mathrm{L}})^{-1}\mathbf{x} + \beta^{-1}(\mathcal{D}_{\mathrm{L}}))\\
&- \sum_{\mathbf{x}\in\mathcal{D}_{\mathrm{R}}} \log(\mathbf{x}^T\mathbf{A}(\mathcal{D}_{\mathrm{R}})^{-1}\mathbf{x} + \beta^{-1}(\mathcal{D}_{\mathrm{R}}))
\end{aligned}$$

where multiplicative/additive constants are dropped and $\mathbf{A}(\mathcal{D}), \beta(\mathcal{D})$ denote the dependence on $\mathcal{D}$.

The predictive variance $\sigma^2_{\mathrm{Pred}}(\mathbf{x})$ provides an informative measure of uncertainty over the enhanced patch $\mathbf{y}(\mathbf{x})$ by combining two quantities: the degree of noise in the training data, β^{-1} and the degree of 'familiarity', $\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}$ which measures how different the input patch $\mathbf{x}$ is from the observed data. For example, if $\mathbf{x}$ contains previously unseen features such as pathology, the familiarity term becomes large, indicating high uncertainty. The equivalent measure for the original IQT, however, solely consists of the term, β^{-1} determined from the training data, and yields a fixed uncertainty estimate for any new input $\mathbf{x}$ (see Fig.**??** (left) and the second row in Table 2.1). Once a full BIQT forest $\mathcal{F} = \{T_i\}$ is grown, we perform reconstruction in the same way as before. All leaf nodes are endowed with predictive distributions of the form in equation (2.5). Denoting by $\sigma^2_{\mathrm{Pred},T_i}(\mathbf{x})$ the predictive variance at the node in i^{th} tree to which $\mathbf{x}$ has traversed, BIQT quantifies the *uncertainty* over the corresponding output as the ensemble average of the predictive variances over trees in the forest $\mathcal{F}$

$$\langle\sigma^2_{\mathrm{Pred}}(\mathbf{x})\rangle_{\mathcal{F}} = |\mathcal{F}|^{-1} \sum_{T\in\mathcal{F}} \sigma^2_{\mathrm{Pred},T}(\mathbf{x}). \tag{2.6}$$

2.2.4 Hyperparameter optimisation

A priori the hyper-parameters α and β are unknown, so we optimise them by maximising the marginal likelihood $\mathrm{P}(\mathcal{D}|\alpha, \beta)$. Since $\mathbf{W}_{\mathrm{Pred}}$ minimises the L2 regularised error,

$$\sum_{i=1}^{D} ||\mathbf{y}_i - \mathbf{W}\mathbf{x}_i||_2^2 + \frac{\alpha}{\beta}||\mathbf{W}_|||_2^2$$

	Maximum likelihood	Bayesian
Model	$\mathrm{P}(\boldsymbol{\eta}\|\beta) = \mathcal{N}(\boldsymbol{\eta}\|\mathbf{0}, \beta^{-1}\mathbf{I})$ $\mathrm{P}(\mathbf{y}\|\mathbf{x}, \mathbf{W}) = \mathcal{N}(\mathbf{y}\|\mathbf{W}\mathbf{x}, \beta^{-1}\mathbf{I})$	$\mathrm{P}(\boldsymbol{\eta}\|\beta) = \mathcal{N}(\boldsymbol{\eta}\|\mathbf{0}, \beta^{-1}\mathbf{I})$ $\mathrm{P}(\mathbf{W}_{\|}\|\alpha) = \mathcal{N}(\mathbf{W}_{\|}\|\mathbf{0}, \alpha^{-1}\mathbf{I})$ $\mathrm{P}(\mathbf{y}\|\mathbf{x}, \mathbf{W}) = \mathcal{N}(\mathbf{y}\|\mathbf{W}\mathbf{x}, \beta^{-1}\mathbf{I})$
$\mathrm{P}(\mathbf{y}\|\mathbf{x}, \mathcal{D}, \mathcal{H})$	$\mathcal{N}(\mathbf{y}\|\mathbf{W}_{ML}\mathbf{x}, \beta_{ML}^{-1})$ where: $\mathbf{W}_{ML} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$ $\beta_{ML}^{-1} = \frac{1}{N_h p_h \lvert\mathcal{D}\rvert}\sum_{i=1}^{\lvert\mathcal{D}\rvert}\lvert\lvert\mathbf{y}_i - \mathbf{W}_{ML}\mathbf{x}_i\rvert\rvert_2^2$ $\left(\mathbf{W}_{ML}, \beta_{ML} = \arg\max_{\mathbf{W},\beta}\mathrm{P}(\mathcal{D}\|\mathbf{W}, \beta)\right)$	$\mathcal{N}(\mathbf{y}\|\mathbf{W}_{\mathrm{MAP}}\mathbf{x}, \beta_{ML}^{-1} + \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x})$ where: $\mathbf{W}_{\mathrm{Pred}} = \mathbf{Y}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T + \frac{\alpha_{ML}}{\beta_{ML}}\mathbf{I}\right)^{-1}$ $\mathbf{A} = \alpha_{ML}\mathbf{I} + \beta_{ML}\mathbf{X}\mathbf{X}^T$ $\alpha_{ML}, \beta_{ML} = \arg\max_{\alpha,\beta}\mathrm{P}(\mathcal{D}\|\alpha, \beta)$

Table 2.1: . Comparison of ML and Bayesian linear models with isotropic Gaussian noise .

this optimisation procedure can be viewed as a data-driven determination of regularisation coefficient α/β. Although a closed form for $\mathrm{P}(\mathcal{D}|\alpha, \beta)$ exists, exhaustive search is impractical as we have to solve this problem for every binary splitting (characterised by a feature and a threshold) at all internal nodes of the tree. We thus derive and use the multi-output generalisation of the Gull-Mackay fixed-point iteration algorithm [14]

$$\beta_{\mathrm{new}} = \frac{1 - \beta_{\mathrm{old}} \cdot |\mathcal{D}|^{-1}\mathrm{trace}(\mathbf{A}(\alpha_{\mathrm{old}}, \beta_{\mathrm{old}})^{-1}\mathbf{X}\mathbf{X}^T)}{\frac{1}{|\mathcal{D}|N_h p_h}\sum_{j=1}^{N_h p_h}\sum_{i=1}^{|\mathcal{D}|}[y_{ji} - \boldsymbol{\mu}_j(\alpha_{\mathrm{old}}, \beta_{\mathrm{old}})^T\mathbf{x}_i]^2} \tag{2.7}$$

$$\alpha_{\mathrm{new}} = \frac{N_l p_l - \alpha_{\mathrm{old}} \cdot \mathrm{trace}(\mathbf{A}(\alpha_{\mathrm{old}}, \beta_{\mathrm{old}})^{-1})}{\frac{1}{N_h p_h}\sum_{j=1}^{N_h p_h}\boldsymbol{\mu}_j(\alpha_{\mathrm{old}}, \beta_{\mathrm{old}})^T\boldsymbol{\mu}_j(\alpha_{\mathrm{old}}, \beta_{\mathrm{old}})} \tag{2.8}$$

where $\boldsymbol{\mu}_j(\alpha, \beta) = \beta \cdot \mathbf{A}(\alpha, \beta)^{-1}\sum_{i=1}^{D} y_{ji}\mathbf{x}_i$. Whilst the standard MATLAB optimisation solver (e.g. `fminunc`) requires at least 50 times more computational time per node optimisation than for IQT, this iterative method is only average 2.5 times more expensive, making the Bayesian extension viable. We use this over Expectation Maximisation algorithm (viewing $\mathbf{W}_{|}$ as a latent variable) for its twice-as-fast convergence rate.

It is possible define prior distributions $\mathrm{P}(\alpha, \beta)$ on the hyper-parameters (called *hyperprior*), so we can obtain a more general predictive distribution by marginalising them out:

$$\mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int \mathrm{P}(\mathbf{y}|\mathbf{x}, \mathcal{D}, \alpha, \beta) \cdot \mathrm{P}(\alpha, \beta|\mathcal{D})\, d\alpha\, d\beta.$$

However, computing the integral over the hyperparameter posterior $\mathrm{P}(\alpha, \beta|\mathcal{D})$ is numerically challenging and approximations are usually required. Given that every node optimization requires hundreds of hyper-parameter optimisation, adding another layer of expensive approximation such as variational methods or Monte Carlo sampling may impose too much computational burden. One popular strategy is to specify the MAP estimate of the hyper-parameters i.e. $\mathrm{argmax}_{\alpha,\beta}\mathrm{P}(\alpha, \beta|\mathcal{D})$. The currently employed approach effectively aims to finds MAP estimates with flat hyperprior distributions, indicating no prior knowledge about α, β which is a reasonable assumption.

2.3 Experiments and results

Here we demonstrate and evaluate BIQT through the SR of DTI. First we describe the formulation of the application. Second, we compare the baseline performance on the HCP data to the

original IQT. Lastly, we demonstrate on clinical images of diseased brains that our uncertainty measure highlights pathologies.

2.3.1 Testing on HCP dataset.

We test BIQT on another set of 8 subjects from the HCP cohort. To evaluate reconstruction quality, three metrics are used: the root-mean-squared-error of the six independent DT elements (DT RMSE); the Peak Signal-to-Noise Ratio (PSNR); and the mean Structural Similarity (MSSIM) index [15]. We super-resolve each DTI after downsampling by a factor of 2, and these quality measures are then computed between the reconstructed HR image and the ground-truth. BIQT displays highly statistically significant ($p < 10^{-8}$) improvements (see Fig. 2.2) on all three metrics over IQT, linear regression methods and a range of interpolation techniques. In addition, trees obtained with BIQT are generally deeper than those of the original IQT.

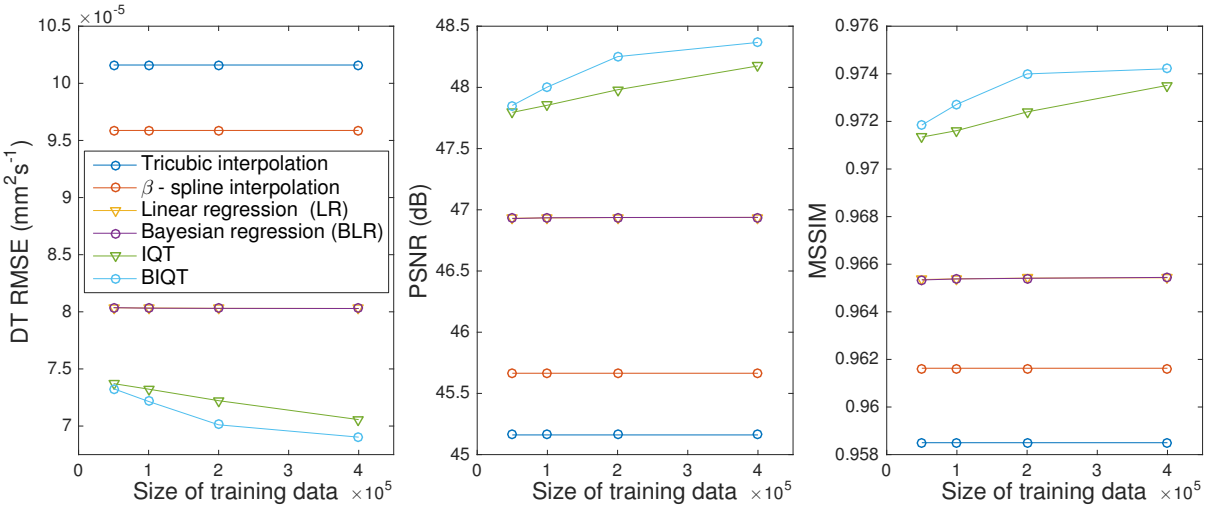


Figure 2.2: Three reconstruction metrics of various SR methods as a function of training data size; RMSE (left), PSNR (middle) and MSSIM (right). The performance of LR (yellow) and BLR (purple) coincide. The results for linear and nearest-neighbour interpolation are omitted for their poor performance.

Standard linear regression performs as well as the Bayesian regression due to the large training data size. However, with BIQT, as you descend each tree, the number of training data points at each node gets smaller, increasing the degree of uncertainty in model fitting, and so the data-driven regularisation performed in each node-wise Bayesian regression becomes more effective, leading to better reconstruction quality. This is also manifested in the deeper structure of BIQT trees, indicating more successful validation tests and thus greater generalisability. Moreover, BIQT performs reconstruction almost as efficiently as the original IQT, taking only a few minutes for full volume.

Fig. 2.3 shows reconstruction accuracies and uncertainty maps for BIQT and IQT. The uncertainty map of BIQT is more consistent with its reconstruction accuracy when compared to the original IQT. Higher resemblance is also observed between the distribution of accuracy (RMSE) and uncertainty (variance). The BIQT uncertainty map also highlights subtle variations in the reconstruction-quality within the white matter, whereas the IQT map contains flatter contrasts with discrete uncertainties that vary greatly in the same region (see histograms in bottom row). This improvement reflects the positive effect of the data-driven regularisation and better generalisability of BIQT and can be observed particularly in the splenium and genu

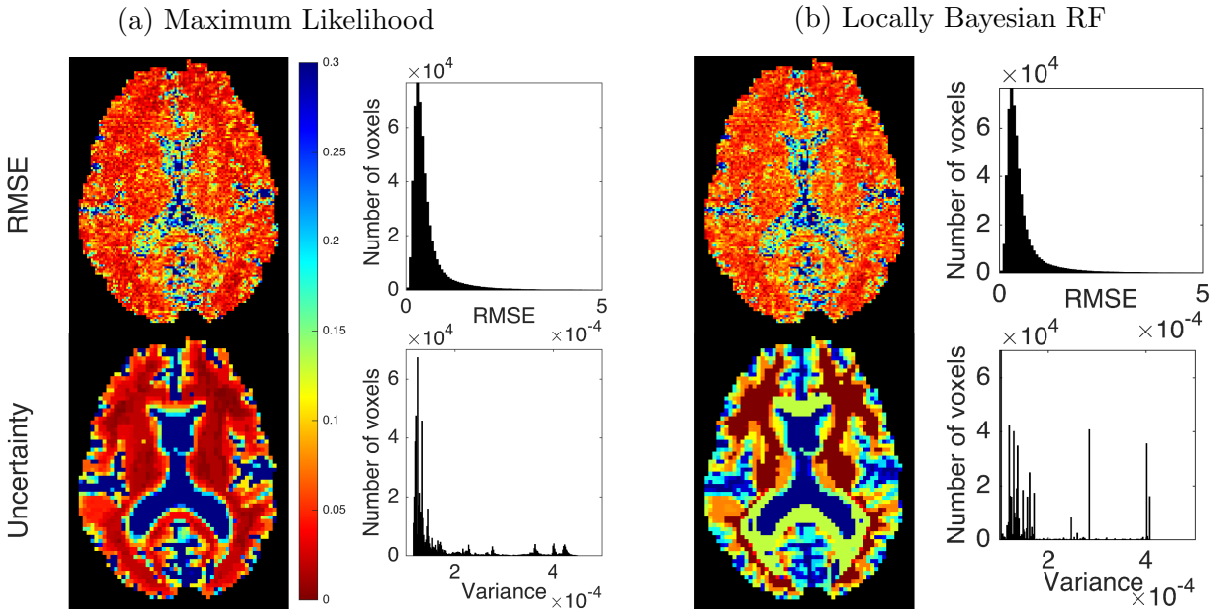


Figure 2.3: Reconstruction accuracy and uncertainty maps. (top row) The voxel-wise RMSE as a normalised colour-map and its distribution; (bottom row) Uncertainty map (variance) over the super-resolved voxels and its distribution for (a) BIQT and (b) IQT. Trees were trained on $\approx 4 \times 10^5$ patch pairs.

of the Corpus Callosum, where despite good reconstruction accuracy, IQT assigns higher uncertainty than in the rest of the white matter and BIQT indicates a lower and more consistent uncertainty. Thus, the BIQT uncertainty map displays higher correspondence with accuracy and allows for a more informative assessment of reconstruction quality. Note that while the uncertainty measure for IQT is governed purely by the training data, for BIQT the uncertainty also incorporates the familiarity of the test data.

2.3.2 Testing on pathological brains

We further validate our method on images with previously unseen abnormalities; we use trees trained on healthy subjects from HCP dataset to super-resolve DTIs of MS and brain tumour patients (10 each). We process the raw data (DWI) as before, and only use $b = 1200$ s/mm^2 measurements for the MS dataset and $b = 700$ s/mm^2 for the tumour dataset. The voxel size for both datasets is 2^3 mm^3. The MS dataset also contains lesion masks manually outlined by a neurologist. Fig. 2.4(a),(c) middle row shows that the uncertainty map of BIQT precisely identifies previously unseen features (pathologies in this case) by assigning lower confidence than for the remaining healthy white matter. Moreover, in accordance with the reconstruction accuracy, the prediction is more confident in pathological regions than in the cerebrospinal fluid (CSF). This is expected since the CSF is essentially free water with low SNR and is also affected by cardiac pulsations, whereas the pathological regions are contained within the white matter and produce better SNR. Each BIQT tree appropriately sends pathological patches into the 'white-matter' subspace and its abnormality is detected there by the 'familiarity' term, leading to a lower confidence with respect to the healthy white matter. By contrast, IQT sends pathological patches into the CSF subspace and assigns the fixed corresponding uncertainty which is higher than what it should be. In essence, BIQT enables an uncertainty measure which highly correlates with the pathologies in a much more plausible way, and this is achieved by its more effective partitioning of the input space and uncertainty estimation conferred by

Bayesian inference. Moreover, Fig. 2.4(b) shows the superior generalisability of BIQT even in reconstruction accuracy (here SR is performed on downsampled clinical DTIs); the RMSE of BIQT for MS patients is even smaller than that of IQT for healthy subjects.

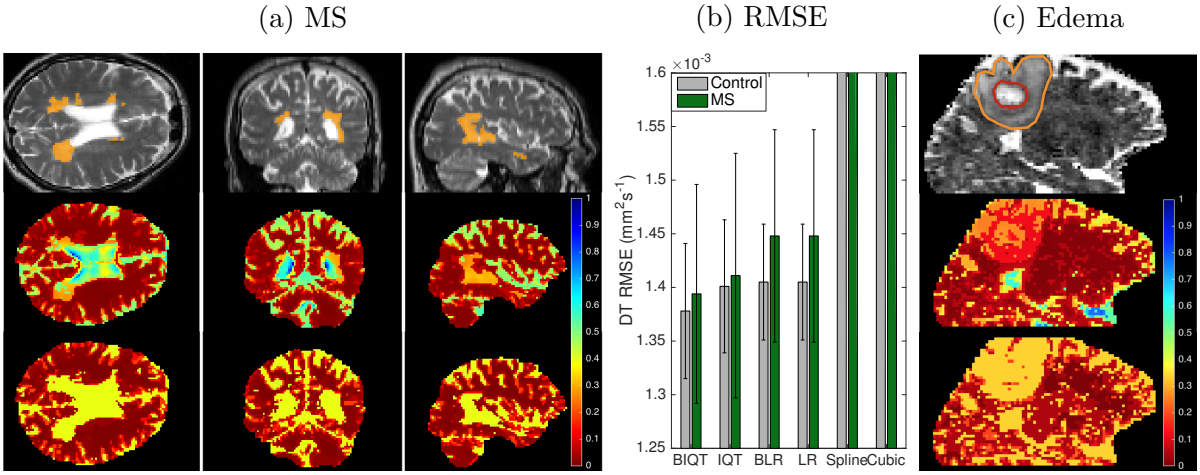


Figure 2.4: (a),(c) Normalised uncertainty map (variance is shown i.e. the smaller the more certain) for BIQT (middle row) and IQT (bottom row) along with the T2-weighted slices (top row) for MS (with focal lesions in orange) and edema (contours highlighted), respectively. (b). The RMSE for MS and control subjects (averaged over 10 subjects in each case).

2.4 Summary

We presented a computationally viable Bayesian extension of Image Quality Transfer (IQT). The application in super resolution of DTI demonstrated that the method not only achieves better reconstruction accuracy even in the presence of pathology (Fig. 2.4b) than the original IQT and standard interpolation techniques, but also provides an uncertainty measure which is highly correlated with the reconstruction quality. Furthermore, the uncertainty map is shown to highlight focal pathologies not observed in the training data. BIQT also performs a computationally efficient reconstruction. Although we have only applied BIQT to DTI-SR, the method preserves the generality of IQT and future work will investigate its performance on tractography and parameter mapping applications. Also, diagnostic values of the method need to be validated in larger-scale experiments.

Chapter 3

Taxonomy of Uncertainty in Deep Learning

Deep learning (DL) has shown great potential in medical image enhancement problems, such as super-resolution or image synthesis. However, to date little consideration has been given to uncertainty quantification over the output image. Here we introduce methods to characterise different components of uncertainty in such problems and demonstrate the ideas using diffusion MRI super-resolution. Specifically, we propose to account for *intrinsic uncertainty* through a heteroscedastic noise model and for *parameter uncertainty* through approximate Bayesian inference, and integrate the two to quantify *predictive uncertainty* over the output image. Moreover, we introduce a method to propagate the predictive uncertainty on a multi-channelled image to derived scalar parameters, and separately quantify the effects of intrinsic and parameter uncertainty therein. The methods are evaluated for super-resolution of two different signal representations of diffusion MR images—Diffusion Tensor images and Mean Apparent Propagator MRI—and their derived quantities such as mean diffusivity and fractional anisotropy, on multiple datasets of both healthy and pathological human brains. Results highlight three key potential benefits of uncertainty modelling for improving the safety of DL-based image enhancement systems. Firstly, incorporating uncertainty modelling improves the predictive performance even when test data departs from training data. Secondly, the predictive uncertainty highly correlates with reconstruction errors, and is therefore capable of detecting predictive "failures". Results on both healthy subjects and patients with brain glioma or multiple sclerosis demonstrate that such an uncertainty measure enables subject-specific and voxel-wise risk assessment of the super-resolved images that can be accounted for in subsequent analysis. Thirdly, we show that the method for decomposing predictive uncertainty into its independent sources provides high-level "explanations" for the model performance by separately quantifying how much uncertainty arises from the inherent difficulty of the task or the limited training examples. The introduced concepts of uncertainty modelling extend naturally to many other imaging modalities and data enhancement applications.

3.1 Introduction

In the last few years, deep learning techniques have permeated the field of medical image processing [16, 17]. Beyond the automation of existing radiological tasks— e.g. segmentation [18], detection [19], disease grading and classification [20]—deep learning has been applied to a diverse set of "data enhancement" problems. Data enhancement aims to improve the quality, the information content, or the quantity of medical images available for research and clinics by transforming images from one domain to another [21]. Previous research has shown the

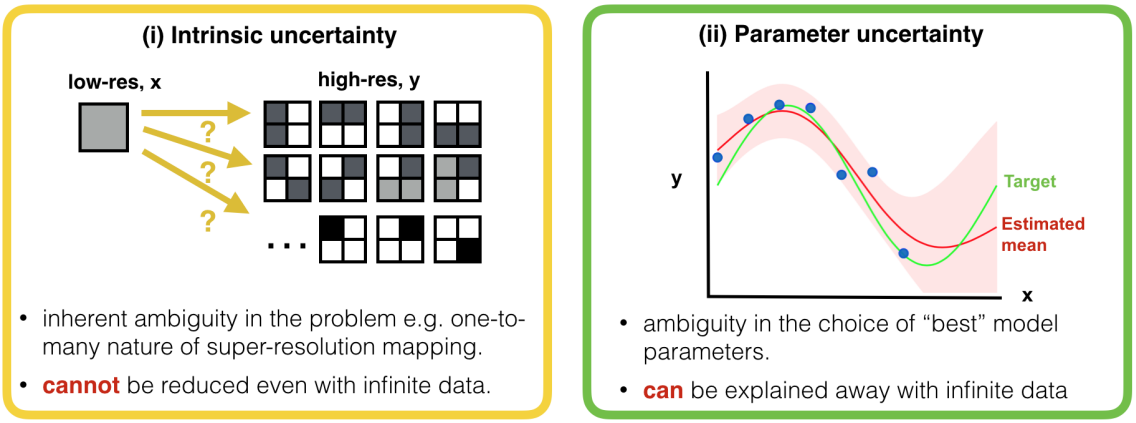


Figure 3.1: Illustration of two different types of uncertainty [44]. Intrinsic uncertainty [45] quantifies the degree of inherent ambiguity in the underlying problem. For example, in the case of super-resolution, there exist many possible high-resolution images $\mathbf{y}$ that would get mapped onto the same low-resolution input $\mathbf{x}$. Intrinsic uncertainty is irreducible with training data. On the other hand, the parameter uncertainty [46] (a subtype of model uncertainty) arises from the finite training set. There exist more than one model that can explain the given training data equally well, and the parameter uncertainty quantifies the ambiguity in selecting the model parameters that best captures the target data-generating process. As illustrated in the figure on the right, parameter uncertainty decreases with more data; the green line shows the target function, the red line is the estimated mean, and the shaded region signifies the associated parameter uncertainty (standard deviation), which is higher in regions where we have fewer observations.

efficacy of data enhancement in different forms such as super-resolution [22, 23, 24], image synthesis [25, 26], denoising [27, 28], data harmonisation [29, 30] across scanners and protocols, reconstruction [31, 32, 33, 34, 35, 36, 37], registration [38, 39] and quality control [40, 41]. These advances have the potential not only to enhance the quality and efficiency of radiological care, but also facilitate scientific discoveries in medical research through increased volume and content of usable data.

However, most efforts in the development of data enhancement techniques have focused on improving the accuracy of deep learning algorithms, with little consideration of risk management. Blindly trusting the output of a given machine learning tool risks undetected failures e.g. spurious features and removal of structures [42]. In medical applications, images inform scientific conclusions in research, and diagnostic, prognostic and interventional decisions in clinics. Therefore, translation of current proofs of principle to such safety-critical applications demands mechanisms for quantifying the risks of failures i.e. quantification of uncertainty/confidence and explanation of its source [43].

Predictive failures of deep learning systems, by and large, occur due to two reasons: i) the task itself is inherently ambiguous or ii) the learned model is not adequate to describe the data [44, 47, 48, 49], as illustrated in Fig. 3.1. The former stems from *intrinsic uncertainty* [45], which describes ambiguity in the underlying data generating process (e.g. presence of stochasticity such as measurement noise and intrinsic ill-posed nature of the problem), and cannot be alleviated by increasing available training data or model complexity[1]. The latter is characterised by *model uncertainty*[46], which describes ambiguity in model specification[2].

[1]Intrinsic uncertainty is also known as *aleaotoric* or statistical uncertainty.

[2]Model uncertainty is a subclass of *epistemic uncertainty* [44] which encompasses types of uncertainties that arise from lack of knowledge.

Model uncertainty arises from a) *parameter uncertainty*: ambiguity in fitting the model to the target mapping due to limited training data, or b) *model bias*: errors due to insufficient flexibility of the model class (e.g. fitting a linear model to a sinusoidal process). These types of uncertainty can be reduced by collecting more data or specifying a different class of models. With the expressivity of deep neural networks, which are known to be universal approximators [50] if sufficiently large, one might reasonably assume that the model bias is small enough to be discounted. Under this assumption, intrinsic and parameter uncertainty (Fig. 3.1) fully characterise the predictive failures of deep learning models. Therefore, accurate estimation of these uncertainties are needed and would potentially allow practioners to understand better the limits of the models, flag doubtful predictions, and highlight test cases that are not well represented in the training data.

In this work, we introduce methods for capturing components of uncertainty in medical image enhancement systems based on deep learning. We propose to model intrinsic uncertainty through a input-dependent (heteroscedastic) noise model [51] and parameter uncertainty through variational dropout [52]. We then combine and propagate these two "source" uncertainties into a spatial map of *predictive uncertainty* over the output image, which can be used to assess the output reliability on subject-specific and voxel-wise basis. Lastly, we propose a method to propagate the predictive uncertainty to arbitrary derived quantities of the output images, such as scalar indices that are commonly used for subsequent analysis, and decompose it into distinct components which separately quantify the contributions of intrinsic and parameter uncertainty. This paper demonstrates the benefits of these ideas to enhancing system safety within the context of Image Quality Transfer (IQT) [53, 1, 54, 55], a data-enhancement framework for propagating information from rare or expensive high quality images to lower quality but more readily available images. We focus on the application of IQT to *super-resolution* of diffusion magnetic resonance imaging (dMRI) scans, and evaluate the utility of uncertainty quantification in terms of three aspects; i) performance on unseen datasets; ii) safety assessment of system output; iii) explainability of failures. For two different types of diffusion signal representations, we evaluate the effects of uncertainty modeling on generalisation by measuring the predictive accuracy on unseen test subjects in the Human Connectome Project (HCP) dataset [10] and the Lifespan dataset [56]. We additionally test the value of improved predictive performance in a downstream tractography application. We then test the capability of the predictive uncertainty map to indicate predictive errors and thus to detect potential failures on images of both healthy subjects and those in which pathologies unseen in the training data arise, specifically from glioma and multiple-sclerosis (MS) patients. Lastly, we perform the decomposition of predictive uncertainty on HCP subjects with benign abnormalities, and assess its potential value in gaining high-level interpretations of predictive performance.

3.2 Related Works

This section provides a review of related works under several different themes. We first review the development of learning-based image enhancement methods in medical imaging applications. We then discuss the recent advances made to model and quantify uncertainty in such image enhancement problems. Lastly, we describe the existing strands of research in uncertainty modelling for other medical imaging problems and fields of applications.

Various forms of image enhancement can be cast as image transformation problems where the input image from one domain is mapped to an output image from another domain. Numerous recent methods have proposed to perform image transformation tasks as supervised regression of low quality against high quality image content. Alexander *et al.* [53] proposed Image Quality Transfer (IQT), a general framework for supervised quality enhancement of medical images. They demonstrated the efficacy of their method through a random forest (RF) implementation of

super-resolution (SR) of brain diffusion tensor images and estimation of advanced microstructure parameter maps from sparse measurements. More recently, deep learning, typically in the form of convolutional neural networks (CNNs), has shown additional promise in this kind of task. For example, Oktay *et al.* [22] proposed a CNN model to upsample a stack of 2D MRI cardiac volumes in the through-plane direction, where the SR mapping is learnt from 3D cardiac volumes of nearly isotropic voxels. This work was later extended by [57] with the addition of global anatomical prior based on auto-encoder. Zhao *et al.* [58] proposed a solution to the same SR problem for brains that utilises the high frequency information in in-plane slices to super-resolve in the through-plane direction without requiring external training data. In addition, a range of different architectures of CNNs have been considered for SR of other modalities and anatomical structures such as structural MRI [23] of brains, retinal fundus images [59] and computer tomography (CT) scans of chest [60]. Another problem of growing interest is image synthesis, which aims to synthesise an image of a different modality given the input image. Nie *et al.* [61] employed a conditional generative adversarial network to synthesise CT from MRI with fine texture details whilst Wolterink *et al.* [62] extended this idea using a CycleGAN [63] to leverage the abundance of unpaired training sets of CT and MR scans. In [64], a variant of CNN was applied to predict 7T images from 3T MRI, where both contrast and resolution are enhanced. Another notable application is the harmonisation of diffusion MRIs [29, 30, 55, 65] where images acquired at different scanners or magnetic field strengths are mapped to the common reference image space to allow for joint analysis.

Despite this advancement, all of these methods commit to a single prediction and lack a mechanism to communicate uncertainty in the output image. In medical applications where images can ultimately inform life-and-death decisions, quantifying reliability of output is crucial. Tanno *et al.* [1] aimed to address this problem for supervised image enhancement for the first time by proposing a Bayesian variant of random forests to quantify uncertainty over predicted high-resolution MRI. They showed that the uncertainty measure correlates well with the accuracy and can highlight abnormality not represented in the training data. In our preliminary work [48], we made an initial attempt to extend this approach with probabilistic deep-learning formulation, and showed that modelling different components of uncertainty—intrinsic and parameter uncertainty—allows one to build a more generalisable model and quantify predictive confidence. Kendall *et al.* [49] concurrently investigated the same problem in computer vision, suggesting its utility for safety-critical applications such as self-driving cars. More recently, Hu *et al.*[66] extended these works in the context of medical image segmentation and proposed a mechanism to learn the intrinsic uncertainty in a supervised manner, when multiple labels are available. Dalca *et al.* [67] proposed a CNN-based probabilistic model for diffeomorphic image registration with a learning algorithm based on variational inference, and demonstrated the state-of-the-art registration accuracy on established benchmarks while providing estimates of registration uncertainty. An alternative approach is ensembling where the variance of the predictions of multiple networks is used to quantify the predictive uncertainty [68]. Schlemper *et al.* [69] proposed a novel combination of the cascaded CNN architecture and compressive sensing, equipped with a variant of ensemble techniques, which enabled robust reconstruction of highly undersampled cardiovascular diffusion MR images, and quantification of reconstruction uncertainty. Bragman *et al.* [70] studied the value of uncertainty modelling for multi-task learning in the context of MR-only radiotherapy treatment planning where the synthetic CT image and the segmentation of organs at risk are simultaneously predicted from the input MRI image.

We should also note that, although not the focus of this work, research on uncertainty modelling in deep learning techniques extend to other medical image processing tasks beyond data enhancement, such as segmentation, detection and classification. For example, Nair *et al.*, [71] demonstrated for lesion segmentation of multiple sclerosis that the voxel-wise uncertainty metrics can be used for quality control; by filtering out predictions with high uncertainty, the model

could achieve higher lesion detection accuracy. A concurrent work by Eaton-Rosen *et al.* [72] showed for the task of brain tumour segmentation that the Monte Carlo (MC) sample variance from dropout [73] can be calibrated to provide meaningful error bars over estimates of tumour volumes. Similarly, [74] introduced ways to turn voxel-wise uncertainty score into structure-wise uncertainty metrics for brain parcellation task, and showed their values in performing more reliable group analysis. The uncertainty metric based on MC dropout has also shown promise in disease grading of retinal fundal images [75, 76], and more recently an extension based on test-time augmentation was introduced by [77]. An alternative approach is to train a model to predict uncertainty score directly; [78] showed that this approach is more effective when opinions from multiple experts are available for each image. Koh *et al.* [79] and Baumgartner *et al.* [80] proposed methods to generate a set of diverse and plausible segmentation proposals on a given image, capturing more realistically the high inter-reader annotation variability, which is commonly observed in medical image segmentation tasks. Lastly, [81, 82] demonstrated for the classification of mammograms and cardiac ultra-sound images, respectively that modelling uncertainty and biases of individual annotators enables robust learning from noisy labels in the presence of large disagreement.

However, within the context of medical image enhancement, these lines of research performed only limited validation of the quality and utility of uncertainty modelling. In this work, we formalise and extend the preliminary ideas in Tanno *et al.* [48] and provide a comprehensive set of experiments to evaluate the proposed uncertainty modelling techniques in a diverse set of datasets, which vary in demographics, scanner types, acquisition protocols or pathology. Moreover, with the exception of [48], none of the previous methods model different components of uncertainty, namely intrinsic and parameter uncertainty. Our method accounts for both, and provides conclusive evidence that this improves performance thanks to different regularisation effects. In addition, we propose a method to decompose predictive uncertainty over an arbitrary function of the output image (e.g. morphological measurements) into its sources, in order to provide a high-level explanation of model performance on the given input.

3.3 Methods

This section describes the methods for modelling different components of uncertainty that arise in data enhancement. Firstly, we provide an overview of Image Quality Transfer (IQT) which formulates data enhancement as a supervised learning problem. Secondly, using the IQT framework, we introduce methods to model *intrinsic* and *parameter uncertainty*, separately, focusing on the application of super-resolution. We then combine the two approaches and estimate the overall uncertainty over prediction (*predictive uncertainty*) by approximating the variance of the predictive distribution (eq. (3.9)). Lastly, we propose a method for decomposing predictive uncertainty into its sources—intrinsic and parameter uncertainty—in an attempt to provide quantifiable explanations for the confidence on model output (eq. (3.13)).

3.3.1 Background: Image Quality Transfer

Alexander *et al.* [53] proposed Image Quality Transfer (IQT), the first supervised learning based framework for data enhancement of medical images, and here we survey its general formulation which forms the testing ground of this work. IQT performs data enhancement via regression of low quality against high quality image content. In order to overcome the memory demands of processing 3-dimensional medical images, along with other subsequent work such as [83, 22, 64, 57], IQT assumes factorisability over local neighbourhoods (also called patches) and models the conditional distribution of high-quality image I_{High} given the corresponding low-quality input

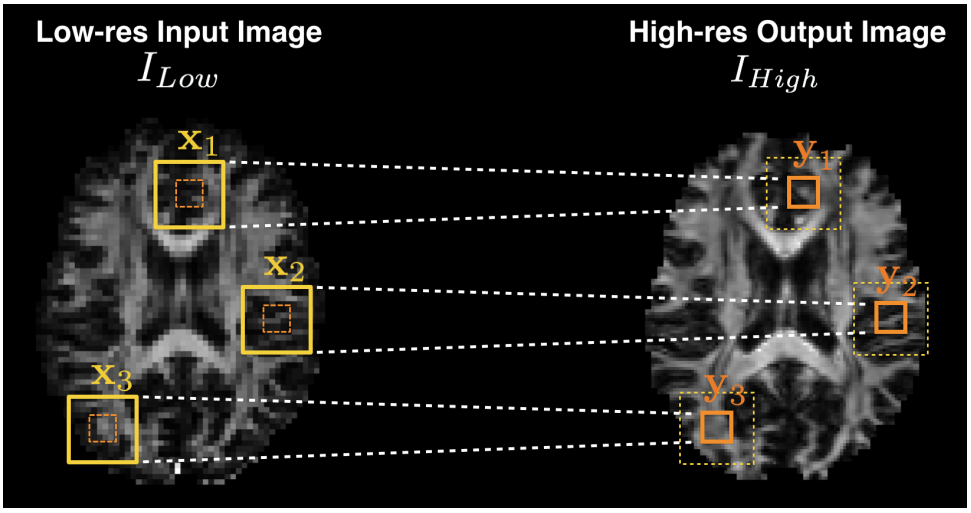


Figure 3.2: Illustration of the patch-wise regression in super-resolution application. The conditional distribution over the high quality image $p(I_{High}|I_{Low})$ is assumed to factorise over local neighbourhoods $\{(\mathbf{x}_i, \mathbf{y}_i)\}_i$. In this case, for each input subvolume $\mathbf{x}_i$ (in yellow), the high resolution version of the smaller centrally located neighbourhood, $\mathbf{y}_i$ (in orange) is regressed.

I_{Low} as:

$$p(I_{High}|I_{Low}) = \prod_{i \in \mathcal{S}} p(\mathbf{y}_i|\mathbf{x}_i) \tag{3.1}$$

where $\{\mathbf{y}_i\}_{i \in \mathcal{S}}$ is a set of disjoint high-quality subvolumes with $\mathcal{S}$ denoting the set of their indices, which together constitute the whole image I_{High}, while $\{\mathbf{x}_i\}_{i \in \mathcal{S}}$ is a set of potentially overlapping low-quality subvolumes, each of which contains and is spatially larger than the corresponding $\mathbf{y}_i$, as illustrated in Fig. 3.2. Here we assume that each local neighbourhood is a cubic sub-volume. The locality assumption reduces the problem of learning $p(I_{High}|I_{Low})$ to the much less memory intensive problem of learning $p(\mathbf{y}|\mathbf{x})$. In other words, IQT formulates the data enhancement task as a patch-wise regression where an input low-quality image I_{Low} is split into smaller overlapping sub-volumes $\{\mathbf{x}_i\}_{i \in \mathcal{S}}$ and the corresponding non-overlapping high-quality sub-volumes $\{\mathbf{y}_i\}_{i \in \mathcal{S}}$ are independently predicted according to the patch regressor $p(\mathbf{y}|\mathbf{x})$. The final prediction for the 3D high-quality volume I_{high} is constructed by tesellating the output patches $\{\mathbf{y}_i\}_{i \in \mathcal{S}}$.

The original implementation of IQT [53, 54, 1] employed a variant of random forests (RFs) to model $p(\mathbf{y}|\mathbf{x})$ while more recent [83, 22, 64, 57] approaches use variants of convolutional neural networks (CNNs). Either way, the machine learning algorithm is trained on pairs of high-quality and low-quality patches $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ extracted from a set of image volumes, and is used to perform the data-enhancement task of interest. Typically, such patch pairs $\mathcal{D}$ are synthesised by down-sampling a collection of high quality images to approximate their counterparts in a particular low-quality scenario [53, 22]. In this work, we focus on the task of super-resolution (SR) where the spatial resolution of I_{high} is higher than the input image I_{low}.

3.3.2 Baseline Super-Resolution Model: 3D-ESPCN

As the baseline architecture for modelling $p(\mathbf{y}|\mathbf{x})$, we adapt efficient subpixel-shifted convolutional network (ESPCN) [84] to 3D data. ESPCN is a recently proposed method with the capacity to perform real-time per-frame SR of videos while retaining high accuracy on 2D natural images. We have chosen to base on this architecture for its simplicity and computational

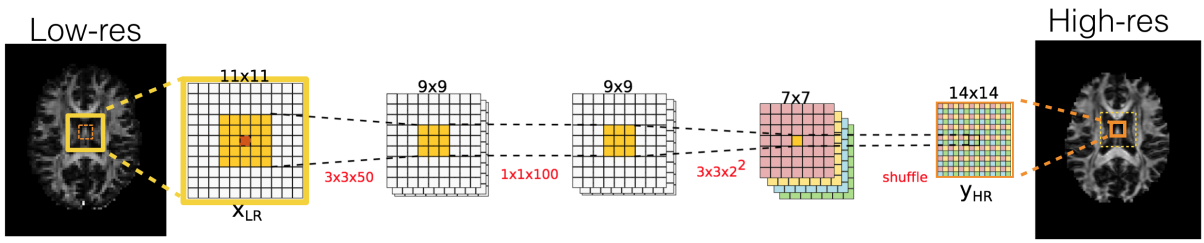


Figure 3.3: 2D illustration of an example baseline network (ESPCN [84]) with upsampling rate, $r = 2$. The receptive field of the central 2^2 pixels in the output patch is 5^2 pixels in the input patch and is shown in yellow. The shuffling operation at the end periodically rearranges the final feature maps from the low-resolution space into the high-resolution space.

performance. Most CNN-based SR techniques first up-sample a low-resolution input image (e.g. through bilinear interpolation[85], deconvolution[22, 86], fractional-strided convolution[87], etc) and then refine the high-resolution estimate through a series of convolutions. These methods suffer from the fact that (1) the up-sampling can be a lossy process and (2) refinement in the high-resolution space has a higher computational cost than in the low-resolution space. By contrast, ESPCN performs convolutions in the low-resolution-space, upsampling afterwards. The reduced resolution of feature maps dramatically decreases the computational and memory costs, which is more pronounced in processing 3D data.

More specifically the ESPCN is a fully convolutional network, with a special *shuffling operation* on the output, which identifies individual feature channel dimensions with spatial locations in the high-resolution output. Fig. 3.3 shows a 2D illustration of an example ESPCN when the fully convolutional part of the network consists of 3 convolutional layers, each followed by a ReLU, and the final layer has cr^2 feature maps where r is the upsampling rate and c is the number of channels in the output image (e.g. 6 in the case of DT images). The shuffling operation takes the feature maps of shape $h \times w \times cr^2$ and remaps pixels from different channels into different spatial locations in the high-resolution output, producing a $rh \times rw \times c$ image, where h and w denote height and width of the pre-shuffling feature maps. This shuffling operation in 3D is given by $\mathcal{S}(F)_{i,j,k,c} = F_{[i/r],[j/r],[k/r],(r^3-1)c+\mathrm{mod}(i,r)+r\cdot\mathrm{mod}(j,r)+r^3\cdot\mathrm{mod}(k,r)}$ where F is the pre-shuffled feature maps. The combined effects of the last convolution and shuffling is effectively a learned interpolation, and an efficient implementation of deconvolution layer [**?**] where the kernel size is divisible by the size of the stride [84]. Therefore, it is less susceptible to checker-board like artifacts commonly observed with deconvolution operations [88].

At test time, the prediction of higher resolution volume is performed through *shift-and-stitch* operation. The network takes each subvolume $\mathbf{x}$ in a low-resolution image, and predicts the corresponding high-resolution sub-volume $\mathbf{y}$. By tessellating the predictions from appropriately shifted inputs $\mathbf{x}$, the whole high-resolution volume is reconstructed. With convolutions being local operations, each output voxel is only inferred from a local region in the input volume, and the spatial extent of this local connectivity is referred to as the *receptive field*. For a given input subvolume, the network increases the resolution of the central voxel of each receptive field e.g. the central 2^3 output voxels are estimated from the corresponding 5^3 receptive field in the input volume, as coloured yellow in Fig. 3.3.

Given training pairs of high-resolution and low-resolution patches $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, we optimise the network parameters by minimising the sum of per-pixel mean-squared-error (MSE) between the ground truth $\mathbf{y}$ and the predicted high-resolution patch $\mu_\theta(\mathbf{x})$ over the training set. Here θ denotes all network parameters. This is equivalent to minimising the negative log likelihood (NLL) under the Gaussian noise model $p(\mathbf{y}|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{y}; \mu_\theta(\mathbf{x}), \sigma^2 I)$ with fixed isotropic variance σ^2.

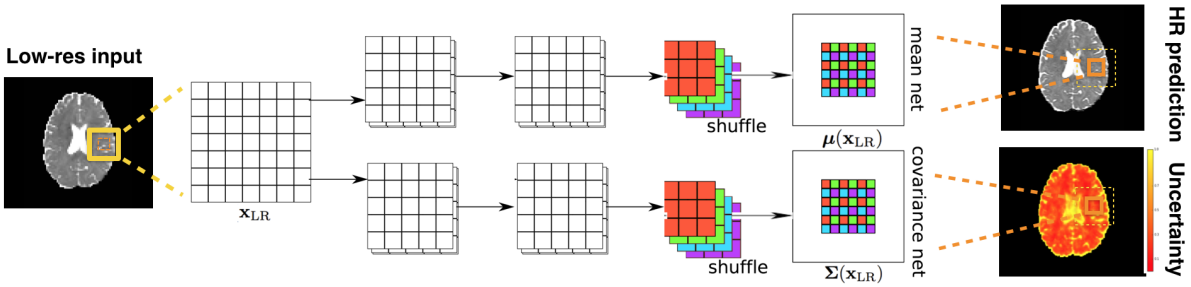


Figure 3.4: 2D illustration of the proposed dual-path architecture which estimates the mean and diagonal covariance of the Gaussian conditional distributions as functions of the input low-resolution subvolume $\mathbf{x}$. The "mean network" $\mu(\cdot)$ at the top generates the high-resolution prediction, while the "covariance network" $\Sigma(\cdot)$ at the bottom estimates the corresponding covariance matrix at the selected location in the volume. The diagonal entries of the covariance are used to quantify the intrinsic uncertainty. The parameters of both networks are learned by minimising the common loss function (eq. (3.5)).

3.3.3 Intrinsic Uncertainty and Heteroscedastic Noise Model

Intrinsic uncertainty quantifies the inherent ambiguity of the underlying problem that is irreducible with data as illustrated in Fig. 3.1(i). Here we capture intrinsic uncertainty by estimating the variance of the target conditional distribution $p(\mathbf{y}|\mathbf{x},\theta)$. In medical images, intrinsic uncertainty is often spatially and channel-wise varying. For example, super-resolution could be fundamentally harder on some anatomical structures than others due to signal variability as shown in [1]. It may also be the case that some channels of the image volume might contain more complex, non-linear and noisy signals than other channels e.g. higher order terms in diffusion signal representations. To capture such potential variation of intrinsic uncertainty, we model $p(\mathbf{y}|\mathbf{x},\theta)$ as a Gaussian distribution with input-dependent varying variance:

$$p(\mathbf{y}|\mathbf{x},\theta_1,\theta_2) = \mathcal{N}\big(\mathbf{y};\mu(\mathbf{x};\theta_1),\Sigma(\mathbf{x};\theta_2)\big) = \frac{\exp\Big(\big(\mathbf{y}-\mu(\mathbf{x};\theta_1)\big)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x};\theta_2)\big(\mathbf{y}-\mu(\mathbf{x};\theta_1)\big)\Big)}{\sqrt{(2\pi)^k \det\,\Sigma(\mathbf{x};\theta_2)}} \tag{3.2}$$

where the mean $\mu(\mathbf{x};\theta_1)$ and the covariance $\Sigma(\mathbf{x};\theta_2)$ are functions of input $\mathbf{x}$ and modelled by two separate 3D-ESPCNs (as shown in Fig. 3.4), which we refer to as "mean network" and "covariance network", and are parametrised by θ_1 and θ_2, respectively. We note that the input patch $\mathbf{x}$ varies spatially, which makes the estimated variance spatially varying and different for respective channels. Fig. 3.4 shows a 2D illustration of our 3D architecture. For each low-resolution input patch $\mathbf{x}$, we use the output of the mean network $\mu(\mathbf{x};\theta_1)$ at the top as the final estimate of the high-resolution ground truth $\mathbf{y}$ whilst the diagonal elements of the covariance $\Sigma(\mathbf{x};\theta_2)$ quantify the corresponding intrinsic uncertainty over individual components in $\mu(\mathbf{x};\theta_1)$ and over different channels. Lastly, we note that this is a specifc instance of a broad class of models, called *heteroscedastic noise models* [89, 51] where the variance is a function of the value of the input. In contrast, the baseline 3D-ESPCN can be viewed as an example of *homoscedastic noise models* with $\mathbf{y} = \mu_\theta(\mathbf{x}) + \sigma\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$ with constant variance σ^2 across all spatial locations and image channels, which is highly unrealistic in most medical images.

We jointly optimise the parameters $\theta = \{\theta_1, \theta_2\}$ of the mean network and the covariance

network by minimising the negative loglikelihood (NLL):

$$\mathcal{L}_\theta(\mathcal{D}) = \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}} -\log\, p(\mathbf{y}|\mathbf{x},\theta_1,\theta_2) \tag{3.3}$$

$$= \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}} -\log\, \mathcal{N}\big(\mathbf{y}_i;\mu(\mathbf{x}_i;\theta_1),\Sigma(\mathbf{x}_i;\theta_2)\big) \tag{3.4}$$

$$= \mathcal{M}_\theta(\mathcal{D}) + \mathcal{H}_\theta(\mathcal{D}) + c \tag{3.5}$$

where c is a constant and the remaining terms are given by

$$\mathcal{M}_\theta(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\big(\mathbf{y}_i-\mu(\mathbf{x}_i;\theta_1)\big)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}_i;\theta_2)\big(\mathbf{y}_i-\mu(\mathbf{x}_i;\theta_1)\big), \quad \mathcal{H}_\theta(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\log\,\det\,\Sigma(\mathbf{x}_i;\theta_2).$$

Here $\mathcal{M}_\theta(\mathcal{D})$ denotes the mean squared Mahalanobis distance with respect to the predictive distribution $p(\mathbf{y}|\mathbf{x},\theta)$. For simplicity, in this work we assume diagonality of the covariance matrix $\Sigma(\mathbf{x};\theta_2)$. This means that the Mahalanobis distance term $\mathcal{M}_\theta(\mathcal{D})$ equates to the sum of MSEs across all pixels and channels in the output, weighted by the inverse of the corresponding variance (estimated intrinsic uncertainty)[3]. This term naturally encourages assigning low uncertainty to regions with higher MSEs, robustifying the training to noisy labels and outliers. On other other hand, $\mathcal{H}_\theta(\mathcal{D})$ represents the mean differential entropy and discourages the spread of $\Sigma_{\theta_2}(\mathbf{x})$ from growing too large. We note that the covariance network is used to modulate the training of the mean network and quantify intrinsic uncertainty during inference while only the mean network generates the final prediction, requiring a single 3D-ESPCN to perform super-resolution.

3.3.4 Parameter Uncertainty and Variational Dropout

Parameter uncertainty signifies the ambiguity in selecting the parameters of the model that best describes the training data as illustrated in Fig. 3.1.(ii). The limitation of the previously introduced 3D-ESPCN baseline (Sec. 3.3.2) and its heteroscedastic extension (Sec. 3.3.3) is their reliance on a single estimate of network parameters. In many medical imaging problems, the amount of training data is modest; in such cases, this point estimate approach increases the risk of overfitting [73].

We combat this problem with a Bayesian approach. Specifically, instead of resorting to a single network of fixed parameters, we consider the (posterior) distribution over all the possible settings of network parameters given training data $p(\theta|\mathcal{D})$. This probability density encapsulates the parameter uncertainty, with its spread of mass describing the ambiguity in selecting most appropriate models to explain the training data $\mathcal{D}$. However, in practice, the posterior $p(\theta|\mathcal{D})$ is intractable due to the difficulty in computing the normalisation constant. We, therefore, propose to approximate $p(\theta|\mathcal{D})$ with a simpler distribution $q_\phi(\theta)$ [90]. Specifically, we adapt a technique called *variational dropout* [52] to convolution operations from its original version introduced for feedforward NNs.

Binary dropout [91] is a popular choice of method for approximating posterior distributions [73] with demonstrated utility in medical imaging applications [75, 83, 76, 74, 72, 71, 70]. However, typically hyper-parameters (dropout rates) need to be pre-set before the training, requiring inefficient cross-validation and thus substantially constraining the flexibility of approximate distribution family $q_\phi(\cdot)$ (often a fixed dropout rate per layer). This limitation motivates us to use variational dropout [52] that extends such approach with a way to learn the dropout rate from

[3] In the case of full covariance, $\mathcal{M}_\theta(\mathcal{D})$ becomes the MSE in the basis of principle components, weighted by the corresponding eigenvalues.

data for every single weight in the network and theoretically enables a more effective approximation of the posterior distribution. Another established class of methods is stochastic gradient Markov chain Monte Carlo (SG-MCMC) method [92, 93, 94, 95]. However, in this work, we do not not consider SG-MCMC methods because they remain, although unbiased, computationally inefficient due to the requirement of evaluating an ensemble of models for posterior computation, and are slow to converge for high-dimensional problems.

Variational dropout [52] employs a form of variational inference to approximate the posterior $p(\theta|\mathcal{D})$ by a member of tractable family of distributions $q_\phi(\theta) = \prod_{ij} \mathcal{N}(\theta_{ij}; \eta_{ij}, \alpha_{ij}\eta_{ij}^2)$ parametrised by $\phi = \{\eta_{ij}, \alpha_{ij}\}_{ij}$, such that Kullback-Leibler (KL) divergence $\mathrm{KL}\big(q_\phi(\theta)||p(\theta|\mathcal{D})\big)$ is minimised. Here, θ_{ij} denotes an individual element in the convolution filters of CNNs as a random variable with parameters α_{ij}(dropout rate) and η_{ij} (mean), and the posterior over the set of all weights is effectively approximated with a product of univariate Gaussian distributions. In practice, introducing a prior $p(\theta)$ and applying Bayes' rule allow us to rewrite the minimization of the KL divergence as maximization of the quantity known as the evidence lower bound (ELBO) [90]. Here during training, we learn the variational parameters $\phi = \{\eta_{ij}, \alpha_{ij}\}_{ij}$ by minimizing the negative ELBO (to be consistent with the NLL cost function in eq.(3)):

$$\mathcal{L}_\phi(\mathcal{D}) = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}} \Big(\mathbb{E}_{q_\phi(\theta)}[-\log\, p(\mathbf{y}_i|\mathbf{x}_i, \theta)] + \mathrm{KL}(q_\phi(\theta)||p(\theta)) \Big) \tag{3.6}$$

An accurate approximation for the KL term for log-uniform prior $p(\theta)$ is proposed in [96], which is employed here. On the other hand, the first term (referred to as the reconstruction term) cannot be computed exactly, thus we employ the following MC approximation by sampling S samples of network parameters from the posterior:

$$\mathbb{E}_{q_\phi(\theta)}[-\log\, p(\mathbf{y}|\mathbf{x}, \theta)] \approx \frac{1}{S} \sum_{s=1}^{S} -\log\, p(\mathbf{y}|\mathbf{x}, \theta^{(s)}), \quad \theta^{(s)} \sim q_\phi(\theta) \tag{3.7}$$

Adapting the local reparametrisation trick presented in [52] to a convolution operation, we derive the implementation of posterior sampling $\theta^{(s)} \sim q_\phi(\theta)$ such that the variance of gradients over each mini-batch is low [4]. In practice, this amounts to replacing each standard convolution kernel with a "Bayesian" convolution, which proceeds as follows. Firstly, we define two separate convolution kernels: $\eta \in \mathbb{R}^{c \times k^2}$ ("mean" kernels) and $\alpha \odot \eta^2 \in \mathbb{R}^{c \times k^2}$ ("variance" kernels) where $\odot$ denotes the element-wise multiplication, c is the number of input channel and k is the kernel width. Input feature maps F_{in} and its elementwise squared values are convolved by respective kernels to compute the "mean" and "variance" of the output feature maps $\mu_Y \triangleq F_{\text{in}} \star \eta$ and $\sigma_Y^2 \triangleq F_{\text{in}}^2 \star (\alpha \odot \eta^2)$. Lastly, the final output feature maps F_{out} are computed by drawing a sample from $\mathcal{N}(\mu_Y, \sigma_Y^2)$ i.e. computing the following quantity:

$$F_{\text{out}} \triangleq \mu_Y + \sigma_Y \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{3.8}$$

Every forward pass (i.e. computation of each $p(\mathbf{y}|\mathbf{x}, \theta^{(s)})$) with variational dropout is thus performed via a sequence of Bayesian convolutions. Since the injected Gaussian noise ϵ is independent of the variational parameters $\phi = \{\eta_{ij}, \alpha_{ij}\}_{ij}$, the approximate reconstruction term in eq. 3.7 is differentiable with respect to them [97].

3.3.5 Joint Modelling of Intrinsic and Parameter Uncertainty

We now describe how to combine the methods for modelling intrinsic and parameter uncertainty. Operationally, we take the dual architecture (Fig. 3.4) used to model intrinsic uncertainty, and apply variational dropout to every convolution layer in it. The intrinsic uncertainty is

[4] See the proof for feedforward networks given in [52] which generalises to convolutions

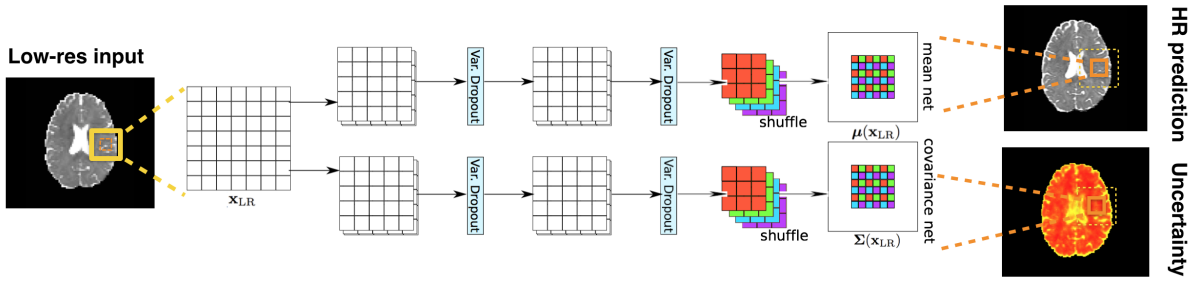


Figure 3.5: 2D illustration of a heteroscedastic network with variational dropout. Diagonal covariance is again assumed. The top 3D-ESPCN estimates the mean and the bottom one estimates the covariance matrix of the likelihood. Variational dropout is applied to feature maps after every convolution where Gaussian noise is injected into feature maps $F_{\text{out}} = \mu_Y + \sigma_Y \odot \epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$ (see eq. 3.8).

modelled in the heteroscedastic Gaussian model $p(\mathbf{y}|\mathbf{x}, \theta_1, \theta_2) = \mathcal{N}\big(\mathbf{y}; \mu(\mathbf{x}; \theta_1), \Sigma(\mathbf{x}; \theta_2)\big)$ while the parameter uncertainty is captured in the approximate posterior $q_\phi(\theta_1, \theta_2) \approx p(\theta_1, \theta_2|\mathcal{D})$ obtained from variational dropout.

At test time, for each low-resolution input subvolume $\mathbf{x}$, we would like to compute the predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ over the high-resolution output $\mathbf{y}$. We approximate this quantity by $q^*_\phi(\mathbf{y}|\mathbf{x})$ by taking the "average" of all possible network predictions $p(\mathbf{y}|\mathbf{x}, \theta) = \mathcal{N}\big(\mathbf{y}; \mu(\mathbf{x}; \theta_1), \Sigma(\mathbf{x}; \theta_2)\big)$ from all settings of the parameters θ_1, θ_2, weighted by the associated approximate posterior distribution $q_\phi(\theta_1, \theta_2)$. More formally, we need to compute the integral below:

$$q^*_\phi(\mathbf{y}|\mathbf{x}) \triangleq \int \underbrace{\mathcal{N}(\mathbf{y}; \mu(\mathbf{x}; \theta_1), \Sigma(\mathbf{x}; \theta_2)))}_{\text{Network prediction}} \cdot \underbrace{q_\phi(\theta_1, \theta_2)}_{\text{Approx. posterior}} \, \mathrm{d}\theta_1 \mathrm{d}\theta_2 \tag{3.9}$$

$$\approx \int p(\mathbf{y}|\mathbf{x}, \theta_1, \theta_2) \cdot p(\theta_1, \theta_2|\mathcal{D}) \mathrm{d}\theta_1 \mathrm{d}\theta_2 = p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \tag{3.10}$$

where the last line represents the true predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ which is estimated by our model $q^*_\phi(\mathbf{y}|\mathbf{x})$. However, in practice, the integral $q^*_\phi(\mathbf{y}|\mathbf{x})$ cannot be evaluated in closed form because the likelihood $\mathcal{N}\big(\mathbf{y}; \mu(\mathbf{x}; \theta_1), \Sigma(\mathbf{x}; \theta_2)\big)$ is a highly non-linear function of input $\mathbf{x}$ as given in eq. 3.2. At test time, we therefore estimate, for each input $\mathbf{x}$, the mean and covariance of the approximate predictive distribution $q^*_\phi(\mathbf{y}|\mathbf{x})$ with the unbiased Monte Carlo estimators:

$$\hat{\mu}_{\mathbf{y}|\mathbf{x}} \triangleq \frac{1}{T} \sum_{t=1}^{T} \mu(\mathbf{x}; \theta_1^t) \xrightarrow[T \to \infty]{} \mathbb{E}_{q^*_\phi(\mathbf{y}|\mathbf{x})}[\mathbf{y}] \tag{3.11}$$

$$\hat{\Sigma}_{\mathbf{y}|\mathbf{x}} \triangleq \frac{1}{T} \sum_{t=1}^{T} \Big(\Sigma(\mathbf{x}; \theta_2^t) + \mu(\mathbf{x}; \theta_1^t) \mu(\mathbf{x}; \theta_1^t)^{\mathrm{T}} \Big) - \hat{\mu}_{\mathbf{y}|\mathbf{x}} \hat{\mu}^{\mathrm{T}}_{\mathbf{y}|\mathbf{x}} \xrightarrow[T \to \infty]{} \mathrm{cov}_{q^*_\phi(\mathbf{y}|\mathbf{x})}[\mathbf{y}, \mathbf{y}] \tag{3.12}$$

where $\{(\theta_1^t, \theta_2^t)\}_{t=1}^T$ are samples of the network parameters (i.e. convolution kernels) drawn from the approximate posterior $q_\phi(\theta_1, \theta_2)$. In the other words, the inference performs T stochastic forward passes at test time by injecting noise into features according to eq. 3.8, and amalgamates the corresponding network outputs to compute the sample mean $\hat{\mu}_{\mathbf{y}|\mathbf{x}}$ and sample covariance $\hat{\Sigma}_{\mathbf{y}|\mathbf{x}}$. We use the sample mean $\hat{\mu}_{\mathbf{y}|\mathbf{x}}$ as the final prediction of an high-resolution ouput patch $\mathbf{y}$ and use the diagonal elements of the sample covariance $\hat{\Sigma}_{\mathbf{y}|\mathbf{x}}$ to quantify the corresponding uncertainty, which we refer to as *predictive mean* and *predictive uncertainty*, respectively.

3.3.6 Uncertainty Decomposition and Propagation

Predictive uncertainty arises from the combination of two source effects, namely intrinsic and parameter uncertainty, for which we have previously introduced methods for estimation. Lastly, we introduce a method based on variance decomposition for disentangling these effects and quantifying their contributions separately in predictive uncertainty. We consider such decomposition problem in the presence of an arbitrary transformation of the output variable $\mathbf{y}$.

The users of super-resolution algorithms are often interested in the quantities that are derived from the predicted high-resolution images, rather than the images themselves. For example, quantities such as the principal direction (first eigenvalue of the DT), mean diffusivity (MD) and fractional anisotropy (FA) are typically calculated from diffusion tensor images (DTIs) and used in the downstream analysis. We therefore consider an generic function[5] $g : \mathcal{Y} \rightarrow \mathbb{R}^m$ which transforms the high-resolution multi-channel data $\mathbf{y}$ to a quantity of interest e.g. MD and FA maps, and propose a way to propagate the predictive uncertainty over $\mathbf{y}$ to the transformed domain (i.e. compute the variance of $p(g(\mathbf{y})|\mathcal{D}, \mathbf{x})$) and decompose it into the "intrinsic" and "parameter" components. Specifically, by using the law of total variance [98], we perform the following decomposition:

$$\mathbb{V}_{p(\mathbf{y}|\mathbf{x},\mathcal{D})}[g(\mathbf{y})] = \Delta_m(g(\mathbf{y})) + \Delta_i(g(\mathbf{y})) \tag{3.13}$$

where the respective component terms are given by:

$$\Delta_m(g(\mathbf{y})) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\mathbb{V}_{p(g(\mathbf{y})|\theta,\mathbf{x},\mathcal{D})}[g(\mathbf{y})] - \mathbb{V}_{p(g(\mathbf{y})|\theta,\mathbf{x},\mathcal{D})}[g(\mathbf{y})|\theta]\right] \tag{3.14}$$

$$= \underbrace{\mathbb{V}_{p(\theta|\mathcal{D})}[\mathbb{E}_{p(g(\mathbf{y})|\theta,\mathbf{x},\mathcal{D})}[g(\mathbf{y})|\theta]]}_{\text{propagated parameter uncertainty}} \tag{3.15}$$

$$\Delta_i(g(\mathbf{y})) = \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathbb{V}_{p(g(\mathbf{y})|\theta,\mathbf{x},\mathcal{D})}[g(\mathbf{y})|\theta]]}_{\text{propagated intrinsic uncertainty}} \tag{3.16}$$

We refer to the components $\Delta_m(g(\mathbf{y}))$ and $\Delta_i(g(\mathbf{y}))$ as "propagated" parameter and intrinsic uncertainty. Intuitively, the first term quantifies the difference in variance between the cases where we have variable parameters and fixed parameters. In other words, this quantifies how much predictive uncertainty on the derived quantity arises, on average, from the variability in parameters. The second term on the other hand quantifies the average variance of the model prediction when the parameters are fixed, which signifies the model-independent uncertainty due to data i.e. intrinsic uncertainty. Assuming that the considered neural network is identifiable[6] and sufficiently complex to capture the underlying data generating process, as the amount of training data increases, the posterior $p(\theta|\mathcal{D})$ tends to a Dirac delta function and thus the first term diminishes to zero while the second term remains. A similar variance decomposition technique was employed in [99] to understand how the variation in cell signals of interest (e.g. gene expression) in a bio-chemical network is caused by the fluctuations of other environmental variables (e.g. transcription rate and biological noise). In our case, we employ the variance decomposition technique to separate the effects of network parameters from the intrinsic uncertainty in the prediction of $g(\mathbf{y})$.

[5] We assume here that the transform g is a measurable function with well-defined expectation and variance.

[6] We note that a neural network is, in general, not identifiable i.e. there exist more than a single set of parameters that capture the same target distribution $p(g(\mathbf{y})|\mathbf{x})$. In such cases, the posterior distribution $p(\theta|\mathcal{D})$ does not collapse to a single Dirac Delta function with infinite amount of observations—it rather converges to a mixture of all sets of network parameters Θ such that $p(g(\mathbf{y})|\theta^*, \mathbf{x}) = p(g(\mathbf{y})|\mathbf{x}) \forall \theta^* \in \Theta$. However, the expectation $\mathbb{E}_{p(g(\mathbf{y})|\theta,\mathbf{x},\mathcal{D})}[g(\mathbf{y})|\theta]$ is the same for all $\theta \in \Theta$ and thus the propagated parameter uncertainty $\Delta_m(g(\mathbf{y}))$ converges to zero.

We first consider a special case where the transform g is an identify map i.e. $g(\mathbf{y}) = \mathbf{y}$. Assuming the likelihood is modelled by a Gaussian distribution with heteroscedastic noise i.e. $p(\mathbf{y}|\theta_1, \theta_2, \mathbf{x}, \mathcal{D}) = \mathcal{N}(\mathbf{y}; \mu(\mathbf{x}; \theta_1), \Sigma(\mathbf{x}; \theta_2))$, then we can show that the parameter and intrinsic uncertainty are given by

$$\Delta_m(\mathbf{y}) = \mathbb{V}_{p(\theta_1|\mathcal{D})}[\mu_{\theta_1}(\mathbf{x})], \quad \Delta_i(\mathbf{y}) = \mathbb{E}_{p(\theta_2|\mathcal{D})}[\Sigma_{\theta_2}(\mathbf{x})] \tag{3.17}$$

which can be approximated by the components of the MC variance estimator in eq. (3.12) :

$$\widehat{\Delta}_m(\mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T} \mu(\mathbf{x}; \theta_1^t)\mu(\mathbf{x}; \theta_1^t)^{\mathrm{T}} - \hat{\mu}_{\mathbf{y}|\mathbf{x}}\hat{\mu}_{\mathbf{y}|\mathbf{x}}^{\mathrm{T}} \tag{3.18}$$

$$\widehat{\Delta}_i(\mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T} \Sigma(\mathbf{x}; \theta_2^t) \tag{3.19}$$

where $\{(\theta_1^t, \theta_2^t)\}_{t=1}^T$ are drawn from the approximate posterior $q_\phi(\theta_1, \theta_2)$.

More generally, when the transform g is complicated, MC sampling provides an alternative implementation. Given samples of model parameters $\{\theta_t\}_{t=1}^T \sim q(\theta|\mathcal{D})$ and $\{g_j^t\}_{j=1}^J \sim p(g(\mathbf{y})|\theta_t, \mathbf{x}, \mathcal{D})$ for $t = 1, ..., T$, we estimate both the progapated parameter and intrinsic uncertainty as follows:

$$\widehat{\Delta}_m(g(\mathbf{y})) \triangleq \frac{1}{T}\sum_{t}(\hat{\mu}^t)^2 - \Big(\frac{1}{(J-1)T}\sum_{j,t}(g_j^t)\Big)^2 \tag{3.20}$$

$$\widehat{\Delta}_i(g(\mathbf{y})) \triangleq \frac{1}{(J-1)T}\sum_{j,t}(g_j^t)^2 - \frac{1}{T}\sum_{t}(\hat{\mu}^t)^2 \tag{3.21}$$

$$\hat{\mu}^t = \frac{1}{J}\sum_{j} g_j^t. \tag{3.22}$$

These estimators are, although unbiased, higher in variance than the case where g is the identity (eq. (3.18) and eq. (3.19)), due to two sources of sampling, thus requiring more samples for reliable estimation of respective uncertainty components.

3.4 Experiments and Results

In this section, we evaluate the proposed uncertainty modelling techniques for super-resolution of diffusion MR images. We first compare quantitatively the reconstruction performance of our probabilistic CNN models against the relevant baselines in two different types of diffusion signal representations. Secondly, we study the real-world utility of the technique in downstream tractography applications. Thirdly, we evaluate the value of predictive uncertainty as a realiability metric of output images on multiple datasets of both healthy subjects and those with unseen pathological structures such as brain tumour (Glioma) and multiple sclerosis (MS).

3.4.1 Datasets

We make use of the following four diffusion MRI datasets to evaluate different benefits of the proposed technique:

- **Human Connectome Project dataset:** we use the diffusion MRI data from the WU-Minn HCP (release Q3) [100] as the source of the training datasets. The dataset enjoys very high image resolution, signal levels and coverage of the measurement space,

enabled by the combination of custom imaging, reconstruction innovations and a lengthy acquisition protocol [10]. Each subject's data set contains 288 diffusion weighted images (DWIs) of voxel size 1.25^3 mm^3 of which 18 have nominal $b = 0$ and the three high-angular-resolution-diffusion-imaging (HARDI) shells of 90 directions have nominal b-values of $1000, 2000$, and 3000 smm^{-2} (see [10] for the full acquisition details). The data are preprocessed by correcting distortions including susceptibility-induced, eddy currents and motion as outlined in [101].

- **Lifespan dataset:** this dataset (available online at `http://lifespan.humanconnectome.org` contains 26 subjects of much wider age range ($8 - 75$ years) than the main HCP cohorts ($22 - 36$ years), and is acquired with a shortened version of the main HCP protocol with lower resolution (1.5 mm isotropic voxels) and only two HARDI shells, with $b = 1000$ and 2500 smm^{-2}. However, we also note that the protocol still leverages the special features of the HCP scanners, providing images of substantially better quality than standard sequences. We utilise this out-of-training-distribution dataset to assess the robustness of our techniques to domain shifts.

- **Prisma dataset:** two healthy male adults (29 and 33 years old respectively) were scanned twice at different image resolutions using the clinical 3T Siemens Prisma scanner in FMRIB, Oxford. Both datasets contain diffusion MRI data with 21 $b = 0$ images and three 90-direction HARDI shells, b-values of 1000, 2000, and 3000 smm^{-2}, each for two resolutions, 2.50 mm and 1.35 mm isotropic voxels (see [54] for full acquisition details). In addition, each of these datasets also includes a standard 3D T1-weighted MPRAGE (1 mm isotropic resolution). The Prisma scanner is less powerful than the bespoke HCP scanner and cannot achieve sufficient signal at 1.25 mm resolution, but the 1.35 mm data provides a pseudo ground-truth for IQT resolution enhancement of the 2.5 mm data.

- **Pathology dataset:** we use two separate datasets which consist of images of brain tumour (Glioma) [102] and multiple sclerosis (MS) patients, respectively. The data of each wubject with glioma contains DWIs with $b = 700$ s/mm^2 while the measurement of each MS patient is of $b = 1200$ s/mm^2. Both datasets have isotropic voxel size 2^3 mm^3, which is closer to the image resolution of commonplace clinical scanners. We use these datasets to assess the behaviour of predictive uncertainty on images with pathological features that are not represented in the training data set.

In all the experiments, super-resolution are performed on diffusion parameter maps derived from the DWIs in the above datasets. In particular, we consider two diffusion MRI models, namely the diffusion tensor (DT) model [103] and Mean Apparent Propagator (MAP) MRI [104], where the former is the simplest and most standard diffusion parameter map, and the latter is a high-order generalisation of the former with the capacity to characterise signals from more complex tissue structures (e.g. fibre crossing regions), a requirement for successful tractography applications. We compute both of these diffusion parameter maps using the implementation from [54], which is available at `https://github.com/ucl-mig/iqt`.

We fit the DT model to the combination of $b = 0$ images and $b = 1000$ s/mm^2 HARDI shell for the HCP and Lifespan datasets, and $b = 700$ s/mm^2 shell for the brain tumour dataset. In all cases, weighted linear least squares are employed for the fitting, taking into account the spatially varying b-values and gradient directions in the HCP dataset. On the other hand, in the case of MAP-MRI, 22 coefficients of basis functions up to order 4 are estimated via (unweighted) least squares to all three shells of the HCP, Lifespan and Prisma datasets. As noted in [54], the choice of scale parameters (see [104]) $\mu_x = \mu_y = \mu_z = 1.2 \times 10^{-3}$ mm empirically minimises the fitting error in the HCP dataset, and is used for all datasets.

Training datasets in all experiments are constructed by artificially downsampling very high-resolution images in the HCP dataset. In particular, we employ the following downsampling

Table 3.1: Details of training data for two diffusion MR signal representations, DTIs and MAP-MRIs. The first two columns from the right denote the size of the input $\mathbf{x}$ and output patches $\mathbf{y}$ of dimension [width, height, depth, channels] while the third and the fourth columns show the number of patch pairs $(\mathbf{x}, \mathbf{y})$ extracted from each subject, and the total number of training subjects used, respectively.

Data	Size of input $\mathbf{x}$	Size of output $\mathbf{y}$	No. pairs $(\mathbf{x}, \mathbf{y})$ per subject	No. subjects
DTIs	$11 \times 11 \times 11 \times 6$	$14 \times 14 \times 14 \times 6$	8000	16
MAP-MRIs	$21 \times 21 \times 21 \times 22$	$14 \times 14 \times 14 \times 22$	4000	16

procedure: (i) the raw DWIs of selected subjects are blurred by applying the mean filter of size $r \times r \times r$ independently over channels with r denoting the upsampling rate; (ii) the DT or MAP parameters are computed for every voxel; (iii) the spatial resolution of the resultant parameter maps are reduced by taking every r pixels. A coupled library of low-resolution and high-resolution patches is then constructed by associating each patch in the downsampled DTI/MAP-MRI with the corresponding patch in the ground truth DTI or MAP-MRI. In this case, we ensure the low-resolution patch to be centrally and entirely contained within the corresponding high-resolution patch (as illustrated by the yellow and orange squares in Fig. 3.3). We then randomly select a pre-set number of patches from each subject in the training pool to create a training dataset as detailed in Table 3.1. In addition to the 8 subjects used in the prior work [53, 1, 48], we randomly select additional 8 subjects from the HCP cohort and include them in the training subject pool. Patches are standardized channel-wise by subtracting the mean of foreground pixel intensities of the corresponding subject and dividing by its standard deviation. Moreover, since MAP-MRI datasets contain outliers due to model fitting, in large enough quantity to influence the training of the baseline 3D-ESPCN model, we remove them by clipping the voxel intensity values of the respective 22 channels separately at 0.1% and 99.9% percentiles computed over all the foreground voxels in the whole training dataset.

3.4.2 Network Architectures and Training

For the training of all CNN models, we minimised the associated loss function using Adam [105] for 200 epochs with initial learning rate of 10^{-3} and $\beta = [0.9, 0.999]$, with minibatches of size 12. We hold out 50% of training patch pairs as a validation set. The best performing model was selected based on the mean-squared-error (MSE) on the validation set.

For the super-resolution of DTIs, as in [84], we use a minimal architecture for the baseline 3D-ESPCN, consisting of three $3D$ convolutional layers with filters $(3^3, 50) \rightarrow (1^3, 100) \rightarrow (3^3, 6r^3)$ where r is upsampling rate and 6 is the number of channels in DTIs. As illustrated in Fig. 3.3, the dimensions of convolution filters are chosen, so each $5^3 \cdot 6$ low-resolution receptive field patch maps to a $r^3 \cdot 6$ high-resolution patch, which mirrors competing random forest based methods [53, 1] for a fair comparison. On the other hand, for MAP-MRI, which is a more complex image modality with 21 channels, we employ a deeper model with 6 convolution layers $(5^3, 256) \rightarrow (3^3, 256) \rightarrow (3^3, 128) \rightarrow (3^3, 128) \rightarrow (3^3, 64) \rightarrow (3^3, 21r^3)$ prior to the shuffling operation, which expands the receptive field on each $r^3 \cdot 21$ high-resolution patch to $15^3 \cdot 21$ input low-resolution patch. Every convolution layer is followed by a ReLU non-linearity except the last one in the architecture, and batch-normalization [106] is additionally employed for MAP-MRI super-resolution between convolution layer and ReLU non-linearity.

The mean and variance networks in the heteroscedastic noise model introduced in Sec. 3.3.3 are implemented as two separate baseline 3D-ESPCNs of the architectures, specified above for DTIs and MAP-MRIs. Positivity of the variance is enforced by passing the output through a

softplus function $f(x) = \ln(1 + e^x)$ as in [68].

For variational dropout, we considered two flavours: Var.(I) optimises per-weight dropout rates, and Var.(II) optimises per-filter dropout rates. More formally, the "drop-out rate" α_{ij} in the approximate posterior $q_\phi(\theta_{ij}) = \mathcal{N}(\theta_{ij}; \eta_{ij}, \alpha_{ij}\eta_{ij}^2)$ is different for every element in each convolution kernel in the former while the latter has common α_{ij} shared across each kernel. In preliminary analysis, we found that the number of samples per data point for estimating reconstruction term (eq. 3.7) can be set to $S = 1$ so long as the batch size is sensibly large ($M = 12$).

We also note the default training with binary and Gaussian dropout also employs $S = 1$ [91] along with other MC variational inference methods for neural networks such as [97, 52, 107]. Variational dropout is applied to both the baseline and heteroscedastic models without changing the architectures. For both binary and Gaussian dropout modes, we incorporate the dropout operations of fixed rate p in every convolution layer of the baseline 3D-ESPCN architecture.

All models are trained on simulated datasets generated from 16 HCP subjects as detailed in Sec. 3.4.1. We also retrained the random forest models employed in [1, 54] on equivalent datasets. It takes under 60/360 mins to train a single network on DTI/MAP-MRI data on a single TITAN X GPU. All models are implemented in the TensorFlow framework [108] and the codes will be released at `https://github.com/rtanno21609/UncertaintyNeuroimageEnhancement` upon publication.

3.4.3 Quantitative Evaluation of Super-resolution Performance

We evaluate the prediction performance of our models for super-resolution of DTI and MAP-MRI on two datasets—HCP and Lifespan as detailed in Sec. 3.4.1. The first dataset contains 16 unseen subjects from the same HCP cohort used for training, while the second one consists of 10 subjects from the HCP Lifespan dataset. The latter tests generalisability, as they are acquired with a different protocol at lower resolution (1.5 mm isotropic), and contain subjects of a different age range (45-75 years) to the original HCP data (22-36 years). We perform $\times 2$ upsampling in all spatial directions. The reconstruction quality is measured with root-mean-squared-error (RMSE), peak-signal-to-noise-ratio (PSNR) and mean-structural-similarity (MSSIM) [15] on two separate regions: i) "interior"; set of patches contained entirely within the brain mask; ii) "exterior"; set of patches containing some brain and some background voxels, as shown in Fig. 3.6. This is because the current state-of-the-art methods based on random forests (RFs) such IQT-RF [54] and BIQT-RF [1] are only trained on patches from the interior region and requires a separate procedure on the brain boundary. In addition, the estimation problem is quite different in boundary regions, but remains valuable particularly for applications such as tractography where seed or target regions are often in the cortical surface of the brain. We only present the RMSE results, but the derived conclusions remain the same for the other two metrics. Aside from the interpolation techniques, for each method an ensemble of 10 models are trained on different trainings set (generated by randomly extracting patch pairs from the common 16 HCP training subjects) and for each model, the average error metric over the test subjects are first calculated. The mean and standard deviations of such average errors are computed across the model ensemble and reported in Table 3.2 and Table 3.3.

Table 3.2 shows that our baseline achieves 8.5%/39.8% reduction in RMSE for the super-resolution of DTIs on the HCP dataset on the interior/exterior regions with respect to the best published method, BIQT-RF[1]. While the standard deviations are higher, the improvements are more pronounced in MAP-MRI super-resolution, reducing the average RMSEs by 49.6% and 63.5% on the interior and exterior regions. We note that that IQT-RF and BIQT-RF are only trained on interior patches, and super-resolution on boundary patches requires a separate *ad hoc* procedure. Despite including exterior patches in training our model, which complicates the learning task, the baseline CNN out-performs the RF methods on both regions. We see

Figure 3.6: Visualisation of "interior" (yellow) and "exterior" regions (red). The interior region consists of a set of patches contained entirely within the brain while the exterior region consists of partial patches that contain mixtures of brain and background voxels

similar improvements in the out-of-distribution Lifespan dataset.

Reconstruction is faster than the RF baselines; the 3D-ESPCN is capable of estimating the whole high-resolution DTI/MAP-MRI under 10/60 seconds on a CPU and 1/10 second(s) on a GPU. On the other hand, BIQT-RF takes $\sim$ 10 mins with 8 trees on both DTIs and MAP-MRIs. The fully convolutional architecture of the model enables to process input patches of different size from that of training inputs, and we achieve faster reconstruction by using larger input patches of dimension $25^3 \cdot c$ where c is the number of channels. We also note that the reconstruction time of the variational dropout based models increases by a factor of the number of MC samples used at test time, although it is possible, with more memory, to leverage GPU parallelisation by making multiple copies of each input patch and treating them as a mini-batch. On the other hand, the heteroscedastic CNN enjoys the same inference speed of the baseline since only the mean network is used for reconstruction (the covariance network is only employed to quantify the estimated intrinsic uncertainty).

Table 3.2 shows that, on both HCP and Lifespan data, modelling both intrinsic and parameter uncertainty (i.e. Hetero. + Variational Dropout (I), (II)) achieves the best reconstruction accuracy in DTI super-resolution. We observe that modelling intrinsic uncertainty with the heteroscedastic network on its own further reduces the average RMSE of the baseline 3D-ESPCN on the interior region with high statistical significance ($p < 10^{-3}$). However, poorer performance is observed on the exterior than the baseline. On the other hand, using 200 MC weight samples, we see modelling parameter uncertainty with variational dropout (see Variational Dropout.(I)-CNN) performs best on both datasets on the exterior region. Combination of heteroscedastic model and variational dropout (i.e. Hetero. + Variational Dropout (I) or (II)) leads to the top 2 performance on both datasets on the interior region and reduces errors on the exterior to the level comparable or better than the baseline.

Similarly, Table 3.3 shows that the best performance in MAP-MRI super-resolution comes from the combined models (i.e. Hetero.+Variational Dropout.(I) and (II)). We observe that as with the DTI case, modelling intrinsic uncertainty through the heteroscedastic network improves the reconstruction accuracy on the interior region, whilst the errors on the exterior are increased with respect to the baseline 3D-ESPCN. Moreover, the improvement is pronounced when the outliers due to model fitting errors are not removed in the training data. In this case, we see that the reconstruction accuracy of 3D-ESPCN dramatrically decreases, whilst in contrast it is only marginally compromised when equipped with the heteroscedastic noise model, displaying robustness to outliers. Lastly, we note that the top-2 accuracy are consistently achieved by the joint modelling of intrinsic and parameter uncertainty (i.e. Hetero.+Variational Dropout.(I)

Models	HCP (interior)	HCP (exterior)	Life (interior)	Life (exterior)
CSpline-interpolation	10.069± n/a	31.738± n/a	32.483± n/a	49.066± n/a
β-Spline interpolation	9.578± n/a	98.169± n/a	33.429± n/a	186.049± n/a
IQT-RF	6.974 ± 0.024	23.139 ± 0.351	10.038 ± 0.019	25.166 ± 0.328
BIQT-RF	6.972 ± 0.069	23.110 ± 0.362	9.926 ± 0.055	25.208 ± 0.290
3D-ESPCN(baseline)	6.212 ± 0.017	13.609 ± 0.084	8.902 ± 0.020	16.389 ± 0.114
+ Binary Dropout ($p = 0.1$)	6.319 ± 0.015	13.738 ± 0.048	9.093 ± 0.024	16.489 ± 0.099
+ Gaussian Dropout ($p = 0.05$)	6.463 ± 0.034	14.168 ± 0.051	9.184 ± 0.048	16.653 ± 0.092
+ Variational Dropout (I)	6.194 ± 0.013	$\mathbf{13.412 \pm 0.041}$	8.874 ± 0.027	$\mathbf{16.147 \pm 0.051}$
+ Variational Dropout (II)	6.201 ± 0.015	13.479 ± 0.047	8.878 ± 0.031	16.230 ± 0.075
+ Hetero.	6.135 ± 0.029	15.469 ± 0.231	8.885 ± 0.041	17.208 ± 0.211
+ Hetero. + Variational Dropout (I)	6.121 ± 0.015	13.591 ± 0.051	$\mathbf{8.837 \pm 0.043}$	16.261 ± 0.053
+ Hetero. + Variational Dropout (II)	$\mathbf{6.116 \pm 0.013}$	13.622 ± 0.099	8.861 ± 0.031	16.387 ± 0.098

Table 3.2: Super-resolution results on diffusion tensor images (DTIs) of HCP and Lifespan datasets for different upsampling methods. For each method, an ensemble of 10 models are trained on different training sets generated by randomly extracting a set of patch pairs from the common 16 HCP subjects. For each model, the average RMSE ($\times 10^{-4}$mm^2/s) over subjects in respective datasets is first computed and the mean/std of such average RMSE over the ensemble are then reported. Best results in bold red, and the second best in blue.

Models	HCP (interior)	HCP (exterior)	Life (interior)	Life (exterior)
CSpline interpolation	5.234± n/a	30.362± n/a	7.135± n/a	29.232± n/a
β-Spline interpolation	4.852± n/a	63.446± n/a	6.523± n/a	56.937± n/a
IQT-RF [54]	4.538 ± 0.113	25.541 ± 0.131	5.882 ± 0.121	26.137 ± 0.279
BIQT-RF [1]	4.838 ± 0.129	25.523 ± 0.175	5.949 ± 0.131	27.509 ± 0.233
3D-ESPCN(baseline)	2.285 ± 0.126	9.316 ± 0.127	4.195 ± 0.163	11.922 ± 0.192
+ Binary Dropout ($p = 0.1$)	2.283 ± 0.154	9.272 ± 0.132	4.120 ± 0.178	11.652 ± 0.204
+ Gaussian Dropout ($p = 0.1$)	2.370 ± 0.155	9.335 ± 0.144	4.327 ± 0.157	11.907 ± 0.211
+ Variational Dropout (I)	2.155 ± 0.122	9.205 ± 0.193	3.997 ± 0.153	11.547 ± 0.177
+ Variational Dropout (II)	2.172 ± 0.128	9.112 ± 0.173	3.972 ± 0.132	11.511 ± 0.172
+ Hetero.	1.998 ± 0.132	11.294 ± 0.216	3.872 ± 0.140	12.084 ± 0.129
+ Hetero + Variational Dropout (I)	$\mathbf{1.951 \pm 0.122}$	9.102 ± 0.181	$\mathbf{3.572 \pm 0.171}$	$\mathbf{11.037 \pm 0.192}$
+ Hetero + Variational Dropout (II)	1.969 ± 0.119	$\mathbf{9.052 \pm 0.162}$	3.606 ± 0.141	11.311 ± 0.195
3D-ESPCN(without outlier removal)	3.425 ± 0.163	13.284 ± 0.239	6.032 ± 0.229	15.513 ± 0.273
+ Hetero.	2.264 ± 0.153	11.306 ± 0.172	3.919 ± 0.140	12.821 ± 0.150
+ Hetero + Variational Dropout (I)	2.138 ± 0.159	10.022 ± 0.187	3.681 ± 0.193	12.133 ± 0.205
+ Hetero + Variational Dropout (II)	2.133 ± 0.188	9.988 ± 0.209	3.690 ± 0.184	12.052 ± 0.212

Table 3.3: Super-resolution results on MAP-MRIs of HCP and Lifespan datasets for different upsampling methods. For each method, an ensemble of 5 models are trained on different training sets generated by randomly extracting a set of patch pairs from the common 16 HCP subjects. For each model, the average RMSE over subjects in respective datasets is first computed and the mean/std of such average RMSEs over the ensemble are then reported. Best results in bold red, and the second best in blue. In addition, the performance of 3D-ESPCN and its probabilistic variants trained on data without outlier removal are also included.

and (II)) on both the interior and exterior regions on both HCP and Lifespan datasets.

The performance difference of heteroscedastic network between the interior and the exterior region roots from the loss function. The Mahalanobis term $\mathcal{M}_\theta(\mathcal{D})$ in eq.(3.5) imposes a larger penalty on the regions with smaller intrinsic uncertainty. The network therefore allocates less of its resources towards the regions with higher uncertainty (e.g. boundary regions) where the statistical mapping from the low-resolution to high-resolution space is more ambiguous, and biases the model to fit the regions with lower uncertainty. However, we note that the performance of the heteroscedastic network is still considerably better than the standard interpolation and RF-based methods. By augmenting the model with variational dropout, the exterior error of the heteroscedastic model is dramatically reduced, indicating its regularisation effect against overfitting to low-uncertainty areas. We also observe concomitant performance improvement on the interior regions on both datasets, which additionally shows the benefits of such regularisation even in low-uncertainty areas.

Both Table 3.2 and Table 3.3 show that the use of variational dropout attains lower errors than the models with fixed dropout probabilities p, namely, Binary and Gaussian dropout [91]. Different instances of both dropout models are trained for a range of p by linearly increasing on the interval $[0.05, 0.3]$ with increment 0.05, and the test errors for the configurations with smallest RMSE on the validation set are reported in Table 3.2 and Table 3.3. As with variational dropout models, 200 MC samples are used for inference. In all cases, two variants of variational dropout (I) and (II) outperform the networks with the best binary or Gaussian dropout models, showing the benefits of learning dropout probabilities p rather than fixing them in advance.

3.4.4 Tractography with MAP-MRI

Reconstruction accuracy does not necessarily reflect real world utility. We thus further assessed the benefits of super-resolution with a tractography experiment on the Prisma dataset, which contains two DWIs of the same subject at two different image resolutions—1.35 mm and 2.5 mm isotropic voxels, as detailed in Sec. 3.4.1. An ensemble of 10 best performing CNN (3D-ESPCN+Hetero.+Variational Dropout(I)) is used to super-resolve the MAP-MRI coefficients [104] derived from the low-resolution DWIs, and the ensemble predictions aggregated into the final output by taking the average estimate weighted by the inverse of the estimated intrinsic uncertainty. Lastly, the high-resolution multi-shell DWIs are obtained from this super-resolved MAP volume. Specifically, the Spherical Mean Technique (SMT) is used to fit a microscopic tensor model to the predicted dataset [109]. The voxel-by-voxel estimated model parameters inform the spatially varying fibre response function that is used to recover the fibre orientation distribution through spherical deconvolution. Afterwards, we perform probabilistic tractography [110] with the fibre pathways randomly seeded in the brain. In a similiar fashion, we also generate high-resolution datasets by using IQT-RF and linear interpolation.

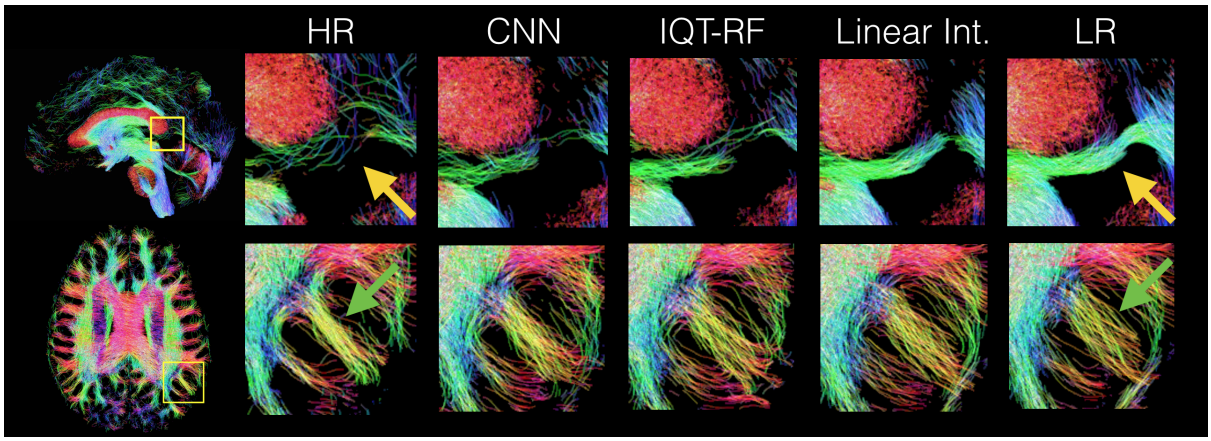


Figure 3.7: Streamline maps of the probabilistic tractography [110] applied on Prisma dataset for different upsampling methods and visualised with MRtrix3 [111]. From left to right: (i) High-res acquisition, (ii) CNN (3D-ESPCN+Hetero.+Variational Dropout(I)) prediction; (iii) Random Forest IQT [54]; (iv) Linear interpolation; (v) Low-res acquisition. The yellow arrows in the top row indicate the location of a false positive tract detected in the low-resolution acquisition, whilst the green arrows in the bottom row show an example white matter tract, which is more sharply reconstructed at high resolution.

Fig. 3.7 shows that IQT via our best performing CNN makes a tangible difference in downstream tractography. In the top row, tractography on the low-resolution data produces a false-positive tract under the corpus callosum (yellow arrow), which tractography at high resolution avoids. Reconstructured high-resolution images from IQT-RF and CNN predictions avoid the false positive better than linear interpolation. Note that we do not expect to reproduce the high-resolution tractography map exactly, as the high-resolution and low-resolution images are not aligned exactly and the high-resolution and prediction have different resolutions (1.35 mm vs. 1.25 mm). The bottom row shows sharper recovery of small gyral white matter pathways (green arrow) at high-resolution than low-resolution resulting from reduced partial volume effect. CNN reconstruction produces a sharper pathway than RF-IQT and linear interpolation, more closely reflecting the high-resolution tractography.

3.4.5 Uncertainty Quantification

In this section, we investigate the value of uncertainty modelling in enhancing the safety of super-resolution system beyond reduced reconstruction errors. Firstly, in Sec. 3.4.5.1, we study the utility of predictive uncertainty map as a proxy measure of reconstruction accuracy on healthy test subjects from both HCP and Lifespan datasets. Secondly, in Sec. 3.4.5.2, we look into the behaviour of uncertainty maps in the presence of abnormal features that are not present in the training data.

3.4.5.1 Healthy Test Subjects

We employ the most performant CNN model (3D-ESPCN + Hetero. + Variational Dropout(I)) to generate the high-resolution predictions of *mean diffusivity* (MD) and *fractional anisotropy* (FA), and their associated predictive uncertainty maps. Here we draw 200 samples of high-resolution DTI predictions for each subject from the predictive distribution $q^*_\phi(\mathbf{y}|\mathbf{x})$, and then the FA and MD maps of each prediction are computed. The sample mean and standard deviation are then calculated from these samples to generate the final estimates of high-resolution MD/FA maps and their corresponding predictive uncertainty.

Fig. 3.8 displays high correspondence between the error (RMSE) maps and the predictive uncertainty on both FA and MD of a HCP test subject. This demonstrates the potential utility of uncertainty map as a surrogate measure of prediction accuracy. In particular, the MD uncertainty map captures subtle variations within the white matter and the cerebrospinal fluid (CSF) at the centre. Also, in accordance with the low reconstruction accuracy, high predictive uncertainty is observed in the CSF in MD. This is expected since the CSF is essentially free water with low signal-to-noise-ratio (SNR) and is also affected by biological noise such as cardiac pulsations. The reconstruction errors are high in FA prediction on the bottom-right quarter of the brain boundary, close to the skull, which is also reflected in the uncertainty map.

Fig. 3.9 tests the utility of predictive uncertainty map in discriminating potential predictive failures in the predicted high-resolution MD map. We define ground truth "safe" voxels as the ones with reconstruction error (RMSE) smaller than a fixed value, and the task is to separate them from the remaining ground-truth "risky" voxels by thresholding on their predictive uncertainty values. The threhold for defining safe voxels is set to 1.5×10^{-4} s/mm^2, such that the risky voxels mostly concentrate on the outer-boundary and the CSF regions (which account for 17.5% of all voxels under consideration). Here the positive class is defined as "safe" while the negative class is defined as "risky". Fig. 3.9 (a) shows the corresponding receiver operating characteristic (ROC) curve of such binary classification task, which plots the true-positive-rate (TPR) against the false-positive-rate (FSR) computed based on all the voxels in the 16 HCP training subjects. In this case, TPR decribes the percentage of correctly detected safe voxels out of all the safe ones, while FPR is defined as the percentage of risky voxels that are wrongly

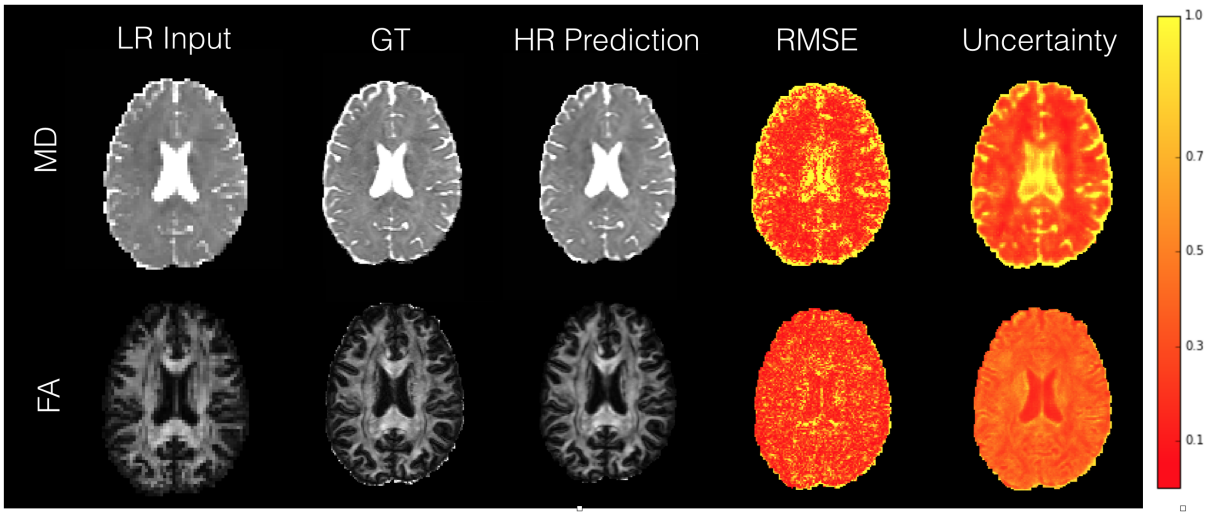


Figure 3.8: Comparison between voxel-wise RMSE and predictive uncertainty maps for FA and MD computed on a HCP test subject (min-max normalised for MD and FA separately). Low-res input, ground truth and the mean of high-resolution predictions are also shown.

classified as safe out of all the risky voxels. We then select the best threshold by maximising the F1 score, and use this to classify the voxels in each predicted high-resolution MD into "safe" and "risky" ones for all subjects in the test HCP dataset and the Lifespan dataset. Fig.3.9 (b) shows the inter-subject average of the TPR and FPR on both datasets. While on average TPR slightly worsens compared to the results on the training subjects, FPR improves in both cases—notably, this uncertainty-based classification is able to correctly identify 96% of risky predictions on unseen subjects from out-of-training-distribution dataset, namely Lifespan, which differs in demographics and underlying acquisition. Fig.3.9 (c) visualises the classification results to the pre-defined "ground truth" on one of the Lifespan subjects, which illustrates that the generated "warning" aggressively flags potentially risky voxels at the cost of thresholding out the safe ones.

3.4.5.2 Unseen Abnormalities and Uncertainty Decomposition

We separately visualise the propagated intrinsic and parameter uncertainty over the predicted high-resolution MD map on images of subjects with a variety of different unseen abnormal structures, such as benign cysts, tumours (Glioma) and focal lesions caused by multiple sclerosis (MS). We emphasise here that the all these images have been acquired with different protocols. Specifically, benign cysts in the HCP datasets represent abnormalities in images acquired with the same protocol as the training data, while tumours and MS lesions are examples of pathologies present in out-of-distribution imaging protocols. In all cases, we use the SR network, Hetero.+Variational Dropout (I), trained on healthy subjects from HCP dataset. For each of 200 different sets of parameters $\{\theta_t\}_{t=1}^{200}$ sampled from the posterior distribution $q(\theta|\mathcal{D})$, we draw 10 samples of high-resolution DTIs from the likelihood, $\{\mathbf{y}_j^t\}_{j=1}^{10} \sim p(\mathbf{y}|\theta_t, \mathbf{x}, \mathcal{D})$, compute the corresponding MD, and approximate the two constituents of predictive uncertainty with the MC estimators given in eq.(3.20) and (3.21).

Fig. 3.10 shows the reconstruction accuracy along with the components of predictive uncertainty over the high-resolution MD map of a HCP test subject, which contains a benign abnormality (a small posterior midline arachnoid cyst). The error (RMSE) and propagated intrinsic uncertainty are plotted on the same scale whereas the propagated model uncertainty is plotted on 1/5 of the scale for clear visualisation. In this case, the predictive uncertainty

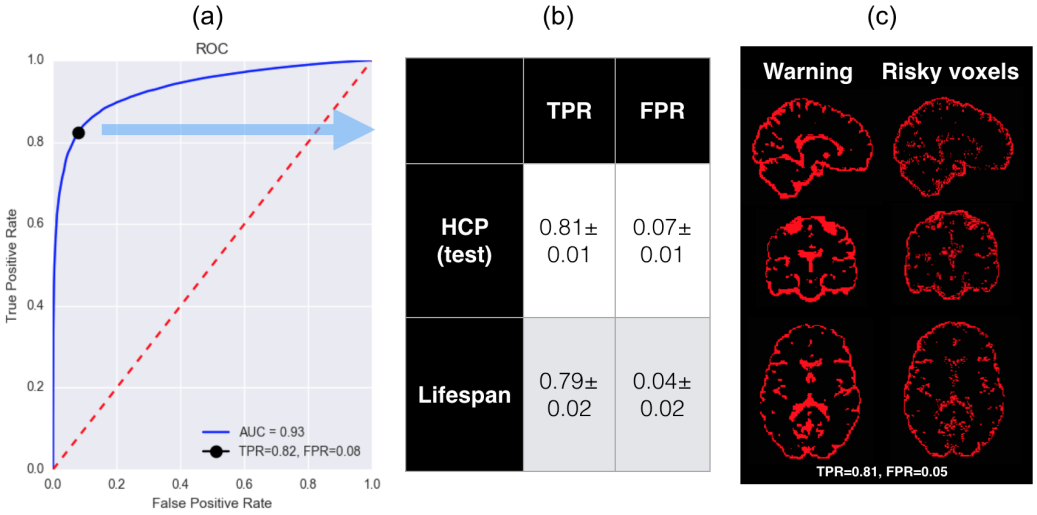


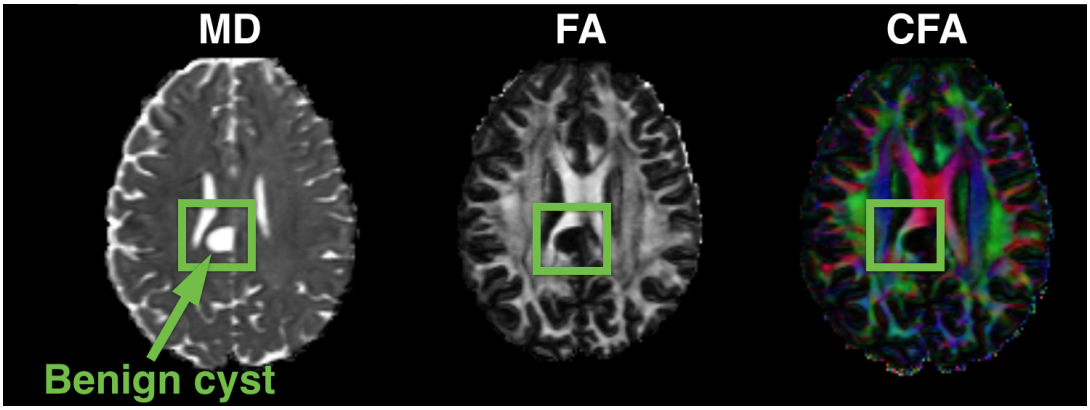
Figure 3.9: Discrimination of "safe" voxels in the predicted high-resolution MD map by thresholding on predictive uncertainty. Here a single 3D-ESPCN + Hetro. + Variational Dropout (I) model is used to quantify the predictive uncertainty over each image volume. (a) the ROC curve plots the true positive rate (TPR) against false positive rate (FPR) computed for a range of threshold values on the foreground voxels in the training subjects. Best threshold (black dot) was selected such that F1 score is maximised and is employed to separate "safe" voxels from "risky" ones; (b) the average TPR and FPR over the 16 test HCP subjects and the 16 Lifespan subjects are shown; (c) an example visualisation of the "ground truth" safe (black) and risky (red) voxels on a Lifespan subject along with the corresponding classification results denoted as "warning".

is dominated by the intrinsic component. In particular, low propagated intrinsic uncertainty is observed in the interior of the cyst relative to its boundary in accordance with the high accuracy in the region. This is expected as the interior structure of a cyst is highly homogeneous with low variance in signals and the super-resolution task should therefore be relatively straightforward. On the other hand, the component of parameter uncertainty is high on the interior structure which also makes sense as such homogeneous features are underrepresented in the training data of healthy subjects. This example illustrates how decoupling the effects of intrinsic and parameter uncertainty potentially allows one to make sense of the predictive performance.

Fig.3.11 visualises the uncertainty components generated by the same CNN model trained on datasets of varying size. We see that the propagated parameter uncertainty diminishes as the training set size increases, while the propagated intrinsic uncertainty stays more or less constant. This result is indeed what is expected as described in Fig. 3.1; the specification of network weights becomes more confident i.e. the variance of the posterior distribution decreases as the amount of training data increases, while the effect of intrinsic uncertainty is irreducible with the amount of data. On the other hand, when the standard binary or Gaussian dropout was employed instead of variational dropout, we observed that the effect of parameter uncertainty stayed more or less constant with the size of training data. This may be a consequence of the posterior variance, largely determined by the prespecified drop-out rates, which in turn results in more static variance of predictive distribution.

We further validate our method on clinical images with previously unseen pathologies. We note that the pathology data contain images acquired with standard clinical protocols with voxel size slightly smaller than that of the training low-resolution images and lower signal-to-noise ratio.

(a) HCP subject with a benign cyst



(b) Errors vs. uncertainty components

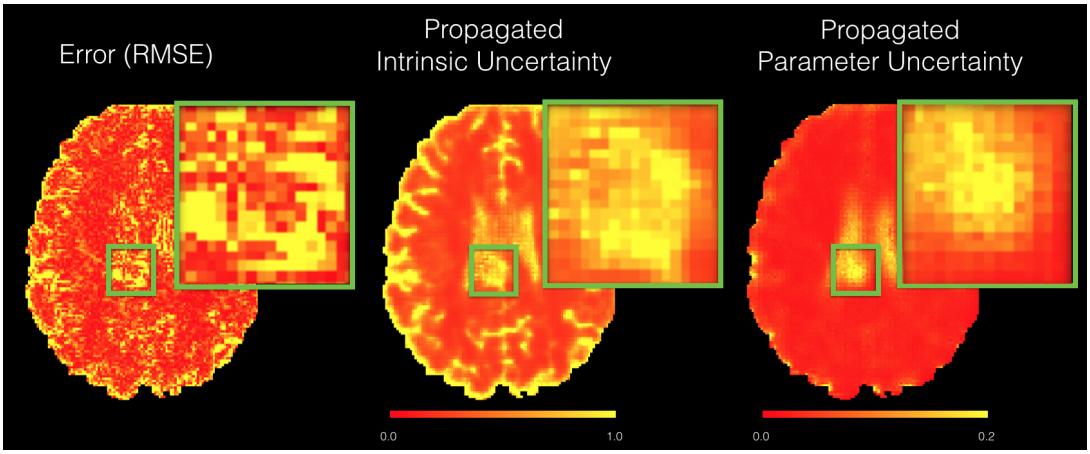


Figure 3.10: Visualisation of (a) MD, FA and colour FA maps computed from the DTI of a HCP subject with a small posterior midline arachnoid cyst in the central part of the brain. (b) the corresponding reconstruction accuracy (RMSE) in MD and the corresponding components of predicted uncertainty.

Fig. 3.12 shows that pathological areas not represented in the training set are flagged as highly uncertain. Although the ground truth is not available in this case, the uncertainty can be quantified instead to flag potential low accuracy areas. Fig. 3.12 (a) shows that the propagated parameter uncertainty highlights the tumour core, and speckly artefacts in the input image, which are not represented in the training data. On the other hand, the intrinsic uncertainty component is high on the whole region of pathology covering both the tumour core and its surrounding edema. Fig. 3.12 (b) shows that high parameter uncertainty is assigned to a large part of focal lesions in MS, while the intrinsic uncertainty is mostly prevalent around the boundaries between anatomical structures and CSF. We also observe that the super-resolution sharpens the original image without introducing noticeable artifacts; in particular, for the brain tumour image, some of the partial volume effects are cleared.

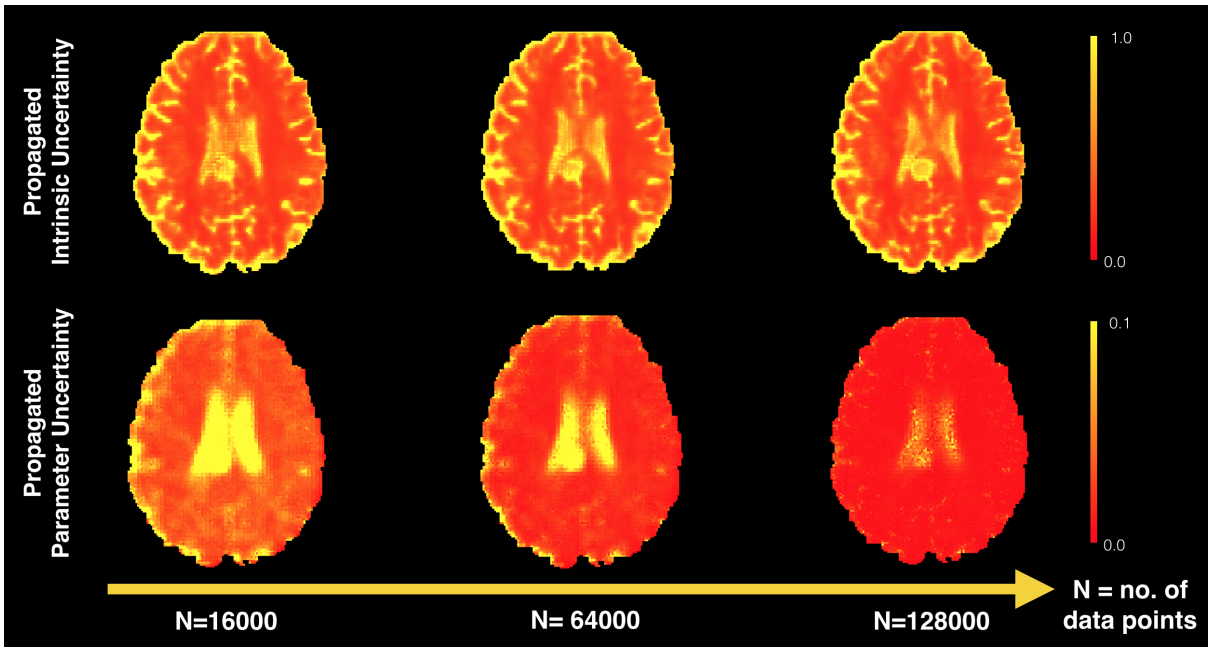
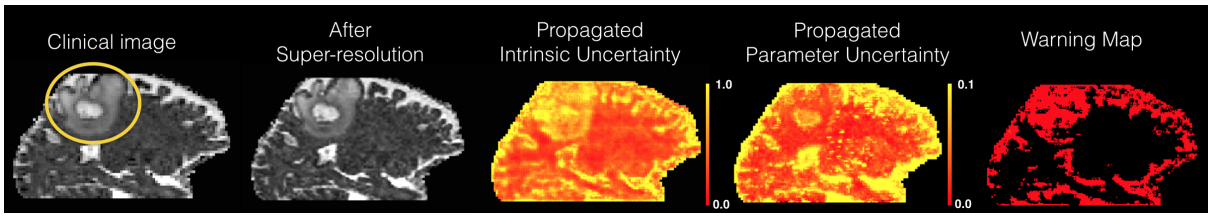


Figure 3.11: Training set size vs propagated intrinsic/parameter uncertainty on the MD map of an unseen HCP subject with a benign cyst. The uncertainty maps are normalised across all the figures.

(a) Brain Tumour



(b) Multiple Sclerosis

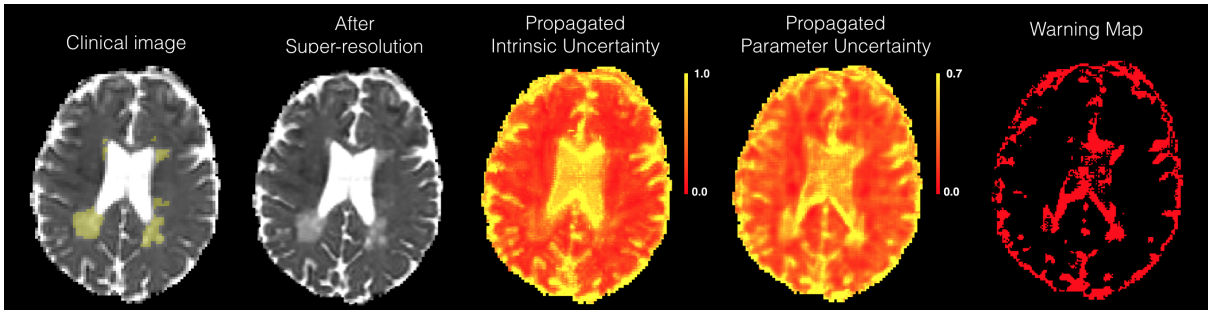


Figure 3.12: Visualisation of propagated uncertainty components on clinical images with pathology that was not present in the training data. The super-resolution is performed on the clinical images due to low-resolution, and thus the ground truths are not available in both cases. (a) shows the results on the data of a Glioma patient, and the yellow circle indicates the region of tumour. (b) shows the same set of results on a MS patient with labels of focal lesions obtained from a neurologist indicated in yellow. Each row shows from left to right: (i) MD map computed from the original DTI; (ii) MD map computed from the output of super-resolution; (iii), (iv) maps of the estimated propagated intrinsic and parameter uncertainty; (v) "warning map" obtained from the same threshold value used in Sec. 3.4.5.1, which flag large parts of the pathological features in both cases.

3.5 Discussion and Conclusion

We introduce a probabilistic deep learning (DL) framework for quantifying three types of uncertainties that arise in data-enhancement applications, and demonstrate its potential benefits in improving the safety of such systems towards practical deployment. The framework models *intrinsic uncertainty* through heteroscedastic noise model and *parameter uncertainty* through approximate Bayesian inference in the form of variational dropout, and finally integrates the two to quantify *predictive uncertainty* over the system output. Experiments focus on the super-resolution application of image quality transfer (IQT)[54] and study several desirable properties of such framework, which lack in the existing body of data enhancement methods based on deterministic DL models.

Firstly, results on a range of applications and datasets show that modelling uncertainty improves overall prediction performance. Table 3.2 and 3.3 show that modelling the combination of both *intrinsic* and *parameter* uncertainty achieves the state-of-the-art accuracy on super-resolution of DTIs and MAP-MRI coefficients in both of the HCP test dataset and the Lifespan dataset, improving on the present best methods based on random-forests (RF-IQT[54] and RF-BIQT[1]) and interpolation—the standard method to estimate sub-voxel information used in clinical visualisation software. In particular, results on the Lifespan dataset, which differs from the training data in age range and acquisition protocol, indicates the better generalizability of our method. In addition, Fig. 3.7 shows that such combined model also benefits downstream tractography in comparison with the previous methods, illustrating the potential utility of the method for downstream connectivity analysis. Such improvement in the predictive performance arises from the regularisation effects imparted by the modelling of respective uncertainty components. Specifically, modelling intrinsic uncertainty through the heteroscedastic network improves robustness to outliers, while modelling parameter uncertainty via variational dropout defends against overfitting. For example, Table 3.3 shows that the predictive performance of the 3D-ESPCN + Hetero. model is only marginally compromised even when the outliers are not removed from training data, while the baseline 3D-ESPCN results in much poorer performance. This can be ascribed to the ability of the variance network $\Sigma_{\theta_2}(\cdot)$ in the 3D-ESPCN + Hetero. architecture to attenuate the effects of outliers by assigning small weights (i.e. high uncertainty) in the weighted MSE loss function as shown in eq. (3.21). However, this loss attenuation mechanism can also encourage the network to overfit to low-uncertainty regions, potentially focusing less on ambiguous yet important parts of the data—we indeed observe in Table 3.3 that the heteroscedastic network performs considerably worse than the baseline 3D-ESPCN on the exterior regions while the reverse is observed on the interior part. Such overfitting to low-uncertainty interior regions is alleviated by modelling parameter uncertainty with variational dropout [52], as evidenced by the dramatic error reduction in the exterior region on both HCP and Lifespan datasets.

Secondly, experiments on the images of healthy and pathological brains have demonstrated the utility of *predictive uncertainty* as a reliability metric of output images. Fig. 3.8 illustrates the strong correspondence between the maps of predictive uncertainty and the reconstruction quality (voxel-wise RMSE) in the downstream derived quantites such as FA and MD maps. In addition, Fig. 3.12 shows that such uncertainty measure also highlights pathological structures not observed in the training data. We have also tested the utility of predictive uncertainty in discriminating voxels with sufficiently low RMSEs in the predicted high-resolution MD maps. As shown in Fig. 3.9, the optimal threshold selected on the HCP training dataset is capable to detecting over 90% of non-reliable predictions—voxels with RMSE above a certain threshold—not only on the unseen subjects in the same HCP cohort but also on subjects from the out-of-sample Lifespan dataset, that are statistically disparate from the training distribution (e.g. different age range and acquisition protocol). These results combined demonstrate the utility of predictive uncertainty map as a means to quantify output safety, and provides a subject-

specific alternative to standard population-group reliability metrics (e.g. mean reconstruction accuracy in a held-out cohort of subjects). Such conventional group statistics can be misleading in practice; for instance, the information that a super-resolution algorithm is reliable 99% of the time on a dataset of 1000 subjects may not accurately represent the performance on a new unseen individual if the person is not well-represented in the cohort (e.g. pathology, different scanners, etc). In contrast, predictive uncertainty provides a metric of reliability, tailored to each individual at hand.

Thirdly, our preliminary experiments show that decomposition of the effects of intrinsic and parameter uncertainty in the predictive uncertainty provides a layer of explanations into the performance of the considered deep learning methods. Fig. 3.10 shows that the low reconstruction error in the centre of the benign cyst can be explained by the dominant intrinsic uncertainty, which indicates the inherent simplicity of super-resolution task in such homogeneous region, whilst the unfamiliarity of such structure in the healthy training dataset is reflected in the high parameter uncertainty. Assuming that the estimates of decomposed uncertainty components are sufficiently accurate, we could act on them to further improve the overall safety of the system. Imagine a scenario where reconstruction error is consistently high on certain image structures, if the parameter uncertainty is high but intrinsic uncertainty is low, this indicates that collecting more training data would be beneficial. On the other hand, if the parameter uncertainty is low and intrinsic uncertainty is high, this would mean that we need to regard such errors as inevitability, and abstain from predictions to ensure safety or account for them appropriately in subsequent analysis.

The proposed methods for estimating intrinsic and parameter uncertainty, however, make several simplifying assumptions in the forms of likelihood model $p(\mathbf{y}|\theta, \mathbf{x})$ and posterior distributions over network parameters $p(\theta|\mathcal{D})$. Firstly, the likelihood model takes the form of a Gaussian distribution with a diagonal covariance matrix. This means that the likelihood model is not able to capture multi-modality of the predictive distribution i.e. the presence of multiple different solutions. While the full predictive distribution (eq. (3.9)) is not necessarily unimodal in theory due to the integration with the posterior distribution, we observe in practice that the drawn samples are not very diverse. Future work should explore the benefits of employing more complex forms of likelihood functions such as mixture models [112, 79], diversity losses [113, 114, 115] and more powerful density estimators [116, 117, 118, 119, 79]. Also, the diagonality of covariance matrices means that the output pixels are assumed statistically independent given the input. Although the predicted images display high inter-pixel consistency, modelling the correlations between neighbouring pixels [120] may further improve the reconstruction quality. Analogous to the likelihood function, variational dropout [52], which is used in this work, approximates the posteriors $p(\theta|\mathcal{D})$ by Gaussian distributions with diagonal covariance, imposing restrictive assumptions of unimodality and statistical independence between neural network weights. More recent advances in the Bayesian deep learning research [121, 122, 123, 124, 125, 126] could be used to enhance the quality of parameter uncertainty estimation by allowing the model to capture multi-modality and statistical dependencies between parameters.

An important future challenge is the clinical validation of predictive uncertainty as a reliability metric of output images. To this end, we need to design a more clinically meaningful definition of success and failure of the data enhancement algorithm at hand. Despite the high accuracy in distinguishing between predictive failures and successes attained with our method (Fig. 3.9), our definition of reconstruction quality, namely voxel-wise RMSE, does not necessarily represent the real utility of the output image. One possible approach would be to have clinical experts to label the potential failures in the super-resolved images, be it for a targeted application (e.g. diagnosis of some neurological conditions) or for general usage in clinical practice. A more economical alternative, which does not require extra label acquisition, is to define the prediction success in downstream measurements of interest i.e. functions of the output

images $g(\cdot)$, such as morphometric measurements of anatomical or pathological structures (e.g. volumes). The propagation method (eq. (3.13)) introduced in Sec. 3.3.6 can be utilised to quantify uncertainty components in the space of target measurement $g(\cdot)$. Measuring the correlation between such propagated uncertainty estimates and the corresponding errors would be a useful indicator of how well the uncertainty measure reflects the accuracy of the chosen measurement $g(\cdot)$. Lastly, our initial results on the brain tumour dataset motivate a larger-scale quantitative validation of uncertainty estimates in the presence of pathology. Future work must examine the effect of including patients' dataset in the training data on the estimate of uncertainty components.

There are many ways in which uncertainty information could be utilised by radiologists or other users of data enhancement algorithms. First, predictive uncertainty can be used to decide when to abstain from predictions in high-risk regions of images (e.g. anomalies, out-of-distribution examples or inherently ambiguous features). For example, the original input low-resolution image can be augmented by overlaying the high-resolution prediction only in locations with sufficiently low uncertainty, before presenting to clinicians. As demonstrated by Fig. 3.9 in the context of super-resolution, such uncertainty-based quality control of predictions is potentially an effective means to maintain high accuracy of output images and also to safeguard against hallucination or removal of structures [42]. Second, the uncertainty information could be used for active learning [127] to decide which images should be labelled and included in the training set to maximally improve the model performance. Prior work [128, 129] define the acquisition function so as to select examples with high parameter uncertainty, and achieve promising results in classification and segmentation tasks. In particular, these methods are able to construct a compact and effective training dataset, and consequently improve the prediction accuracy while reducing the training time. The same idea could be naturally extended to data enhancement problems, that are typically formulated as multivariate regression tasks. For example, in the case of IQT, we could simulate a library of low-resolution and high-resolution image pairs from a large public dataset (e.g. HCP), and incrementally expand the training data by adding more examples from such a library. We should note, however, that in many data enhancement applications, obtaining a new "label" may require an extra acquisition possibly with a different scanner or modality, which may be logistically challenging. Third, another important application is transfer learning [130] where uncertainty information could be used to leverage knowledge from different but related domains or tasks. In many data enhancement applications, the test distribution can considerably deviate from the training distribution. For example, the algorithm might be trained on a synthetic dataset or images acquired from a scanner that is very different from the one used in the hospital where one plans to deploy the model. Therefore, a mechanism to adapt performance within a specific environment (e.g., based on the local patient population) [131], possibly in an online fashion [132, 133], is in demand. Recent work have shown that the Bayesian formalism provides a natural framework to use uncertainty in order to account for the difference and commonality between distributions to guide information transfer in continual learning [134, 135] or few-shot learning [136, 137] settings. Exploring the benefits of these ideas in the context of medical image enhancement remains future work.

The proposed framework for uncertainty quantification is formulated for multivariate regression in the general form, and thus is naturally applicable to many other image enhancement challenges such as: rapid image acquisition techniques e.g., compressed sensing [31], MR fingerprinting [138, 139] or sparse reconstruction [34, 33]; denoising [27] and dealiasing [36, 140]; image synthesis tasks e.g., estimating T2-weighted images from T1 [2, 6, 7], estimating CT images from MRI [8, 70, 61], and generating a high-field scan from a low-field scan [64]; data harmonisation [141, 29, 30] which aims to learn mappings among imaging protocols to reduce confounds in multicentre studies. Our results on image quality transfer [54] illustrate the potential of the uncertainty modelling techniques to improve the safety of these applications by not only improving the predictive accuracy, but also providing a mechanism to quantify risks

and safeguard against potential malfunction.

Chapter 4

Modelling Human Uncertainty

The predictive performance of supervised learning algorithms depends on the quality of labels. In a typical label collection process, multiple annotators provide subjective noisy estimates of the "truth" under the influence of their varying skill-levels and biases. Blindly treating these noisy labels as the ground truth limits the accuracy of learning algorithms in the presence of strong disagreement. This problem is critical for applications in domains such as medical imaging where both the annotation cost and inter-observer variability are high. In this work, we present a method for simultaneously learning the individual annotator model and the underlying true label distribution, using only noisy observations. Each annotator is modeled by a confusion matrix that is jointly estimated along with the classifier predictions. We propose to add a regularization term to the loss function that encourages convergence to the true annotator confusion matrix. We provide a theoretical argument as to how the regularization is essential to our approach both for the case of single annotator and multiple annotators. Despite the simplicity of the idea, experiments on image classification tasks with both simulated and real labels show that our method either outperforms or performs on par with the state-of-the-art methods and is capable of estimating the skills of annotators even with a single label available per image.

4.1 Introduction

In many practical applications, supervised learning algorithms are trained on noisy labels obtained from multiple annotators of varying skill levels and biases. When there is a substantial amount of disagreement in the labels, conventional training algorithms that treat such labels as the "truth" lead to models with limited predictive performance. To mitigate such variation, practitioners typically abide by the principle of "wisdom of crowds" [142] and aggregate labels by computing the majority vote. However, this approach has limited efficacy in applications where the number of annotations is modest or the tasks are ambiguous. For example, vision applications in medical image analysis [17] require annotations from clinical experts, which incur high costs and often suffer from high inter-reader variability [143, 144, 145, 146].

However, if the exact process by which each annotator generates the labels was known, we could correct the annotations accordingly and thus train our model on a cleaner set of data. Furthermore, this additional knowledge of the annotators' skills can be utilized to decide on which examples to be labeled by which annotators [147, 148, 149]. Therefore, methods that can accurately model the label noise of annotators are useful for improving not only the accuracy of the trained model, but also the quality of labels in the future.

Previous work proposed various methods for jointly estimating the skills of the annotators and the ground truth (GT) labels. We categorize these methods into two groups: (1) *two-*

stage approach and (2) *simultaneous* approach. Methods in the first category perform label aggregation and training of a supervised learning model in two separate steps. The noisy labels $\widetilde{\mathbf{Y}}$ are first aggregated by building a probabilistic model of annotators. The observable variables are the noisy labels $\widetilde{\mathbf{Y}}$, and the latent variables/parameters to be estimated are the annotator skills and GT labels $\mathbf{Y}$. Then, a machine learning model is trained on the pairs of aggregated labels $\mathbf{Y}$ and input examples $\mathbf{X}$ (e.g. images) to perform the task of interest. The initial attempt was made in [150] in the early 1970s and more recently, numerous lines of research [151, 146, 152, 153, 154] proposed extensions of this work e.g. by estimating the difficulty of each example. However, in all these cases, information about the raw inputs $\mathbf{X}$ is completely neglected in the generative model of noisy labels used in the aggregation step, and this highly limits the quality of estimated true labels in practice.

The *simultaneous* approaches [155, 156, 157, 158] address this issue by integrating the prediction of the supervised learning model (i.e. distribution $p(\mathbf{Y}|\mathbf{X})$) into the probabilistic model of noisy labels, and have been shown to improve the predictive performance. These methods employ variants of the expectation-maximization (EM) algorithm during training, and require a reasonable number of labels for each example. However, in most real world applications, it is practically prohibitive to collect a large number of labels per example, and this requirement limits their applications. A notable exception is the Model Boostrapped EM (MBEM) algorithm presented in [159] that is capable of learning even with little label redundancy.

In this paper, we propose a more effective alternative to these EM-based approaches for jointly modeling the annotator skills and GT label distribution. Our method separates the annotation noise from true labels by (1) ensuring high fidelity with the data by minimizing the cross entropy loss and (2) encouraging the estimated annotators to be maximally unreliable by minimizing the trace of the estimated confusion matrices. Our method is also simpler to implement, only requiring an addition of a regularization term to the cross-entropy loss. Furthermore, we provide a theoretical result that such regularization is capable of recovering the annotation noise as long as the average confusion matrix (CM) over annotators is diagonally dominant.

Experiments on image classification tasks with both simulated and real noisy labels demonstrate that our method, despite being much simpler, leads to better or comparable performance with MBEM [159] and generalized EM [155, 81], and is capable of recovering CMs even when there is only one label available per example. We simulated a diverse range of annotator types on MNIST and CIFAR10 data sets while we used a ultrasound dataset for cardiac view classification to test the efficacy in a real-world application. We also show importance of modeling individual annotators by comparing against various modern noise-robust methods [160, 161, 162, 163], when the inter-annotator variability is high.

4.1.1 Other Related Works.

More broadly, our work is related to methods for robust learning in the presence of label noise. There is a large body of literature that do not explicitly model individual annotators unlike our method.

The effects of label noise are well studied in common classifiers such as SVMs and logistic regression, and robust variants have been proposed [164, 165, 166]. More recently, various attempts have been made to train deep neural networks under label noise. Reed et al. [160] developed a robust loss to model "prediction consistency", which was later extended by [167]. In [168] and [161], label noise was parametrized in the form of a transition matrix and incorporated into neural networks for binary and multi-way classification. A more effective alternative for estimating such transition matrix was proposed in [169], and a method for capturing image dependency of label noise was shown in [170]. We will later compare our model to several of these methods to test the value of modelling individual annotators in gaining robustness to label

noise.

Multiple lines of work have shown that a small portion of clean labels improves robustness. [171] proposed to learn from clean labels to correct the labels of noisy examples. [172] proposed a method for learning to weigh examples during each training iteration by using the validation loss on clean labels as the meta-objective. [173] employs a similar approach, but trains a separate network that proposes weighting. However, curating a set of clean labels of sufficient size is expensive for many applications, and this work focuses on the scenario of learning from purely noisy labels.

4.2 Methods

We assume that a set of images $\{\mathbf{x}_i\}_{i=1}^N$ are assigned with noisy labels $\{\tilde{y}_i^{(r)}\}_{i=1,...,N}^{r=1,...,R}$ from multiple annotators where $\tilde{y}_i^{(r)}$ denotes the label from annotator r given to example $\mathbf{x}_i$, but no ground truth (GT) labels $\{y_i\}_{i=1,...,N}$ are available. In this work, we present a new procedure for multiclass classification problem that can simultaneously estimate the annotator noise and GT label distribution $p(y|\mathbf{x})$ from such noisy set of data $\mathcal{D} = \{\mathbf{x}_i, \tilde{y}_i^{(1)}, ..., \tilde{y}_i^{(R)}\}_{i=1,...,N}$. The method only requires adding a regularization term, that is the average accuracy of all annotator models, to the cross-entropy loss function. Intuitively, the method biases ours models of each annotator to be as inaccurate as possible while having the model still explain the data. We will show that this is capable of decoupling the annotation noise from the true label distribution, as long as the average labels of the real annotators are "sufficiently" correct (which we formalize in Sec. 4.2.3). For simplicity, we first describe the method in the *dense label* scenario in which each image has labels from all annotators, and then extend to scenarios with *missing* labels where only a subset of annotators label each image. As we shall see later, the method works even when each image is only labelled by a single annotator.

4.2.1 Noisy Observation Model

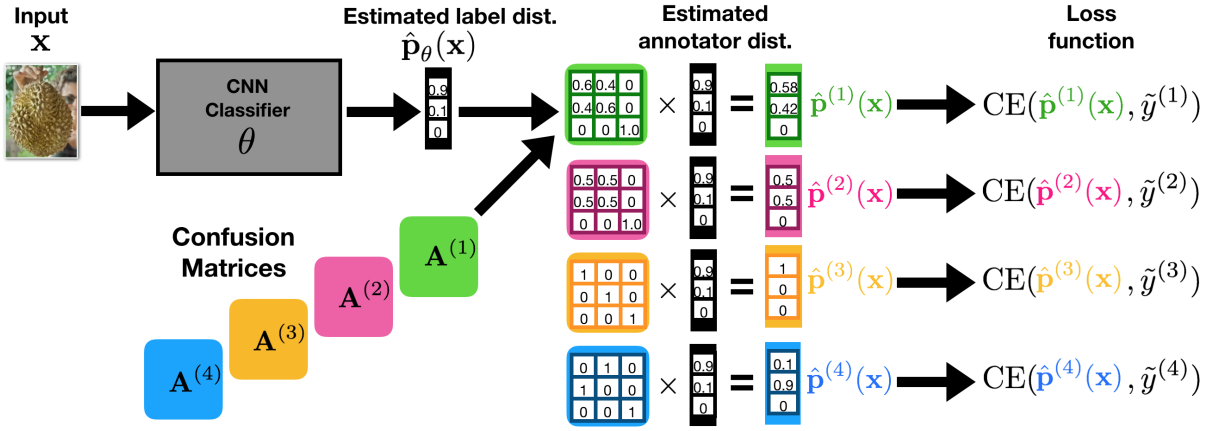


Figure 4.1: General schematic of the model (eq. 4.2) in the presence of 4 annotators. Given input image $\mathbf{x}$, the classifier parametrised by θ generates an estimate of the ground truth class probabilities, $\mathbf{p}_\theta(\mathbf{x})$. Then, the class probabilities of respective annotators $\mathbf{p}^{(r)}(\mathbf{x}) := \mathbf{A}^{(r)}\mathbf{p}_\theta(\mathbf{x})$ for $r \in \{1, 2, 3, 4\}$ are computed. The model parameters $\{\theta, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \mathbf{A}^{(4)}\}$ are optimized to minimize the sum of four cross-entropy losses between each estimated annotator distribution $\mathbf{p}^{(r)}(\mathbf{x})$ and the noisy labels $\tilde{y}^{(r)}$ observed from each annotator. The probability that each annotator provides accurate labels can be estimated by taking the average diagonal elements of the associated confusion matrix (CM), which we refer to as the "skill level" of the annotator.

We first describe our probabilistic model of the observed noisy labels from multiple annotators. In particular, we make two key assumptions: (1) annotators are statistically independent, (2) annotation noise is independent of the input image. By assumption (1), the probability of observing noisy labels $\{\tilde{y}^{(1)}, ..., \tilde{y}^{(R)}\}$ on image $\mathbf{x}$ can be written as:

$$p(\tilde{y}^{(1)}, ..., \tilde{y}^{(R)}|\mathbf{x}) = \prod_{r=1}^{R} \int_{y \in \mathcal{Y}} p(\tilde{y}^{(r)}|y, \mathbf{x}) \cdot p(y|\mathbf{x}) dy \tag{4.1}$$

where $p(y|\mathbf{x})$ denotes the true label distribution of the image, and $p(\tilde{y}^{(r)}|y, \mathbf{x})$ describes the noise model by which annotator r corrupts the ground truth label y. For classification problems, the label y takes a discrete value in $\mathcal{Y} = \{1, ..., L\}$. From assumption (2), the probability that annotator r corrupts the GT label $y = i$ to $\tilde{y}^{(r)} = j$ is independent of the image $\mathbf{x}$ i.e. $p(\tilde{y}^{(r)} = j|y = i, \mathbf{x}) = p(\tilde{y}^{(r)} = j|y = i) =: a_{ji}^{(r)}$. Here we refer to the associated $L \times L$ transition matrix $\mathbf{A}^{(r)} = (a_{ji}^{(r)})$ as the *confusion matrix* (CM) of annotator r. The joint probability over the noisy labels is simplified to:

$$p(\tilde{y}^{(1)}, ..., \tilde{y}^{(R)}|\mathbf{x}) = \prod_{r=1}^{R} \sum_{y=1}^{L} a_{\tilde{y}^{(r)},y}^{(r)} \cdot p(y|\mathbf{x}) \tag{4.2}$$

Fig. 6.5 provides a schematic of our overall architecture, which models the different constituents in the above joint probability distribution. In particular, the model consists of two components: the *base classifier* which estimates the ground truth class probability vector $\hat{\mathbf{p}}_\theta(\mathbf{x})$ whose i^{th} element approximates $p(y = i|\mathbf{x})$, and the set of the CM estimators $\{\hat{\mathbf{A}}^{(r)}\}_{r=1}^{R}$ which approximate $\{\mathbf{A}^{(r)}\}_{r=1}^{R}$. Each product $\hat{\mathbf{p}}^{(r)}(\mathbf{x}) := \hat{\mathbf{A}}^{(r)}\hat{\mathbf{p}}_\theta(\mathbf{x})$ represents the estimated class probability vector of the corresponding annotator. At inference time, we use the most confident class in $\hat{\mathbf{p}}_\theta(\mathbf{x})$ as the final classification output. Next, we describe our optimization algorithm for jointly learning the parameters of the base classifier, θ and the CMs, $\{\hat{\mathbf{A}}^{(r)}\}_{r=1}^{R}$.

4.2.2 Joint Estimation of Confusion and True labels

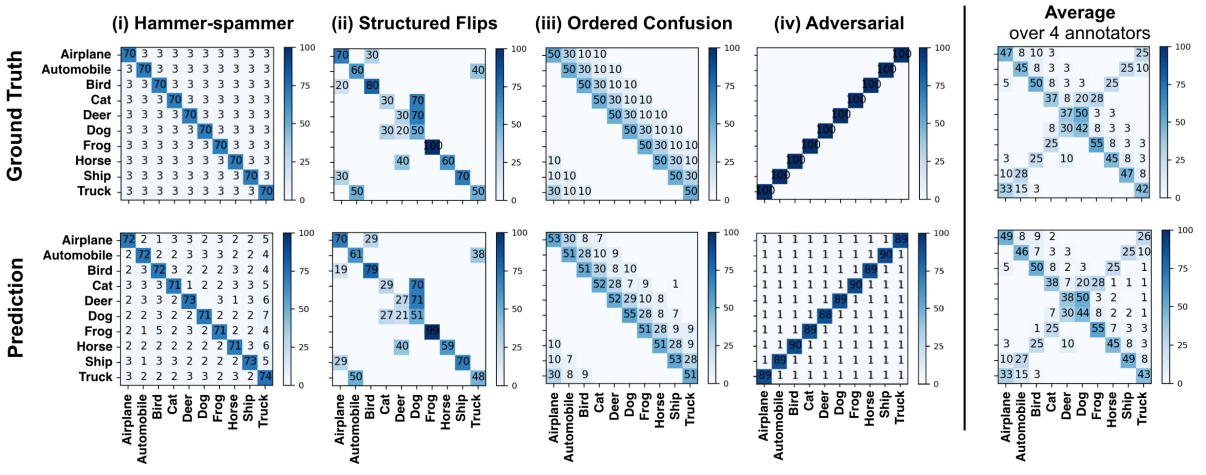


Figure 4.2: A diverse set of 4 simulated annotators on CIFAR-10. The top row shows the ground truths while the bottom row are the estimation from our method, trained with only one label per image.

Given training inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ and noisy labels $\widetilde{\mathbf{Y}}^{(r)} = \{\tilde{y}_i^{(r)}\}_{i=1}^{N}$ for $r = 1, ..., R$, we optimize the parameters $\{\theta, \hat{\mathbf{A}}^{(r)}\}$ by minimizing the negative log-likelihood (NLL), $-\log\, p(\widetilde{\mathbf{Y}}^{(1)}, ..., \widetilde{\mathbf{Y}}^{(R)}|\mathbf{X})$.

From eq. 4.2, this optimization objective equates to the sum of cross-entropy losses between the observed labels and the estimated annotator label distributions:

$$-\log p(\widetilde{\mathbf{Y}}^{(1)}, ..., \widetilde{\mathbf{Y}}^{(R)}|\mathbf{X}) = \sum_{i=1}^{N}\sum_{r=1}^{R} \text{CE}(\mathbf{A}^{(r)}\hat{\mathbf{p}}_\theta(\mathbf{x}_i), \tilde{y}_i^{(r)}). \tag{4.3}$$

Minimizing above encourages each annotator-specific prediction $\hat{\mathbf{p}}^{(r)}(\mathbf{x}) := \hat{\mathbf{A}}^{(r)}\hat{\mathbf{p}}_\theta(\mathbf{x})$ to be as close as possible to the noisy label distribution of the corresponding annotator $\mathbf{p}^{(r)}(\mathbf{x})$. However, this loss function alone is not capable of separating the annotation noise from the true label distribution; there are infinite combinations of $\{\hat{\mathbf{A}}^{(r)}\}_{r=1}^{R}$ and classification model $\hat{\mathbf{p}}_\theta$ such that $\hat{\mathbf{p}}^{(r)}$ perfectly matches the annotator's label distribution $\mathbf{p}^{(r)}$ for any input $\mathbf{x}$.

To formalize this problem, we denote the CM of the estimated true label distribution[1] $\hat{\mathbf{p}}_\theta$ by $\mathbf{P}$. The CM of the estimated annotator's label distribution $\hat{\mathbf{p}}^{(r)}$ is then given by the product $\hat{\mathbf{A}}^{(r)}\mathbf{P}$. Minimizing the cross-entropy loss (eq. 4.3) encourages $\hat{\mathbf{A}}^{(r)}\mathbf{P}$ to converge to the true CM of the corresponding annotator $\mathbf{A}^{(r)}$ i.e. $\hat{\mathbf{A}}^{(r)}\mathbf{P} \to \mathbf{A}^{(r)}$. However, there are infinitely many solutions pairs ($\hat{\mathbf{A}}^{(r)}$, $\mathbf{P}$) that satisfy the equality $\hat{\mathbf{A}}^{(r)}\mathbf{P} = \mathbf{A}^{(r)}$. This means that we need to regularize the optimization to encourage convergence to the desired solutions i.e. $\hat{\mathbf{A}}^{(r)} \to \mathbf{A}^{(r)}$ and $\mathbf{P} \to \mathbf{I}$.

To combat this problem, we propose to add the trace of the estimated CMs to the loss in eq. 4.3. Extending to the "missing labels" regime in which only a subset of annotators label each example, we derive the combined loss:

$$\sum_{i=1}^{N}\sum_{r=1}^{R} \mathbb{1}(\tilde{y}_i^{(r)} \in \mathcal{S}(\mathbf{x}_i)) \cdot \text{CE}(\hat{\mathbf{A}}^{(r)}\hat{\mathbf{p}}_\theta(\mathbf{x}_i), \tilde{y}_i^{(r)}) + \lambda \sum_{r=1}^{R} \text{tr}(\hat{\mathbf{A}}^{(r)}) \tag{4.4}$$

where $\mathcal{S}(\mathbf{x})$ denotes the set of all labels available for image $\mathbf{x}$, and $\text{tr}(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$. We simply perform gradient descent on this loss to learn $\{\theta, \hat{\mathbf{A}}^{(1)}, ..., \hat{\mathbf{A}}^{(R)}\}$.

Numerous previous work have considered the same observation model, but proposed various optimization schemes. The original work [155, 81] employed the generalized EM algorithm to estimate $\{\theta, \hat{\mathbf{A}}^{(1)}, ..., \hat{\mathbf{A}}^{(R)}\}$, and more recent work [157, 158] employed variants of hard-EM to optimize the same model. Khetan et al.,[159] proposed a method called model-bootstrapped EM (MBEM) in which the predictions of the base neural network classifier are used in the M-step update of CMs to learn from singly labelled data, which was not viable with the prior work. However, in all of the above EM-based methods, each M-step for the parameters of NN is not available in closed form and thus performed via gradient descent. This means that every M-step requires a training of the CNN classifier, rendering each iteration of EM expensive. A naive solution to this is to perform only few iterations of gradient descent in each E-step, however, this could limit the performance if sufficient convergence is not achieved. Our approach directly maximizes the likelihood with the trace regularizer and does not suffer from these issues. In Sec. 4, we show empirically this approach leads to an improvement both in terms of accuracy and convergence rate over the previous methods on noisy labels with high inter-annotator variability.

4.2.3 Motivation for Trace Regularization

Here we intend to motivate the addition of the trace regularizer in eq. 4.4. In the last section, we saw that minimizing cross-entropy loss alone encourages $\hat{\mathbf{A}}^{(r)}\mathbf{P} \to \mathbf{A}^{(r)}$. Therefore, if we could devise a regularizer which, when minimized, uniquely ensures the convergence $\hat{\mathbf{A}}^{(r)} \to \mathbf{A}^{(r)}$, then this would make $\mathbf{P}$ tend to the identity matrix, implying that the base model fully captures the

[1] $\mathbf{P}_{ji} = \int_{\mathbf{x}\in\mathcal{X}} p(\text{argmax}_k[\hat{\mathbf{p}}_\theta(\mathbf{x})]_k = j|y = i)p(\mathbf{x})d\mathbf{x}$

true label distribution i.e. $\text{argmax}_k[\mathbf{p}(\hat{\mathbf{x}})_\theta]_k = y \; \forall \mathbf{x}$. We describe below the trace regularizer is indeed a such regularizer when both $\hat{\mathbf{A}}^{(r)}$ and $\mathbf{A}^{(r)}$ satisfy some conditions. We first show this result assuming that there is a single annotator, and then extend to the scenario with multiple annotators.

Lemma 1 (Single Annotator)**.** *Let $\boldsymbol{P}$ be the CM of the estimated true labels $\hat{\boldsymbol{p}}_\theta$ and $\hat{\boldsymbol{A}}$ be the estimated CM of the annotator. If the model matches the noisy label distribution of the annotator i.e. $\hat{\boldsymbol{A}}\boldsymbol{P} = \boldsymbol{A}$, and both $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}$ are diagonally dominant ($a_{ii} > a_{ij}$, $\hat{a}_{ii} > \hat{a}_{ij}$) for all $i \neq j$, then $\hat{\boldsymbol{A}}$ with the minimal trace uniquely coincides with the true confusion matrix $\boldsymbol{A}$.*

Proof. We show that each diagonal element in the true CM $\mathbf{A}$ forms a lower bound to the corresponding element in its estimation.

$$a_{ii} = \sum_j \hat{a}_{ij} p_{ji} \leq \sum_j \hat{a}_{ii} p_{ji} = \hat{a_{ii}} (\sum_j p_{ji}) = \hat{a}_{ii} \tag{4.5}$$

for all $i \in \{1, ..., L\}$. It therefore follows that $\text{tr}(\mathbf{A}) \leq \text{tr}(\hat{\mathbf{A}})$. We now show that the equality $\hat{\mathbf{A}} = \mathbf{A}$ is uniquely achieved when the trace is the smallest i.e. $\text{tr}(\mathbf{A}) = \text{tr}(\hat{\mathbf{A}}) \Rightarrow \mathbf{A} = \hat{\mathbf{A}}$. From (4.5), if the trace of $\mathbf{A}$ and $\hat{\mathbf{A}}$ are the same, we see that their diagonal elements also match i.e. $a_{ii} = \hat{a}_{ii} \forall i \in \{1, ..., L\}$. Now, the non-negativity of all elements in CMs $\mathbf{P}$ and $\hat{\mathbf{A}}$, and the equality $a_{ii} = \sum_j \hat{a}_{ij} p_{ji}$ imply that $p_{ji} = \mathbb{1}[i = j]$ i.e. $\mathbf{P}$ is the identity matrix.

□

We note that the above result was also mentioned in [161] in a more general context of label noise modelling (that neglects annotator information). Here we further augment their proof by showing the uniqueness of solutions (i.e. $\text{tr}(\mathbf{A}) = \text{tr}(\hat{\mathbf{A}}) \Rightarrow \mathbf{A} = \hat{\mathbf{A}}$). In addition, the trace regularization was never used in practice in [161] — for implementation reason, the Frobenius norm was used in all their experiments. We now extend this to the multiple annotator regime. We will show later that minimizing the mean trace of all annotators indeed enhances the estimation quality of both CM and true label distributions, particularly in the presence of high annotator disagreement.

Theorem 1 (Multiple Annotators)**.** *Let $\hat{\boldsymbol{A}}^{(r)}$ be the estimated CM of annotator r. If $\hat{\boldsymbol{A}}^{(r)}\boldsymbol{P} = \boldsymbol{A}^{(r)}$ for $r = 1, ..., R$, and the average true and estimated CMs $\boldsymbol{A}^* := R^{-1} \sum_{r=1}^R \boldsymbol{A}^{(r)}$ and $\hat{\boldsymbol{A}}^* := R^{-1} \sum_{r=1}^R \hat{\boldsymbol{A}}^{(r)}$ are diagonally dominant, then $\boldsymbol{A}^{(1)}, ..., \boldsymbol{A}^{(R)} = \underset{\hat{\boldsymbol{A}}^{(1)}, ..., \hat{\boldsymbol{A}}^{(R)}}{\text{argmin}} \left[tr(\hat{\boldsymbol{A}}^*) \right]$ and such solutions are unique. In other words, when the trace of the mean CM is minimized, the estimation of respective annotator's CMs match the true values.*

Proof. As the average CMs $\mathbf{A}^*$ and $\hat{\mathbf{A}}^*$ are diagonally dominant and we have $\mathbf{A}^* = \hat{\mathbf{A}}^* \mathbf{P}$, Lemma 1 yields that $\text{tr}(\mathbf{A}^*) \leq \text{tr}(\hat{\mathbf{A}}^*)$ with equality if and only if $\mathbf{A}^* = \hat{\mathbf{A}}^*$. Therefore, when the trace of the average CM of annotators is minimized i.e. $\text{tr}(\hat{\mathbf{A}}^*) = \text{tr}(\mathbf{A}^*)$, the estimated CM of the true label distribution $\mathbf{P}$ reduces to identity, giving $\hat{\mathbf{A}}^{(r)} = \mathbf{A}^{(r)}$ for all $r \in \{1, ..., R\}$. □

The above result shows that if each estimated annotator's distribution $\hat{\mathbf{A}}^{(r)} \hat{\mathbf{p}}_\theta(\mathbf{x})$ is very close to the true noisy distribution $\mathbf{p}^{(r)}(\mathbf{x})$ (which is encouraged by minimizing the cross-entropy loss), and on average for each class c, the number of correctly labelled examples exceeds the number of examples of every other class c' that are mislabelled as c (the mean CM is diagonally dominant), then minimizing its trace will drive the estimates of CMs towards the true values. To encourage $\{\hat{\mathbf{A}}^{(1)}, ..., \hat{\mathbf{A}}^{(R)}\}$ to be also diagonally dominant, we initialize them with identity matrices. Intuitively, the combination of the trace term and cross-entropy separates the true distribution from the annotation noise by finding the maximal amount of confusion which can explain the noisy observations well.

4.3 Experiments

We now aim to verify the proposed method on various image recognition tasks. Particularly, we demonstrate (1) advantage of our simpler optimization scheme compared to EM-based approaches (Sec. 4.3.2), (2) importance of modeling multiple annotators (Sec. 4.3.3) and (3) the applicability of the model in a challenging real world application (Sec. 4.3.2). We address the first two questions by testing the proposed method on MNIST and CIFAR-10 datasets with a diverse set of simulated annotators. To answer the final question, we evaluate our approach on the task of cardiac view classification using ultrasound images where the labels are noisy and sparse, and are acquired from multiple annotators of varying levels of expertise.

4.3.1 Set-Up

We focus on a regime in which models have only access to noisy labels from multiple annotators. For MNIST and CIFAR-10 data sets, we simulate noisy labels from a range of annotators with different skill levels and biases.

MNIST Experiments. We consider two different models of annotator types: (i) *pairwise-flipper*: each annotator is correct with probability p or flips the label of each class to another label (the flipping target is chosen uniformly at random for each class), (ii) *hammer-spammer*: each annotator is always correct with probability p or otherwise chooses labels uniformly at random [159]. For each annotator type and skill level p, we create a group of 5 annotators by generating CMs from the associated distribution (illustration of CMs are given in the supplementary material). Given the GT labels, we generate noisy labels as defined by the CM per annotator. These noisy labels are used during training.

CIFAR-10 Experiments. We consider a diverse group of 4 annotators with different patterns of CMs as shown in Fig. 4.2: (i) is a "hammer-spammer" as defined above, (ii) tends to mix up semantically similar categories of images e.g. cats and dogs, and automobiles and trucks, (iii) is likely to confuse "neighbouring" classes and (iv) is an adversarial annotator who has a wrong association of class names to object categories. On average, labels generated by these annotators are correct only 45% of the time.

In synthetic experiments, we assume that equal number of labels are generated by each annotator on average. We also note that all models are trained on noisy labels and do not have access to the ground truth. Unless otherwise stated, we hold out 10% of training images as a validation set, on which the best performing model is selected. We also perform no data augmentation during training. Full details of training and model architectures are provided in the supplementary material. In Sec. 4.3.2 and Sec. 4.3.3 below, we compare our model against two separate sets of baselines to address different questions.

4.3.2 Comparing with EM-based Approaches

This section examines the ability of our method in learning the CMs of annotators and the GT label distribution on MNIST and CIFAR-10. In particular, we compare against two prior methods: (1) generalized EM [81], the first method for end-to-end training of the CM model in the presence of multiple annotators, and (2) Model Bootstrapped EM (MBEM) [159], the present state-of-the-art method. We analyze the performance in two cases, one in which all labels from 5 annotators are available for each image ("dense labels"), and another where only one randomly selected annotator labels each example ("1 label per image"). We quantify the error of CM estimation by the average Frobenius norm between each CM and its estimate over

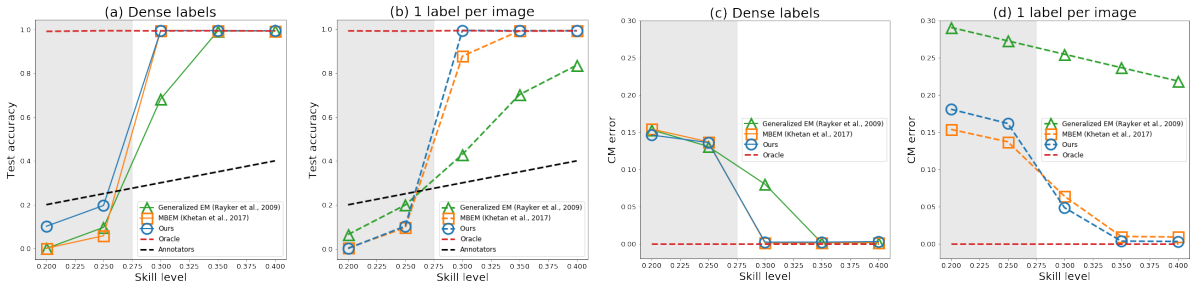


Figure 4.3: Comparison between our method, generalized EM, MBEM trained on noisy labels on MNIST from "pairwise flippers" for a range of mean skill level p. (a), (b) show classification accuracy in two cases, one where all annotators label each example and the other where only one label is available per example. (c), (d) quantify the CM recovery error as the annotator-wise average of the normalized Frobenius norm between each ground truth CM and its estimate. The shaded areas represent the cases where the average CM over the annotators are not diagonally dominant.

the annotators, and this metric is normalized to be in the range $[0, 1]$ by dividing by the number of classes L i.e. $R^{-1}L^{-1}\sum_r \sum_{i,j} ||a_{ij}^{(r)} - \hat{a}_{ij}^{(r)}||^2$.

Performance Comparison. Fig. 4.3 compares the classification accuracy and the error of CM estimation on MNIST for a range of mean skill-levels p where labels are generated by a group of 5 "pairwise-flippers". The "oracle" model is the idealistic scenario where CMs of the annotators are a priori known to the model while "annotators" indicate the average labeling accuracy of each annotator group.

Fig. 4.3 shows a strong correlation between the classification accuracy and the error of CM estimation. We observe our model displays consistently better or comparable performance in terms of both classification accuracy and estimation of CMs with dense labels (Fig. 4.3(a) and (c)). When each example receives only one label from one of the annotators, we observe the same trend as long as the mean CMs are diagonally dominant (Fig. 4.3(b,d)). We also observe that when the diagonal dominance holds, all three methods perform better than the annotators. On the other hand, when the diagonal dominance does not hold (see the grey regions), all models undergo a steep drop in classification accuracy due to the inability to estimate CMs accurately as reflected in Fig. 4.3(c,d), which is consistent with Theorem. 1. Fig. 4.4 also visualizes the average of the estimated CMs at this break point. We also note that with only one label per image, the generalized EM algorithm [155, 81] is not capable of recovering CMs at all and predict identity matrices (Fig. 4.4), which renders the model equivalent to a vanilla classifier directly trained on noisy labels. A similar set of results in the "spammer-hammer" case are also available in the supplementary materials.

On CIFAR-10 dataset, Tab. 4.1 shows that our method outperforms MBEM and the generalized EM in terms of both classification accuracy and CM estimation by a large margin. In addition, the standard deviations of these metrics are generally smaller for our method than for the baselines. Fig. 4.2 illustrates that our method can estimate CMs of the 4 very different annotators even when each image receives only one label. Interestingly, Tab. 4.1 shows that even removing the trace norm can achieve reasonably high classification accuracy and low CM estimation error. We believe this is because of the unexplained robustness of a deep CNN to label noise. Nevertheless, adding the trace norm improves the performance, and we also observe on MNIST that such improvement is pronounced in the presence of larger noise (see supplementary materials).

(a) Dense labels

Method	Accuracy	CM error
Our method	$\mathbf{81.23 \pm 0.21}$	$\mathbf{0.72 \pm 0.01}$
Our method (no trace norm)	80.29 ± 0.65	1.37 ± 0.12
MBEM [159]	73.33 ± 0.46	2.53 ± 0.24
generalized EM [155]	70.49 ± 0.23	6.13 ± 0.28
Single CM [161]	68.82 ± 2.27	-
Weighted Doctor Net [163]	60.11 ± 1.80	-
Soft-bootstrap [160]	54.73 ± 1.33	-
Vanilla CNN [160]	52.33 ± 0.31	-

(b) 1 label per image

Method	Accuracy	CM error
Our method	$\mathbf{77.65 \pm 0.31}$	$\mathbf{1.22 \pm 0.01}$
Our method (no trace norm)	76.31 ± 0.49	1.46 ± 0.27
MBEM [159]	55.97 ± 1.23	4.58 ± 0.64
generalized EM [155]	53.38 ± 0.71	4.47 ± 0.64
Single CM [161]	59.91 ± 0.98	-
Weighted Doctor Net [163]	57.98 ± 0.14	-
Soft-bootstrap [160]	42.91 ± 1.08	-
Vanilla CNN [160]	36.04 ± 1.04	-

Table 4.1: Mean classification accuracy and CM estimation errors ($\times 10^{-2}$) on CIFAR-10 with dense labels. Average annotator accuracy is 45%. Standard deviations are computed based on 3 runs with varied weight initialization.

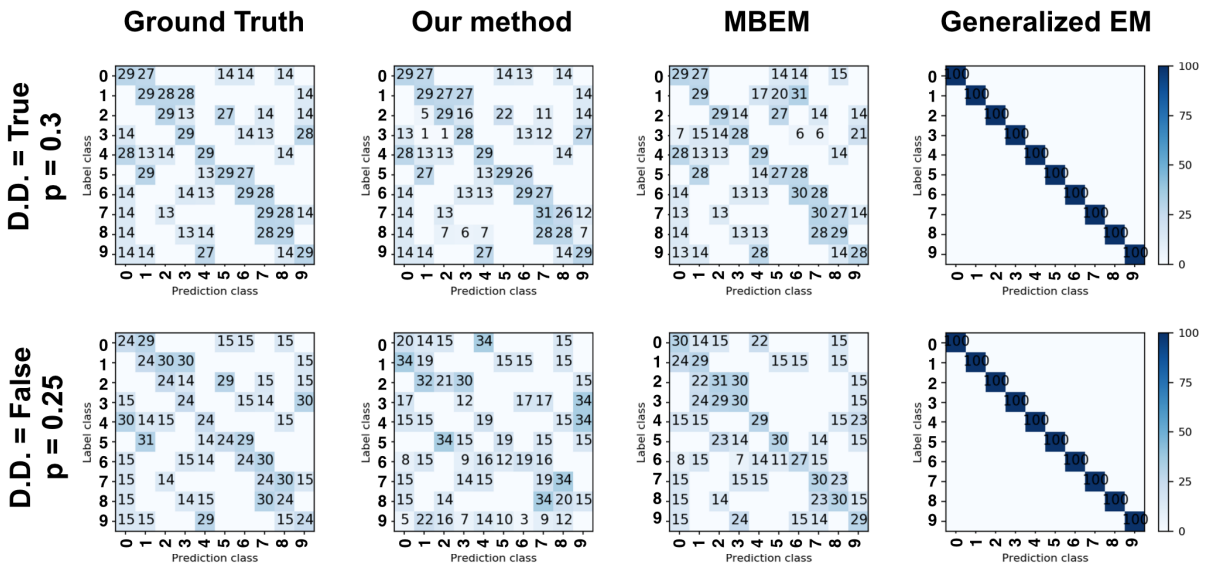


Figure 4.4: Visualization of the mean CM estimates when the diagonal dominance (D.D.) holds (mean skill level, $p = 0.3$) and does not hold ($p = 0.25$). In all cases, only one label is provided per image. The numbers are rounded to nearest integers. Here the respective models are trained on the noisy labels from 5 "pairwise flippers". Note that when each image receives only 1 label, the generalised EM [155] completely fails to recover the CM due to the failure of M-step for updating the confusion matrices (see Algorithm. 2 in the supplementary material).

Sensitivity to Hyper-parameters. We next study the robustness of our method against the generalized EM and MBEM to the specification of hyper-parameters. We used the group of five pairwise-flippers with the mean skill level $p = 0.35$ to generate noisy labels on MNIST data set. For our model, we compare the effects of the scaling λ of the trace-norm in eq. 4.4 on the trajectory of classification accuracy on the validation set and the quality of CM estimation. For the baselines, we experiment by varying the number of EM steps (denoted by T) and the number of stochastic gradient descent for each E-step (denoted by G) while fixing the total number of training iterations at $100,000$. We observed our model presents robustness to different values of λ as long as the trace-norm loss is not larger than the cross-entropy loss (where the estimated CMs will start to diffuse too much), and Fig. 4.5 shows the stability of the validation curves for $\lambda \in \{0.1, 0.01, 0.001\}$. Both the MBEM and generalized EM show evident dependence on the values of T and G and by and large display slower convergence than our method. We also observe that if too few gradient descents are performed ($G = 1000$) during each E-step, the model converges to a lower accuracy in both classification and CM estimation.

4.3.3 Value of Modelling Individual Annotators

Now, we compare the performance of our method against the prior work that aim to improve robustness to noisy labels without explicitly modelling the individual annotators. The first baseline is the vanilla classifier trained on the majority vote labels. We also compare against the noise robust approaches proposed in [160] and [161]. Reed *et al.* [160] adds to the cross-entropy loss a label consistency term based on the negative entropy of the softmax outputs, and we used the default hyper-parameter $\beta = 0.95$ for comparison. Sukhbaatar *et al.* [161] explicitly accounts for the label noise with a single CM, but does not model individual annotators. We add the trace-norm of the same scaling used in our method ($\lambda = 0.01$) to the loss function for training. We also include Weighted Doctor Net architecture (WDN) [163] in the comparison, a recent method that models the annotators individually and then learns averaging weights for

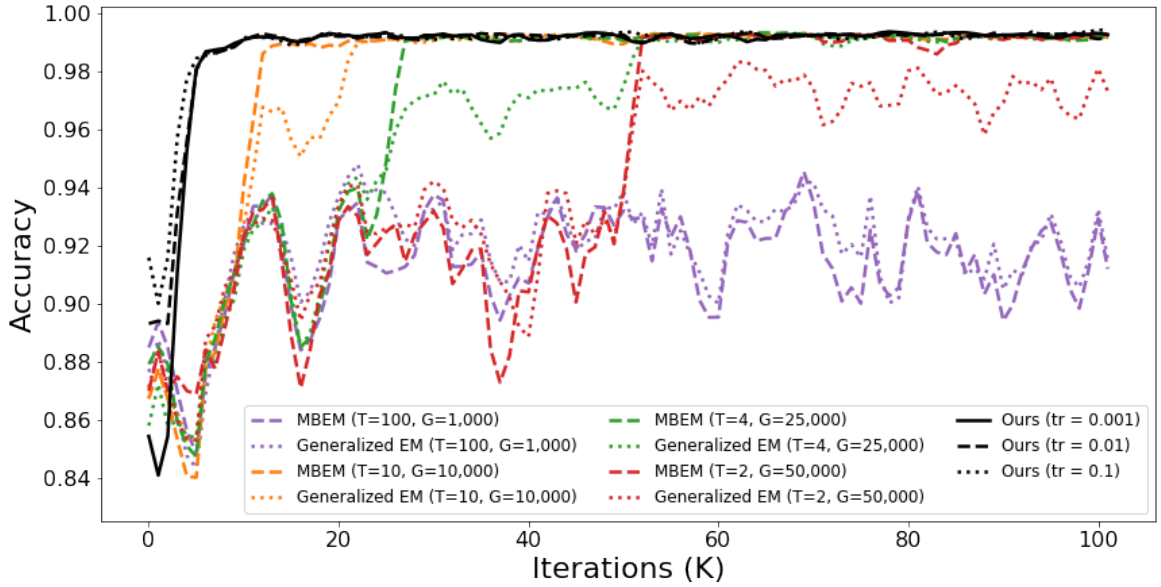


Figure 4.5: Curves of validation accuracy during training of our method, generalized EM and MBEM for a range of hyper-parameters. For our method, the scaling of the trace regularizer is varied in $[0.001, 0.01, 0.1]$. while, for EM and MBEM, we vary the number of EM steps (T), and the number of gradient descent steps per E-step (G) while fixing the total number of training iterations at $100,000$.

combining them. It should be noted that this model considers a different observation model of the labels and does not explicitly model the true label distribution. When we have access to multiple labels per example, with the exception of WDN, we aggregated the labels by computing the majority vote and trained all models. This is because we observed a consistent improvement on validation accuracy (thus poses a tougher challenge against our method) and this would be a more realistic utilization of such data set. For both MNIST and CIFAR-10 experiments, we test on the same set of simulated labels as used in Sec. 4.3.2.

Fig. 4.6 shows better or comparable classification accuracy than all the baselines when the diagonal dominance of the mean CM holds. In particular, our methods show significant improvement when the mean skill level of the annotators are relatively low (e.g. $p = 0.3$ and 0.35). The results are pronounced in the case with only one label available per image for which the baseline methods undergo a steep drop in accuracy (see Fig. 4.6(b)). Results in the "spammer-hammer" case are available in the supplementary material. Similarly on CIFAR-10 data set, Tab. 4.1 shows that our method improves the classification accuracy upon the baselines. Such improvement is pronounced in the case of sparse labels. On the other hand, a vanilla CNN with only L2 weight decay overfits to the training data very quickly in the presence of such high noise.

4.3.4 Experiments on Cardiac View Classification

Lastly, we illustrate the results of our approach for a real data set with sparse and noisy labels from the medical domain. This data set consists of images of the cardiac region in different views, acquired using a hand-held ultrasound probe. The task is to classify a given ultrasound image into one of six different view classes (see Fig. 4.7(a)). The process of obtaining a cardiac view label is crucial for guiding the user to the correct locations of measurements, and affects the quality of the downstream cardiac tasks.

A committee of sonographers (with varying levels of experience) were tasked with providing the cardiac view labels to a large volume of ultra-sound images, and each example is only

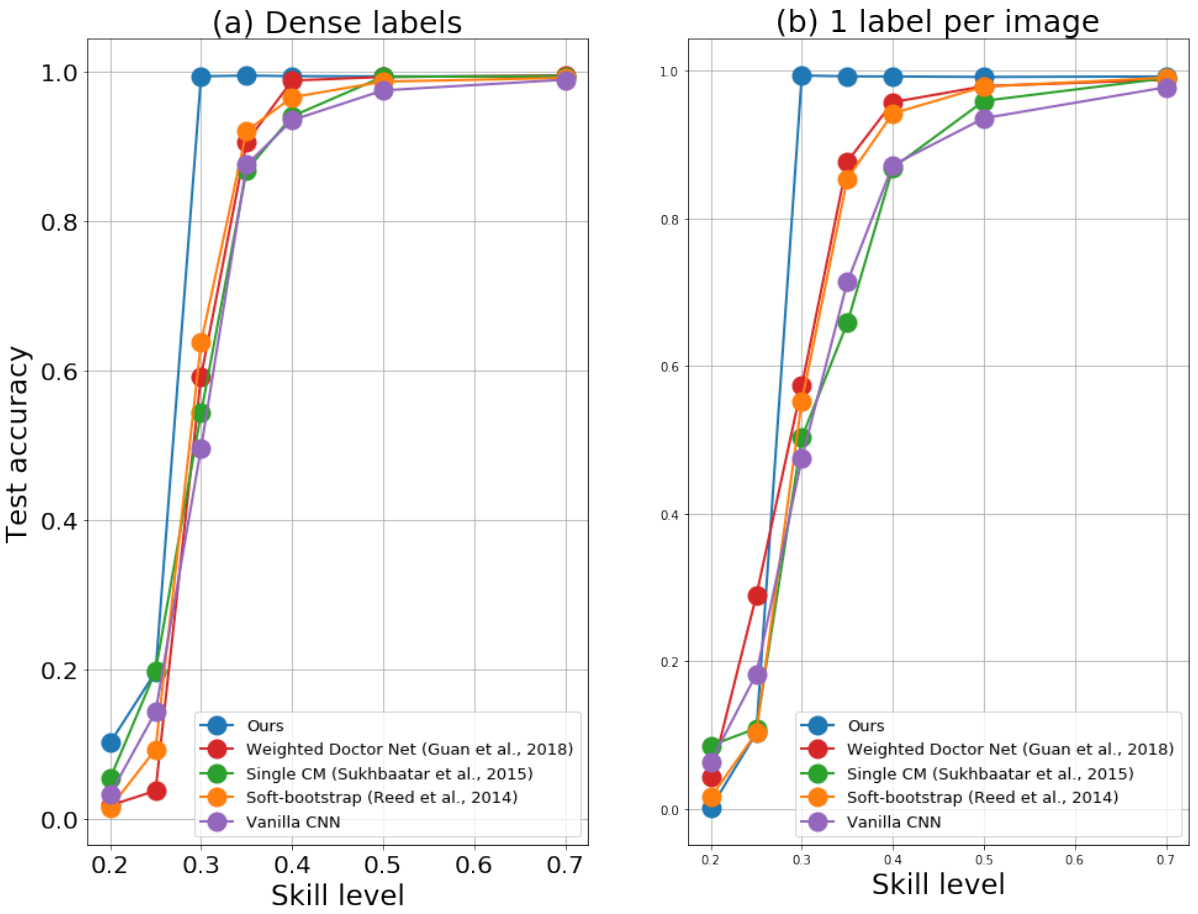
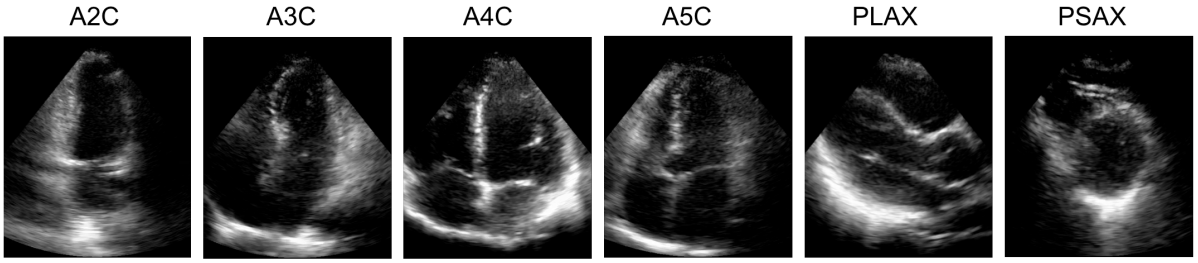


Figure 4.6: Classification accuracy on MNIST of different noise-robust models as a function of the mean annotator skill level p in two cases. Here, for each mean skill-level p, a group of 5 "pairwise flippers" is formed and used to generate labels. (a). each example receives labels from all the annotators. (b). each example is labelled by only 1 randomly selected annotator.

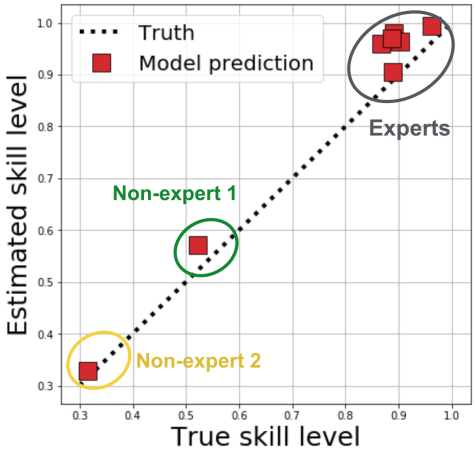
labelled by a subset of them. To acquire ground truth in this setting, we chose those samples where the three most experienced sonographers agreed on a given label. The resulting data set consists of noisy labels provided by the remaining less experienced 6 sonographers for a total of $240,000$ training images and $22,000$ validation images. In addition, we also acquired labels from two non-expert users and included in the training data.

We estimated the skill-level of each annotator by computing the average value of the diagonal elements in the corresponding learned CM, and Fig. 4.7(b) shows that the group of experts can be separated from the two non-experts with varying levels of experinces (one is less competent than the other). Fig. 4.7(c) shows that confusion between $A3C$ and $A5C$, even common among experts, can be detected (see the result for 'Expert 1') while clearly capturing the patterns of mistakes for the non-experts. In addition, Fig. 4.7(d) shows that our model outperforms MBEM [159] again in classification accuracy and the quality of CM estimation. Lastly, the higher classification accuracy of our model with respect to the other baseline models illustrates again that modelling individual annotators improves robustness to label noise.

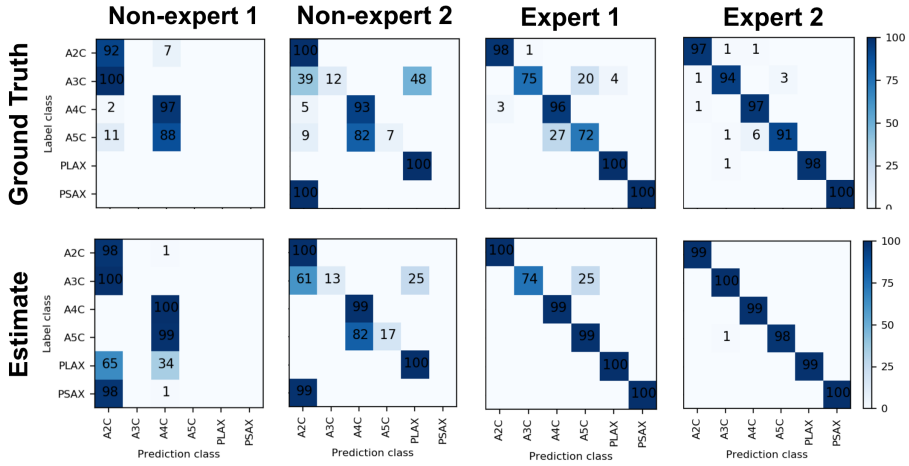
(a) Different classes of cardiac views



(b) Skill estimation



(c) Learned CMs



(d) Performance comparison

Method	Accuracy	CM error
Ours	$\mathbf{75.57 \pm 0.16}$	$\mathbf{11.48 \pm 0.48}$
w/o trace norm	70.99 ± 3.31	15.22 ± 0.94
MBEM [159]	73.91 ± 0.11	12.18 ± 0.29
WDN [163]	59.15 ± 1.60	-
Single-CM [161]	74.38 ± 0.29	-
Soft-bootstrap [160]	72.99 ± 0.17	-
Vanilla CNN	70.95 ± 0.44	-

Figure 4.7: Results on the cardiac view classification dataset: (a) illustrates examples of different cardiac view images. (b) plots the estimated skill level of each annotator (average of the diagonal elements of its estimated CM) against the ground truth (c) compares the estimated CMs of the two least skilled and two most skilled annotators according the GT labels (d) summarizes the classification accuracy and error of CM estimation for different methods.

4.4 Discussion and Conclusion

We introduced a new theoretically grounded algorithm for simultaneously recovering the label noise of multiple annotators and the ground truth label distribution. Our method enjoys implementation simplicity, requiring only adding a regularization term to the loss function. Experiments on both synthetic and real data sets have shown superior performance over the common EM-based methods in terms of both classification accuracy and the quality of confusion matrix estimation. Comparison against the other modern noise-robust methods demonstrates that the modelling individual annotators improves robustness to label noise. Furthermore, the method is capable of estimating annotation noise even when there is a single label per image.

Our work was primarily motivated by medical imaging applications for which the number of classes are mostly limited to below 10. However, future work shall consider imposing structures on the confusion matrices to broaden up the applicability to massively multi-class scenarios e.g. introducing taxonomy based sparsity [158] and low-rank approximation. We also assumed that there is only one ground truth for each input; this no longer holds true when the input images are truly ambiguous—recent advances in modelling multi-modality of label distributions [174, 79] potentially facilitate relaxation of such assumption. Another limiting assumption is the image independence of the annotator's label noise. The majority of disagreement between annotators arise in the difficult cases. Integrating such input dependence of label noise [156, 175] is also a valuable next step.

Chapter 5

Part I: Uncertainty in Multitask Learning

Multi-task neural network architectures provide a learning mechanism that jointly synthesises information from distinct sources. The success of multi-task learning stems from inductive transfer, which is known to improve the performance of a model by learning task-invariant representations. Multi-task learning is ideal in the context of MR-only radiotherapy planning as it can jointly synthesize a synthetic CT (synCT) scan (regression) and an automated contour of organs-at-risk (segmentation) from MRI data. We propose to use a probabilistic deep-learning formulation to estimate the *intrinsic* and *parameter* uncertainty through a heteroscedastic noise model, where parameter uncertainty is modelled using approximate Bayesian inference. This allows the sampling of multiple segmentations and synCTs that share their network representation, enabling a fully end-to-end uncertainty aware MRI-based probabilistic planning system. We show that our model produces more accurate and consistent synCTs with a better estimation in the variance of the errors, equivalent state of the art results in OAR segmentation and a methodology for quality assurance in treatment planning. The model also produces samples with anatomically-consistent segmentations and synCTs, a necessary characteristic in probabilistic planning models.

5.1 Introduction

The state-of-the-art in radiotherapy treatment planning requires acquiring a magnetic resonance (MR) scan to accurately segment the target and surrounding organs-at-risk (OARs), and a registered computed tomography (CT) scan to inform the photon attenuation. This approach has seen limited translation to clinical practice as extensive data acquisition is time-consuming and modality registration may introduce unacceptable errors that propagate in the planning process. MR-only treatment planning has been proposed as a solution to these issues and involves the generation of a synthetic CT (synCT) scan from the MR scan. This deterministic synthesis process, when combined with a hand drawn region of interest and a set of safety margins, provides a deterministic radiotherapy plan that is intrinsically dependent on the quality of the inputs. On the other side, probabilistic planning systems allow the implicit estimation of dose delivery uncertainty through a Monte Carlo sampling scheme. A system that can sample possible synCT and OAR segmentations would enable the development of a fully end-to-end uncertainty-aware probabilistic planning system.

Classical methods for simulating a synCT and segmenting corresponding MR scans have originated from multi-atlas propagation [176]. Recently, applications of convolutional neural networks (CNNs) to CT synthesis from MRI have become a topic of growing interest due to their

reconstruction performance. To alleviate the problem of missing high-frequency information in synCT due to mean-squared reconstruction loss, Nie et al. [177] employed a conditional generative adversarial network to capture fine texture details whilst Wolterink et al. [178] extended this idea using a CycleGAN to leverage the abundance of unpaired training sets of CT and MR scans. However, despite this advancement, all of these methods commit to a single prediction, with no measure of confidence. The lack of uncertainty information limits utility in view of current and future probabilistic dose delivery systems. Also, knowing the variance of the predictions would enable the development of quality assurance systems, improving the safety of such a system. Finally, none of the CNN-based methods segment OARs, and if a model was trained to do so, it would not produce OAR segmentations and synCT results that are consistent, which is necessary in radiotherapy planning.

Kendall et al.[179] recently devised a method for uncertainty modelling, showing its value in multi-task learning problems. However, this model assumes homoscedastic (constant) noise in the task output, which is unrealistic for most imaging data and yields non-meaningful measures of predictive uncertainty in a regression task. Tanno et al. [48] and Kendall et al. [49] raised the importance of modelling both *intrinsic* and *parameter* uncertainty to build more robust predictive models for medical image analysis and computer vision. The former describes noise inherent in the observations whilst the latter quantifies the degree of ambiguity in the model parameters given the observed data.

This paper makes use of [48] to enrich the probabilistic multi-task learning method proposed in [179], enabling modelling of the spatial variation of intrinsic uncertainty via heteroscedastic noise, and integrating parameter uncertainty via dropout. We thus propose a probabilistic dual-task network, which operates on a MR image and simultaneously provides three valuable outputs necessary for probabilistic radio-therapy planning: (1) generation of syncCT; (2) segmentation of the OARs; (3) quantification of predictive uncertainty in (1) and (2). The proposed architecture integrates the methods of uncertainty modeling in CNNs [48, 49] into a multi-task learning framework with hard-parameter sharing, in which the initial few layers of the network are shared across the two tasks (synthesis and segmentation) and branch out into task-specific layers (Fig.5.1). This probabilistic formulation not only provides an estimate of uncertainty over predictions from which one can stochastically sample the space of solutions, but also naturally confers a mechanism to automatically select the relative weighting of task losses, in contrast with prior work which relied on uniform or hand-tuned weighting [180].

5.2 Methods

We propose a probabilistic dual-task CNN-based algorithm which takes an MRI image, and simultaneously estimates the distribution over the corresponding CT image and the segmentation probability of the OARs. We use a heteroscedastic noise model and binary dropout to account for *intrinsic* and *parameter* uncertainty, respectively, and show that we obtain not only a measure of uncertainty over prediction, but also a mechanism for data-driven adaptation of weightings of task losses, which is integral for benefiting from the multi-task learning framework. We employ a patch-based approach to perform both tasks, in which the input MR image is split into smaller overlapping patches that are processed independently. For each input patch $\mathbf{x}$, our dual-task model estimates the conditional distributions $p(\mathbf{y}_i|\mathbf{x})$ for tasks $i = 1, 2$ where $\mathbf{y}_1$ and $\mathbf{y}_2$ denote the Hounsfield Unit and class probabilities of OARs at the center of the input patch. At inference, the probability maps over the synCT and OARs are obtained by stitching together outputs from appropriately shifted versions of the input patches.

5.2.1 Dual-task architecture

We perform multi-task learning with hard-parameter sharing [181]. The model shares the initial few layers across the two tasks and branches out into four task-specific networks with separate parameters (Fig.5.1). There are two networks per task, where one aims to performs CT synthesis (regression) or OAR segmentation, and the remaining models *intrinsic* uncertainty associated to the data and the task.

The rationale behind shared layers is to learn a joint representation between two tasks to regularise the learning of features for one task by using cues from the other. We used a high-resolution network architecture (HighResNet) [182] as the shared trunk of the model for its compactness and accuracy shown in brain parcellation. HighResNet is a fully convolutional architecture that utilises dilated convolutions and residual connections to produce an end-to-end mapping from an input patch ($\mathbf{x}$) to voxel-wise predictions ($\mathbf{y}$).

The final layer of the shared representation is split into two task-specific compartments (Fig. 5.1). Each compartment consists of two fully convolutional networks which operate on the output of representation network and together learn task-specific representation and define likelihood function $p\left(\mathbf{y}_i|\mathbf{W},\mathbf{x}\right)$ for each task $i=1,2$ where $\mathbf{W}$ denotes the set of all parameters of the model.

5.2.2 Task weighting with heteroscedastic uncertainty.

Previous probabilistic multitask learning methods based on deep learning [179] assumed constant intrinsic uncertainty in respective tasks. In our context, this means that the inherent ambiguity present in synthesis or segmentation tasks do not depend on the spatial locations within an image volume. This is a highly unrealistic assumption as these tasks can be more challenging on some anatomical structures (e.g. tissue boundaries) than others. In order to capture potential spatial variation in intrinsic uncertainty, we adapt the *heteroscedastic* (data-dependent) noise model to our multitask learning problem.

In particular, for the CT synthesis task, we define our likelihood as a normal distribution $p\left(\mathbf{y}_1|\mathbf{W},\mathbf{x}\right)=\mathcal{N}(f_1^{\mathbf{W}}(\mathbf{x}),\sigma_1^{\mathbf{W}}(\mathbf{x})^2)$ where mean $f_1^{\mathbf{W}}(\mathbf{x})$ and variance $\sigma_1^{\mathbf{W}}(\mathbf{x})^2$ are modelled by the regression output and uncertainty branch as functions of the input patch $\mathbf{x}$ (see Fig.5.1). We define the task loss for CT synthesis to be the negative log-likelihood $\mathcal{L}_1(\mathbf{y}_1,\mathbf{x};\mathbf{W})=\frac{1}{2\sigma_1^{\mathbf{W}}(\mathbf{x})^2}||\mathbf{y}_1-f_1^{\mathbf{W}}(\mathbf{x})||^2+\log\sigma_1^{\mathbf{W}}(\mathbf{x})^2$. This loss encourages assigning high-uncertainty to regions of high errors, enhancing the robustness of the network against noisy labels and outliers, which are prevalent at organ boundaries especially close to the bone.

For the segmentation, we define the classification likelihood as softmax function of scaled logits i.e. $p\left(\mathbf{y}_2|\mathbf{W},\mathbf{x}\right)=\text{Softmax}\big(f_2^{\mathbf{W}}(\mathbf{x})/2\sigma_2^{\mathbf{W}}(\mathbf{x})^2\big)$ where the segmentation output $f_2^{\mathbf{W}}(\mathbf{x})$ is scaled by the uncertainty term $\sigma_2^{\mathbf{W}}(\mathbf{x})^2$ before softmax (Fig.5.1). As the uncertainty term $\sigma_2^{\mathbf{W}}(\mathbf{x})$ increases, the Softmax output approaches a uniform distribution, which corresponds

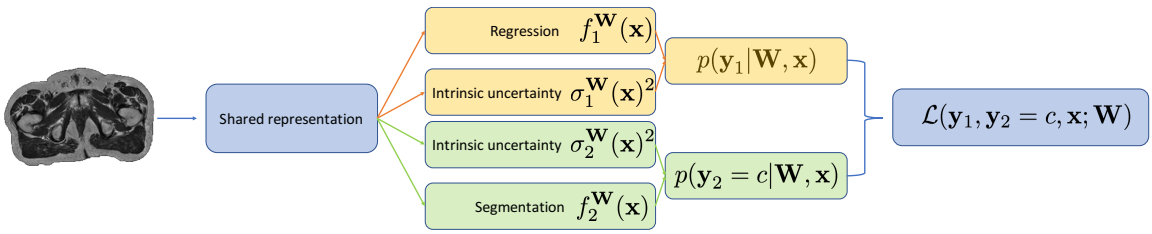


Figure 5.1: Multi-task learning architecture. The predictive mean and variance $[f_i^{\mathbf{W}}(\mathbf{x}),\sigma_i^{\mathbf{W}}(\mathbf{x})^2]$ are estimated for the regression and segmentation. The task-specific likelihoods $\text{p}(\mathbf{y}_i|\mathbf{W},\mathbf{x})$ are combined to yield the multi-task likelihood $\text{p}(\mathbf{y}_1,\mathbf{y}_2|\mathbf{W},\mathbf{x})$.

to the maximum entropy discrete distribution. We simplify the scaled Softmax likelihood by considering an approximation in [179]

$$\frac{1}{\sigma^2}\sum_{c'}\exp(\frac{1}{2\sigma_2^{\mathbf{W}}(\mathbf{x})^2}f_{2,c'}^{\mathbf{W}}(\mathbf{x})) \approx \left(\sum_{c'}\exp(f_{2,c'}^{\mathbf{W}}(\mathbf{x}))\right)^{1/2\sigma_2^{\mathbf{W}}(\mathbf{x})^2}$$

where c' is denotes a segmentation class. This yields the NLL task-loss of the form $\mathcal{L}_2(\mathbf{y}_2 = c, \mathbf{x}; \mathbf{W}) \approx \frac{1}{2\sigma_2^{\mathbf{W}}(\mathbf{x})^2}\text{CE}(f_2^{\mathbf{W}}(\mathbf{x}), \mathbf{y}_2 = c) + \log\sigma_2^{\mathbf{W}}(\mathbf{x})^2$, where CE denotes cross-entropy. Finally, assuming that the two tasks $\mathbf{y}_1, \mathbf{y}_2$ are statistically independent given the input image $\mathbf{x}$, the joint likelihood factorises over tasks $p\left(\mathbf{y}_1, \mathbf{y}_2|\mathbf{W}, \mathbf{x}\right) = p\left(\mathbf{y}_1|\mathbf{W}, \mathbf{x}\right)p\left(\mathbf{y}_2|\mathbf{W}, \mathbf{x}\right)$, and thus we can derive the NLL loss for the dual-task model as

$$\mathcal{L}(\mathbf{y}_1, \mathbf{y}_2 = c, \mathbf{x}; \mathbf{W}) = \frac{||\mathbf{y}_1 - f_1^{\mathbf{W}}(\mathbf{x})||^2}{2\sigma_1^{\mathbf{W}}(\mathbf{x})^2} + \frac{\text{CE}(f_2^{\mathbf{W}}(\mathbf{x}), \mathbf{y}_2 = c)}{2\sigma_2^{\mathbf{W}}(\mathbf{x})^2} + \log\Big(\sigma_1^{\mathbf{W}}(\mathbf{x})^2\sigma_2^{\mathbf{W}}(\mathbf{x})^2\Big)$$

where the MSE and CE terms are weighted by the inverse of heteroscedastic intrinsic uncertainty terms $\sigma_i^{\mathbf{W}}(\mathbf{x})^2$, that enables automatic weighting of task losses on a per-sample basis. The log-term controls the spread.

5.2.3 Parameter uncertainty with approximate Bayesian inference.

In data-scarce situations, the choice of best parameters is ambiguous, and resorting to a single estimate without regularisation often leads to overfitting. Gal et al.[183] have shown that dropout improves the generalisation of a NN by accounting for *parameter* uncertainty through an approximation of the posterior distribution over its weights $q(\mathbf{W}) \approx (\mathbf{W}|\mathbf{X}, \mathbf{Y_1}, \mathbf{Y_2})$ where $\mathbf{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$, $\mathbf{Y}_1 = \{\mathbf{y}_1^{(1)}, ..., \mathbf{y}_1^{(N)}\}$, $\mathbf{Y}_2 = \{\mathbf{y}_2^{(1)}, ..., \mathbf{y}_2^{(N)}\}$ denote the training data. In this work, we also use dropout in our model to assess the benefit of modelling parameter uncertainty in the context of our multitask learning problem.

During training, for each input (or minibatch), network weights are drawn from the approximate posterior $w' \sim q(\mathbf{W})$ to obtain the multi-task output (predictive mean and predictive variance), $\mathbf{f}^{w'}(\mathbf{x}) := [f_1^{w'}(\mathbf{x}), f_2^{w'}(\mathbf{x}), \sigma_1^{w'}(\mathbf{x})^2, \sigma_2^{w'}(\mathbf{x})^2]$. At test time, for each input patch $\mathbf{x}$ in a MR scan, we collect output samples $\{\mathbf{f}^{w^{(t)}}(\mathbf{x})\}_{t=1}^{\text{T}}$ by performing T stochastic forward-passes with $\{w^{(t)}\}_{t=1}^{\text{T}} \sim q(\mathbf{W})$. For the regression, we calculate the expectation over the T samples in addition to the variance, which is the *parameter* uncertainty. For the segmentation, we compute the expectation of class probabilities to obtain the final labels. The *parameter* uncertainty in the segmentation is obtained by considering variance of the stochastic class probabilities. The final predictive uncertainty is the sum of the *intrinsic* and *parameter* uncertainties.

5.2.4 Implementation details

We trained our model on randomly selected 2D axial slices and reconstructed the 3D volume at test time. The representation network was composed of a convolutional layer followed by 3 sets of twice repeated dilated convolutions with dilation factors $[1, 2, 4]$ and a final convolutional layer. Each layer (l) used a 3×3 kernel with features $f_R = [64, 64, 128, 256, 2048]$. Each task-specific branch was a set of 5 convolutional layers of size $[256_{l=1,2,3,4}, n_{i,l=5}]$ where $n_{i,l=5}$ is equal to 1 for regression and $\boldsymbol{\sigma}$ and equal to the number of segmentation classes. The first two layers were 3×3 kernels whilst the final convolutional layers were fully connected. A Bernouilli drop-out mask with probability $p = 0.5$ was applied on the final layer of the representation network. We minimised the loss using ADAM with a learning rate 10^{-3} and trained for $19,000$ iterations. For the stochastic sampling, we performed model inference 10 times at iterations 18000 and 19000 leading to a set of $T = 20$ samples.

5.3 Experiments and Results

5.3.1 Data

We validated on 15 prostate cancer patients, who each had a T2-weighted MR image (3T, $1.46 \times 1.46 \times 5\text{mm}^3$) and a CT image (140kVp, $0.98 \times 0.98 \times 1.5\text{mm}^3$) acquired on the day. Organ delineation was performed by a clinician with labels for the left and right femur head, bone, prostate, rectum and bladder. All images were resampled to isotropic resolution. The CT scans were spatially aligned with the T2 scans using the method of Burgos et al. [176]. In the segmentation, we predicted labels for the background, left/right femur head, prostate, rectum and bladder. The bone region was used for quantifying the synCT.

5.3.2 Experiments

We performed a 3-fold cross-validation. Statistics over all hold-out sets are reported. We considered four separate models; 1) baseline networks for regression/segmentation (M1), 2) baseline network with drop-out for regression/segmentation (M2a), 3) the baseline with drop-out and heteroscedastic noise (M2b), 4) multi-task network using homoscedastic task weighting (M3) [179] and 5) multi-task network using task-specific heteroscedastic noise and drop-out (M4). The baseline networks used only the representation network with $1/2 f_R$ and a fully-connected layer for the final output. We also compared our results against the current state of the art in atlas propagation (AP) [176], which was validated on the same dataset.

5.3.3 Model performance

We calculated the Mean Absolute Error (MAE) between the predicted and reference scans across the body and at each organ (Tab. 5.1). The fuzzy DICE score between the probabilistic segmentation and the reference was calculated for the segmentation (Tab. 5.1). Best performance was in our presented method (M4) for the regression across all segmentation masks

Table 5.1: Model comparison. Bold values indicate where a model was significantly worse than M4 $p < 0.05$. No data was available for significance testing with AP. M2b was statistically better $p < 0.05$ than M4 in the prostate segmentation.

Models	All	Bone	*L* femur	*R* femur	Prostate	Rectum	Bladder
Regression - synCT - Mean Absolute Error (HU)							
M1	**48.1(4.2)**	**131(14.0)**	78.6(19.2)	**80.1(19.6)**	**37.1(10.4)**	63.3(47.3)	**24.3(5.2)**
M2a	**47.4(3.0)**	**130(12.1)**	78.0(14.8)	77.0(13.0)	**36.5(7.8)**	67(44.6)	**24.1(7.5)**
M2b [49]	44.5(3.6)	128(17.1)	75.8(20.1)	74.2(17.4)	31.2(7.0)	56.1(45.5)	17.8(4.7)
M3 [179]	44.3(3.1)	126(14.4)	74.0(19.5)	73.7(17.1)	29.4(4.7)	58.4(48.0)	18.2(3.5)
AP [176]	45.7(4.6)	125(10.3)	-	-	-	-	-
M4 (ours)	43.3(2.9)	121(12.6)	69.7(13.7)	67.8(13.2)	28.9(2.9)	55.1(48.1)	18.3(6.1)
Segmentation - OAR - Fuzzy DICE score							
M1	-	-	0.91(0.02)	0.90(0.04)	0.67(0.12)	0.70(0.15)	0.92(0.05)
M2a	-	-	0.85(0.03)	0.90(0.04)	0.66(0.12)	0.69(0.13)	0.90(0.07)
M2b [49]	-	-	0.92(0.02)	0.92(0.01)	0.77(0.07)	0.74(0.13)	0.92(0.03)
M3 [179]	-	-	0.92(0.02)	0.92(0.02)	0.73(0.07)	0.76(0.10)	0.93(0.02)
AP [176]	-	-	0.89(0.02)	0.90(0.01)	0.73(0.06)	0.77(0.06)	0.90(0.03)
M4 (ours)	-	-	0.91(0.02)	0.91(0.02)	0.70(0.06)	0.74(0.12)	0.93(0.04)

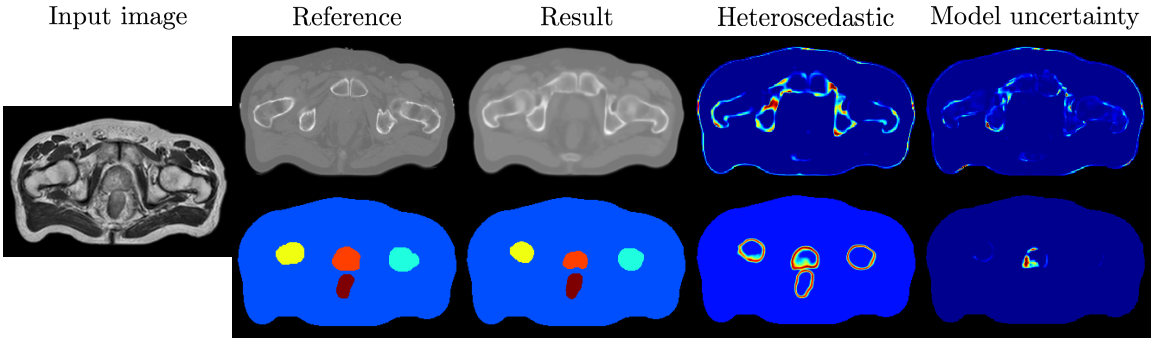


Figure 5.2: Example output from the proposed network. The heteroscedastic intrinsic and the parameter uncertainty both correlate strongly with regions of high contrast (bone in the regression), organ boundary (segmentation) and errors in the output.

except at the bladder. Application of the multi-task heteroscedastic network with drop-out (M4) produced the most consistent synCT across all models with the lowest average MAE and the lowest variation across patients (43.3 ± 2.9 versus 45.7 ± 4.6 [176] and 44.3 ± 3.1 [179]). This was significant lower when compared to M1 ($p < 0.001$) and M2 ($p < 0.001$). This was also observed at the bone and prostate ($p < 0.001$). Whilst differences at $p < 0.05$ levels of significance was not observed versus M2b and M3, the consistent lower MAE and standard deviation across patients in M4 demonstrates the added benefit of modelling heteroscedastic noise and the inductive transfer from the segmentation task. Moreover, we performed better than the current state of the art in atlas propagation [176]. despite equivalence with the state of the art (Tab. 5.1), we did not observe any significant differences between our model and the baselines despite an improvement in mean DICE at the prostate and rectum (0.70 ± 0.06 and 0.74 ± 0.12) versus the baseline M1 (0.67 ± 0.12, 0.70 ± 0.15). The *intrinsic uncertainty* (Fig. 5.2) models the uncertainty specific to the data and thus penalises regions of high error leading to an under-segmentation yet with higher confidence in the result.

5.3.4 Uncertainty estimation for radiotherapy

We tested the ability of the multi-task heteroscedastic network to better predict associated uncertainties in the synCT error. To verify that our network produces clinically viable samples for treatment planning, we quantified the distribution of regression z-scores for the multi-task heteroscedastic and homoscedastic models. In the former, the total predictive uncertainty is the sum of the *intrinsic* and *parameter* uncertainties, which is used to normalise the error between the synCT and the reference. This should lead to a better approximation of the variance in the model. In contrast, the total uncertainty in the latter reduces to the variance of the stochastic test-time samples. This is likely to lead to a mis-calibrated variance. A χ^2 goodness of fit test was performed, showing that the homoscedastic z-score distribution is not normally distributed (0.82 ± 0.54, $p < 0.01$) in contrast to the heteroscedastic model (0.04 ± 0.84, $p > 0.05$), which has overestimated the variance. This is apparent in Fig. 5.3 where there is greater confidence in the synCT produced by our model in contrast the homoscedastic case.

The predictive uncertainty can be exploited for quality assurance (Fig. 5.4). There may be issues whereupon time differences have caused variations in bladder and rectum filling across MR and CT scans causing patient variability in the training data. This is exemplified by large errors in the synCT at the rectum (Fig. 5.4) and quantified by large localised z-scores (Fig. 5.4g), which correlate strongly with the *intrinsic* and *parameter* uncertainty across tasks.

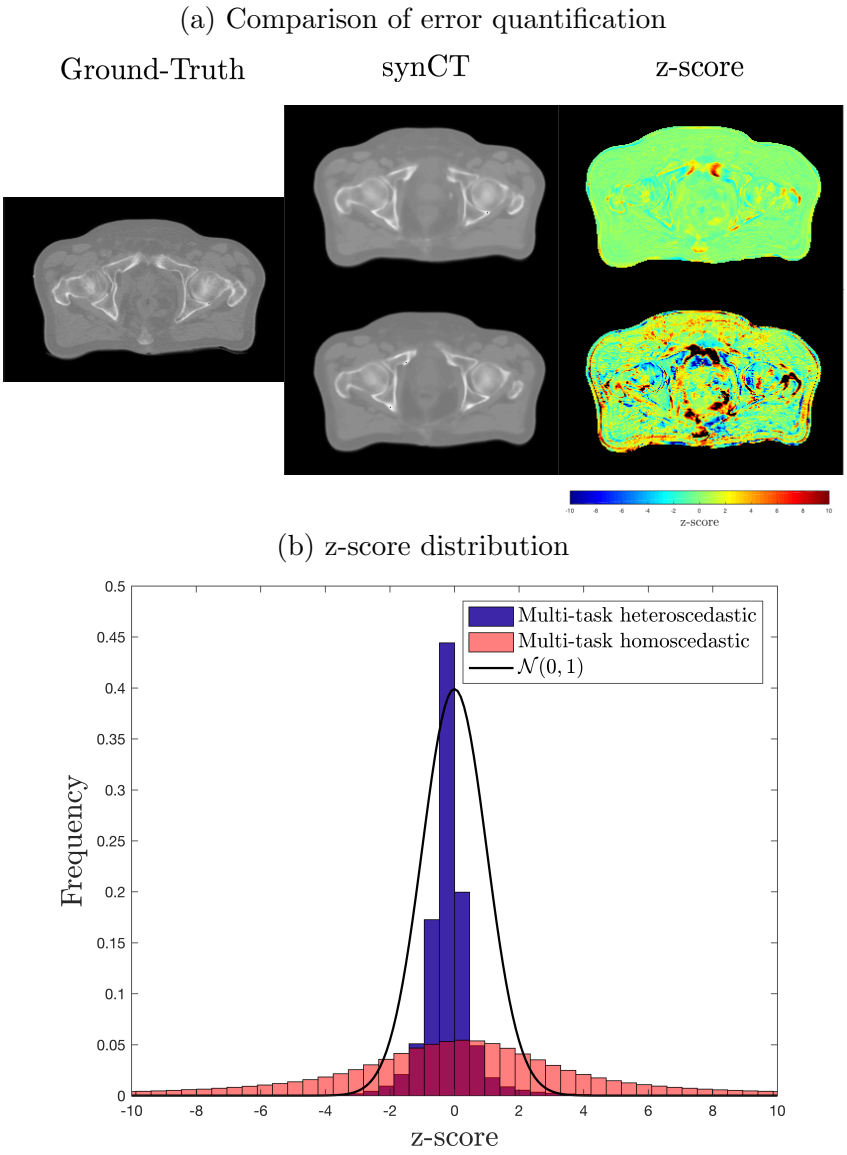


Figure 5.3: a) synCTs and z-scores for the same subject between M4 (top) and M3 (bottom) models. b) z-score distribution of all patients (15) between both models.

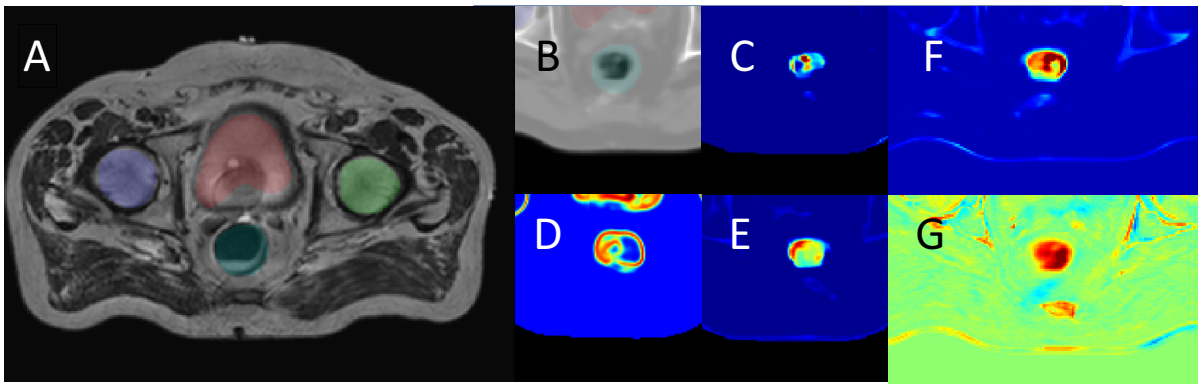


Figure 5.4: Uncertainty in areas prone to errors. a) T2 with reference segmentation, b) synCT with errors at the rectum, c) parameter uncertainty in synCT, d) intrinsic uncertainty in the segmentation, e) intrinsic uncertainty in the regression, f) total predictive uncertainty in the regression and g) error in HU (range [-750HU, 750HU]).

5.4 Conclusions

We have proposed a probabilistic dual-network that combines uncertainty modelling with multi-task learning. Our network extends prior work in multi-task learning by integrating heteroscedastic uncertainty modelling to naturally weight task losses and maximize inductive transfer between tasks. We have demonstrated the applicability of our network in the context of MR-only radiotherapy treatment planning. The model simultaneously provides the generation of synCTs, the segmentation of OARs and quantification of predictive uncertainty in both tasks. We have shown that a multi-task framework with heteroscedastic noise modelling leads to more accurate and consistent synCTs with a constraint on anatomical consistency with the segmentations. Importantly, we have demonstrated that the output of our network leads to consistent anatomically correct stochastic synCT samples that can potentially be effective in treatment planning.

Chapter 6

Part II: Uncertainty in Multitask Learning

The performance of multi-task learning in Convolutional Neural Networks (CNNs) hinges on the design of feature sharing between tasks within the architecture. The number of possible sharing patterns are combinatorial in the depth of the network and the number of tasks, and thus hand-crafting an architecture, purely based on the human intuitions of task relationships can be time-consuming and suboptimal. In this paper, we present a probabilistic approach to learning task-specific and shared representations in CNNs for multi-task learning. Specifically, we propose "stochastic filter groups" (SFG), a mechanism to assign convolution kernels in each layer to "specialist" or "generalist" groups, which are specific to or shared across different tasks, respectively. The SFG modules determine the connectivity between layers and the structures of task-specific and shared representations in the network. We employ variational inference to learn the posterior distribution over the possible grouping of kernels and network parameters. Experiments demonstrate that the proposed method generalises across multiple tasks and shows improved performance over baseline methods.

6.1 Introduction

Multi-task learning (MTL) aims to enhance learning efficiency and predictive performance by simultaneously solving multiple related tasks [184]. Recently, applications of convolutional neural networks (CNNs) in MTL have demonstrated promising results in a wide-range of computer vision applications, ranging from visual scene understanding [185, 186, 187, 188, 189, 190] to medical image computing [180, 191, 192, 193].

A key factor for successful MTL neural network models is the ability to learn shared and task-specific representations [187]. A mechanism to understand the commonalities and differences between tasks allows the model to transfer information between tasks while tailoring the predictive model to describe the distinct characteristics of the individual tasks. The quality of such representations is determined by the architectural design of where model components such as features [194] and weights [195] are shared and separated between tasks. However, the space of possible architectures is combinatorially large, and the manual exploration of this space is inefficient and subject to human biases. For example, Fig. 6.1 shows a typical CNN architecture for MTL comprised of a shared "trunk" feature extractor and task-specific "branch" networks [193, 196, 197, 179, 189, 192]. The desired amount of shared and task-specific representations, and their interactions within the architecture are dependent on the difficulty of the individual tasks and the relation between them, neither of which are a priori known in most cases [198]. This illustrates the challenge of handcrafting an appropriate architecture, and the need for an

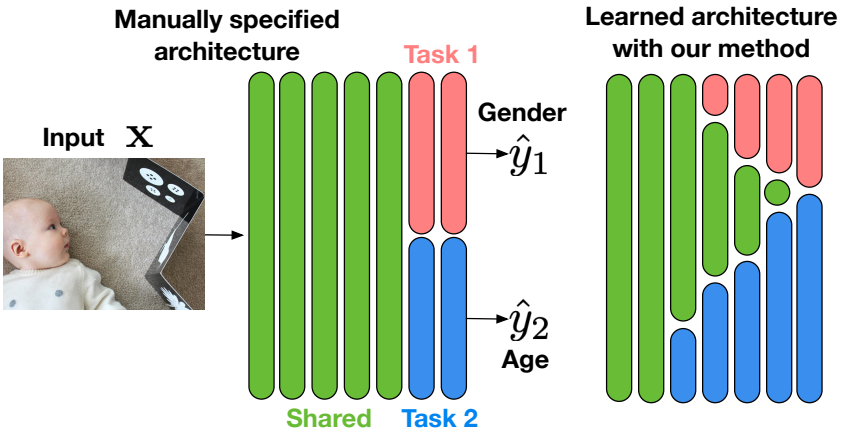


Figure 6.1: Figure on the left illustrates a typical multi-task architecture, while the figure on the right shows an example architecture that can be learned with our method. We propose *Stochastic Filter Groups*, a principled way to learn the assignment of convolution kernels to task-specific and shared groups.

effective automatic method to learn it from data.

In this paper, we propose *Stochastic Filter Groups* (SFGs); a probabilistic mechanism to learn the amount of task-specific and shared representations needed in each layer of MTL architectures (Fig. 6.1). Specifically, the SFGs learns to allocate kernels in each convolution layer into either "specialist" groups or a "shared" trunk, which are specific to or shared across different tasks, respectively (Fig. 6.2). The SFG equips the network with a mechanism to learn inter-layer connectivity and thus the structures of task-specific and shared representations. We cast the learning of SFG modules as a variational inference problem.

We evaluate the efficacy of SFGs on a variety of tasks. In particular, we focus on two multi-task learning problems: 1) age regression and gender classification from face images on UTKFace dataset [199] and 2) semantic regression (i.e. image synthesis) and semantic segmentation on a real-world medical imaging dataset, both of which require predictions over all pixels. Experiments show that our method achieves considerably higher prediction accuracy than baselines with no mechanism to learn connectivity structures, and either higher or comparable performance than a cross-stitch network [187], while being able to learn meaningful architectures automatically.

6.2 Related works

Our work is concerned with the goal of learning where to share neural network components across different tasks to maximise the benefit of MTL. The main challenge of such methods lies in designing a mechanism that determines how and where to share weights within the network. There are broadly two categories of methods that determine the nature of weight sharing in MTL networks.

The first category is composed of methods that directly optimise the sharing of weights in order to maximise task-wise performance. These methods set out to learn a set a vectors that control which features are shared within a layer and how these are distributed across [200, 195, 187, 194]. They start with a baseline CNN architecture where they learn additional connections and pathways that define the final MTL model. For instance, Cross-Stitch networks [187] control the degree of weight sharing at each convolution layer whilst Soft-Layer Ordering [195] goes beyond the assumption of parallel ordering of feature hierarchies to allow features to mix at different layers depending on the task. In contrast, the second group of MTL methods focuses on weight clustering based on task-similarity [201, 202, 203, 204, 205]. For example,

[204] employed a greedy, iterative algorithm to grow a tree-like deep architecture that clusters similar tasks hierarchically or [205] which determines the degree of weight sharing based on statistical dependency between tasks.

Our method falls into first category, and differentiates itself by performing "hard' partitioning of task-specific and shared features. By contrast, prior methods are based on "soft" sharing of features [187, 194] or weights [200, 195]. These methods generally learn a set of mixing coefficients that determine the weighted sum of features throughout the network, which does not impose connectivity structures on the architecture. On the other hand, our method learns a distribution over the connectivity of layers by grouping kernels. This allows our model to learn meaningful grouping of task-specific and shared features as illustrated in Fig. 6.6.

6.3 Methods

We introduce a new approach for determining where to learn task-specific and shared representation in multi-task CNN architectures. We propose *stochastic filter groups* (SFG), a probabilistic mechanism to partition kernels in each convolution layer into "specialist" groups or a "shared" group, which are specific to or shared across different tasks, respectively. We employ variational inference to learn the distributions over the possible grouping of kernels and network parameters that determines the connectivity between layers and the shared and task-specific features. This naturally results in a learning algorithm that optimally allocate representation capacity across multi-tasks via gradient-based stochastic optimization, e.g. stochastic gradient descent.

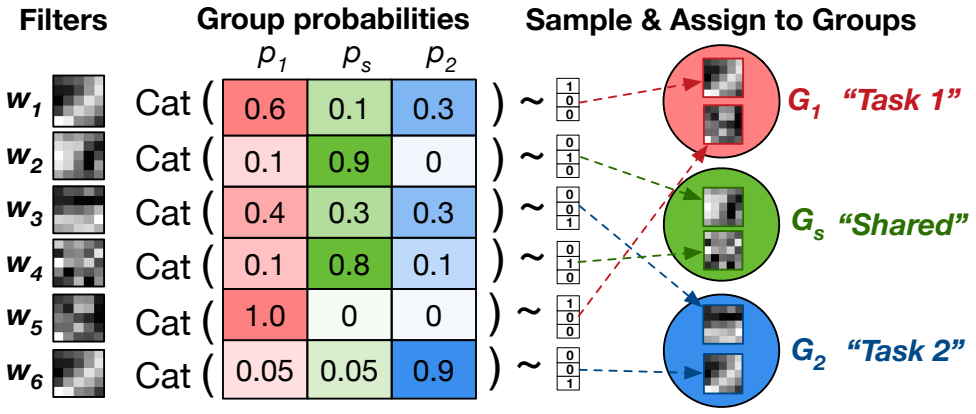


Figure 6.2: Illustration of filter assignment in a SFG module. Each kernel $\{\mathbf{w}_k\}$ in the given convolution layer is probabilistically assigned to one of the filter groups G_1, G_s, G_2 according to the sample drawn from the associated categorical distribution $\text{Cat}(p_1, p_s, p_2)$.

6.3.1 Stochastic Filter Groups

SFGs introduce a sparse connection structure into the architecture of CNN for multi-task learning in order to separate features into task-specific and shared components. Ioannou et al. [206] introduced *filter groups* to partition kernels in each convolution layer into groups, each of which acts only on a subset of the preceding features, and demonstrated that such sparsity reduces computational cost and number of parameters without compromising accuracy. Here we adapt the concept of filter groups to the multi-task learning paradigm and propose an extension with an additional mechanism for learning an optimal kernel grouping rather than pre-specifying them.

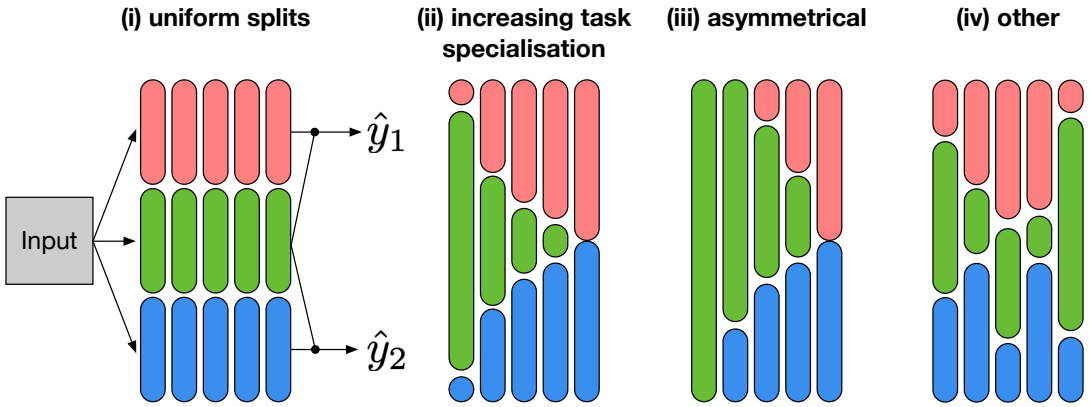


Figure 6.3: Illustration of possible grouping patterns learnable with the proposed method. Each set of green, pink and yellow blocks represent the ratio of filter groups G_1 (red), G_s (green) and G_2 (blue). (i) denotes the case where all kernels are uniformly split. (ii) & (iii) are the cases where the convolution kernels become more task-specific at deeper layers. (iv) shows an example with more heterogeneous splits across tasks.

For simplicity, we describe SFGs for the case of multitask learning with two tasks, but can be trivially extended to a larger number of tasks. At the l^{th} convolution layer in a CNN architecture with K_l kernels $\{\mathbf{w}^{(l),k}\}_{k=1}^{K_l}$, the associated SFG performs two operations:

1. **Filter Assignment:** each kernel $\mathbf{w}_k^{(l)}$ is stochastically assigned to either: i) the "task-1 specific group" $G_1^{(l)}$, ii) "shared group" $G_s^{(l)}$ or iii) "task-2 specific group" $G_2^{(l)}$ with respective probabilities $\mathbf{p}^{(l),k} = [p_1^{(l),k}, p_s^{(l),k}, p_2^{(l),k}] \in [0,1]^3$. Convolving with the respecitve filter groups yields distinct sets of features $F_1^{(l)}, F_s^{(l)}, F_2^{(l)}$. Fig. 6.2 illustrates this operation and Fig. 6.3 shows different learnable patterns.

2. **Feature Routing:** as shown in Fig. 6.4 (i), the features $F_1^{(l)}, F_s^{(l)}, F_2^{(l)}$ are routed to the filter groups $G_1^{(l+1)}, G_s^{(l+1)}, G_2^{(l+1)}$ in the subsequent $(l+1)^{\text{th}}$ layer in such a way to respect the task-specificity and sharedness of filter groups in the l^{th} layer. Specifically, we perform the following routing for $l > 0$:

$$
\begin{aligned}
F_1^{(l+1)} &= h^{(l+1)}\big([F_1^{(l)}|F_s^{(l)}] * G_1^{(l+1)}\big) \\
F_s^{(l+1)} &= h^{(l+1)}\big(F_s^{(l)} * G_s^{(l+1)}\big) \\
F_2^{(l+1)} &= h^{(l+1)}\big([F_2^{(l)}|F_s^{(l)}] * G_2^{(l+1)}\big)
\end{aligned}
$$

   where each $h^{(l+1)}$ defines the choice of non-linear function, $*$ denotes convolution operation and $|$ denotes a merging operation of arrays (e.g. concatenation). At $l = 0$, input image $\mathbf{x}$ is simply convolved with the first set of filter groups to yield $F_i^{(1)} = h^{(1)}\big(\mathbf{x} * G_i^{(1)}\big), i \in \{1, 2, s\}$. Fig. 6.4(ii) shows that such sparse connectivity ensures the parameters of $G_1^{(l)}$ and $G_2^{(l)}$ are only learned based on the respective task losses, while $G_s^{(l)}$ is optimised based on both tasks.

Fig. 6.5 provides a schematic of our overall architecture, in which each SFG module stochastically generates filter groups in each convolution layer and the resultant features are sparsely routed as described above. The merging modules, denoted as black circles, combine the task-specific and shared features appropriately, i.e. $[F_i^{(l)}|F_s^{(l)}], i = 1, 2$ and pass them to the filter groups in the next layer. Each white circle denotes the presence of additional transformations

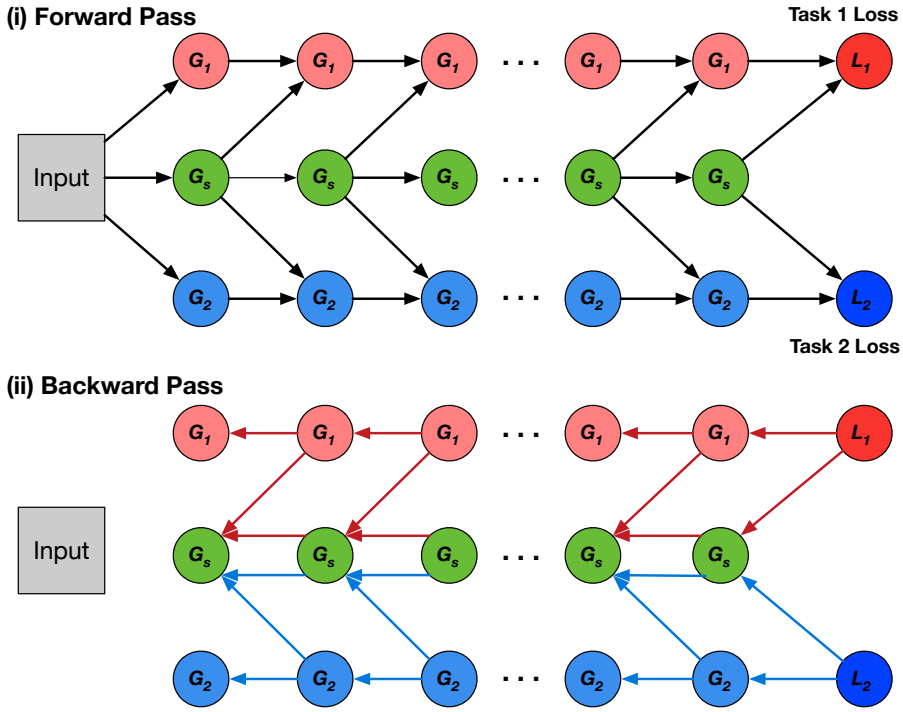


Figure 6.4: Illustration of feature routing. The circles G_1, G_s, G_2 denote the task-specific and shared filter groups in each layer. (i) shows the directions of routing of activations between different filter groups while (ii) shows the directions of the gradient flow from the task losses L_1 and L_2. The red and blue arrows denote the gradients that step from L_1 and L_2, respectively. The task-specific groups G_1, G_2 are only updated based on the associated losses, while the shared group G_s is updated based on both.

(e.g. convolutions or fully connected layers) in each $h^{(l+1)}$, performed on top of the standard non-linearity (e.g. ReLU).

The proposed sparse connectivity is integral to ensure task performance and structured representations. In particular, one might argue that the routing of "shared" features $F_s^{(l)}$ to the respective "task-specific" filter groups $G_1^{(l+1)}$ and $G_2^{(l+1)}$ is not necessary to ensure the separation of gradients across the task losses. However, this connection allows for learning more complex task-specific features at deeper layers in the network. For example, without this routing, having a large proportion of "shared" filter group G_s at the first layer (Fig. 6.3 (ii)) substantially reduces the amount of features available for learning task-specific kernels in the subsequent layers—in the extreme case in which all kernels in one layer are assigned to G_s, the task-specific filter groups in the subsequent layers are effectively unused.

Another important aspect that needs to be highlighted is the varying dimensionality of feature maps. Specifically, the number of kernels in the respective filter groups $G_1^{(l)}, G_s^{(l)}, G_2^{(l)}$ can vary at each iteration of the training, and thus, so does the depth of the resultant feature maps $F_1^{(l)}, F_s^{(l)}, F_2^{(l)}$. Instead of directly working with features maps of varying size, we implement the proposed architecture by defining $F_1^{(l)}, F_s^{(l)}, F_2^{(l)}$ as sparse tensors. At each SFG module, we first convolve the input features with all kernels, and generate the output features from each filter group by zeroing out the channels that root from the kernels in the other groups, resulting in $F_1^{(l)}, F_s^{(l)}, F_2^{(l)}$ that are sparse at non-overlapping channel indices. In the simplest form with no additional transformation (i.e. the grey circles in Fig. 6.5 are identity functions), we define the merging operation $[F_i^{(l)}|F_s^{(l)}], i = 1, 2$ as pixel-wise summation. In the presence of more complex transforms (e.g. residual blocks), we concatenate the output features in the channel-axis and

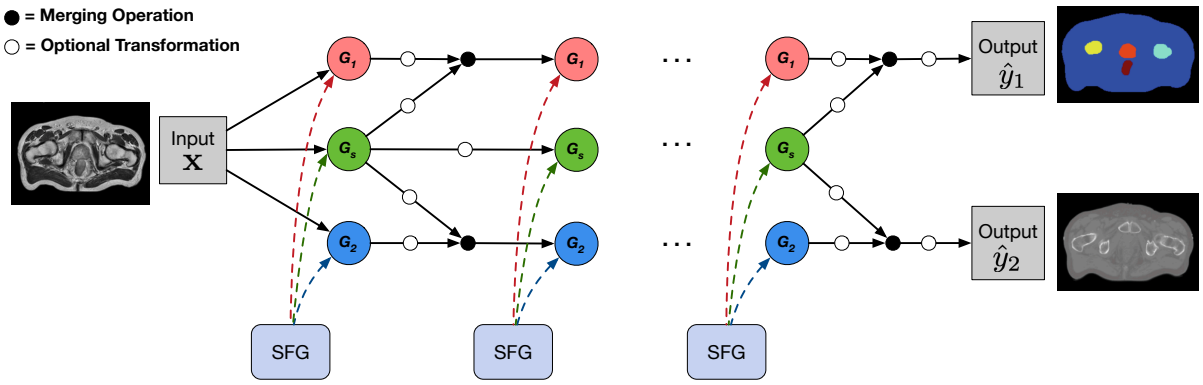


Figure 6.5: Schematic of the proposed multi-task architecture based on a series of SFG modules in the presence of two tasks. At each convolution layer, kernels are stochastically assigned to task-specific and shared filter groups G_1, G_s, G_2. Each input image is first convolved with the respective filter groups to yield three distinct sets of output activations, which are routed sparsely to the filter groups in the second layer layer. This process repeats in the remaining SFG modules in the architecture until the last layer where the outputs of the final SFG module are combined into task-specific predictions $\hat{y}_1$ and $\hat{y}_2$. Each small white circle denotes an optional transformation (e.g. extra convolutions) and black circle merges the incoming inputs (e.g. concatenation).

perform a 1x1 convolution to ensure the number of channels in $[F_i^{(l)}|F_s^{(l)}]$ is the same as in $F_s^{(l)}$.

6.3.2 T+1 Way "Drop-Out"

Here we derive the method for simultaneously optimising the CNN parameters and grouping probabilities. We achieve this by extending the variational interpretation of binary dropout [207, 107] to the $(T+1)$-way assignment of each convolution kernel to the filter groups where T is the number of tasks. As before, we consider the case $T = 2$.

Suppose that the architecture consists of L SFG modules, each with K_l kernels where l is the index. As the posterior distribution over the convolution kernels in SFG modules $p(\mathcal{W}|\mathbf{X}, \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)})$ is intractable, we approximate it with a simpler distribution $q_\phi(\mathcal{W})$ where $\mathcal{W} = \{\mathbf{w}^{(l),k}\}_{k=1,\ldots,K_l, l=1,\ldots,L}$. Assuming that the posterior distribution factorizes over layers and kernels up to group assignment, we defined the variational distribution as:

$$\begin{aligned} q_\phi(\mathcal{W}) &= \prod_{l=1}^{L}\prod_{k=1}^{K_l} q_{\phi_{lk}}(\mathbf{w}^{(l),k}) \\ &= \prod_{l=1}^{L}\prod_{k=1}^{K_l} q_{\phi_{lk}}(\mathbf{w}_1^{(l),k}, \mathbf{w}_s^{(l),k}, \mathbf{w}_2^{(l),k}) \end{aligned}$$

where $\{\mathbf{w}_1^{(l),k}, \mathbf{w}_s^{(l),k}, \mathbf{w}_2^{(l),k}\}$ denotes the k^{th} kernel in l^{th} convolution layer after being routed into task-specific $G_1^{(l)}, G_2^{(l)}$ and shared group $G_s^{(l)}$. We define each $q_{\phi_{lk}}(\mathbf{w}_1^{(l),k}, \mathbf{w}_2^{(l),k}, \mathbf{w}_s^{(l),k})$ as:

$$\mathbf{w}_i^{(l),k} = z_i^{(l),k} \cdot \mathbf{w}^{(l),k} \quad \text{for } i \in \{1, s, 2\} \tag{6.1}$$

$$\mathbf{z}^{(l),k} = [z_1^{(l),k}, z_2^{(l),k}, z_s^{(l),k}] \sim \text{Cat}(\mathbf{p}^{(l),k}) \tag{6.2}$$

where $\mathbf{z}^{(l),k}$ is the one-hot encoding of a sample from the categorical distribution over filter group assignments. The variational parameters ϕ_{lk} consists of the pre-grouping convolution kernel $\mathbf{w}^{(l),k}$ and the grouping probabilities $\mathbf{p}^{(l),k} = [p_1^{(l),k}, p_s^{(l),k}, p_2^{(l),k}]$.

We minimize the KL divergence between the approximate posterior $q_\phi(\mathcal{W})$ and $p(\mathcal{W}|\mathbf{X}, \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)})$. Assuming that the joint likelihood over the two tasks factorizes, we have the following optimization objective:

$$\mathcal{L}_{\text{MC}}(\phi) = -\frac{N}{M}\sum_{i=1}^{M}\Big[\log p(y_i^{(1)}|\mathbf{x}_i, \mathcal{W}_i) + \log p(y_i^{(2)}|\mathbf{x}_i, \mathcal{W}_i)\Big] + \sum_{l=1}^{L}\sum_{k=1}^{K_l}\text{KL}(q_{\phi_{lk}}(\mathbf{W}^{(l),k})||p(\mathbf{W}^{(l),k})) \quad (6.3)$$

where M is the size of the mini-batch, N is the total number of training data points, and $\mathcal{W}_i$ denotes a set of model parameters sampled from $q_\phi(\mathcal{W})$. The last KL term regularizes the deviation of the approximate posterior from the prior $p(\mathbf{W}^{(l),k}) = \mathcal{N}(0, \mathbf{I}/l^2)$ where $l > 0$. Adapting the approximation presented in [207] to our scenario, we obtain:

$$\text{KL}(q_{\phi_{lk}}(\mathbf{C}^{(l),k})||p(\mathbf{C}^{(l),k})) \propto \frac{l^2}{2}||\mathbf{M}^{(l),k}||_2^2 - \mathcal{H}(\mathbf{p}^{(l),k}) \quad (6.4)$$

where $\mathcal{H}(\mathbf{p}^{(l),k}) = -\sum_{i\in\{1,2,s\}} p_i^{(l),k}\log\, p_i^{(l),k}$ is the entropy of the grouping probabilities. While the first term performs the L2-weight norm, the second term pulls the grouping probabilities towards the uniform distribution. Plugging eq.(6.4) into eq.(6.3) yields the overall loss:

$$\mathcal{L}_{\text{MC}}(\phi) = -\frac{N}{M}\sum_{i=1}^{M}\Big[\log p\Big(y_i^{(1)}|\mathbf{x}_i, \mathcal{W}_i\Big) + \log p\Big(y_i^{(2)}|\mathbf{x}_i, \mathcal{W}_i\Big)\Big] + \lambda_1 \cdot \sum_{l=1}^{L}\sum_{k=1}^{K_l}||\mathbf{w}^{(l),k}||^2 - \lambda_2 \cdot \sum_{l=1}^{L}\sum_{k=1}^{K_l}\mathcal{H}(\mathbf{p}^{(l),k}) \quad (6.5)$$

where $\lambda_1 > 0, \lambda_2 > 0$ are regularization coefficients.

We note that the discrete sampling operation during filter group assignment (eq. (6.2)) creates discontinuities, giving the first term in the objective function (eq. 6.5) zero gradient with respect to the grouping probabilities $\{\mathbf{p}^{(l),k}\}$. We therefore, as employed in [179] for the binary case, approximate each of the categorical variables $\text{Cat}(\mathbf{p}^{(l),k})$ by the Gumbel-Softmax distribution, $\text{GSM}(\mathbf{p}^{(l),k}, \tau)$ [208, 209], a continuous relaxation which allows for sampling, differentiable with respect to the parameters $\mathbf{p}^{(l),k}$ through a reparametrisation trick. The temperature term τ adjusts the bias-variance tradeoff of gradient approximation; as the value of τ approaches 0, samples from the GSM distribution become one-hot (i.e. lower bias) while the variance of the gradients increases. In practice, we start at a high τ and anneal to a small but non-zero value as in [209, 107] as detailed in supplementary materials.

6.4 Experiments

We tested *stochastic filter groups* (SFG) on two multi-task learning (MTL) problems: 1) age regression and gender classification from face images on UTKFace dataset [199] and 2) semantic image regression (synthesis) and segmentation on a medical imaging dataset.

UTKFace dataset: We tested our method on UTKFace [199], which consists of 23,703 cropped faced images in the wild with labels for age and gender. We created a dataset with a 70/15/15% split. We created a secondary separate dataset containing only 10% of images from the initial set, so as to simulate a data-starved scenario.

Medical imaging dataset: We used a medical imaging dataset to evaluate our method in a real-world, multi-task problem where paucity of data is common and hard to mitigate. The goal of radiotherapy treatment planning is to maximise radiation dose to the tumour whilst minimising dose to the organs. To plan dose delivery, a Computed Tomography (CT) scan is needed as CT voxel intensity scales with tissue density, thus allowing dose propagation simulations. An MRI scan is needed to segment the surrounding organs. Instead of acquiring both an MRI and a CT, algorithms can be used to synthesise a CT scan (task 1) and segment organs (task 2) given a single input MRI scan. For this experiment, we acquired 15 3D prostate cancer scans with respective CT and MRI scans with semantic 3D labels for organs (prostate, bladder, rectum and left/right femur heads) obtained from a trained radiologist. We created a training set of 10 patients, with the remaining 5 used for testing. We trained our networks on 2D subimages of size 128x128 randomly sampled from axial slices, and reconstructed the 3D volumes of size 288x288x62 at test time by stitching together the subimage-wise predictions.

6.4.1 Baselines

We compared our model against four baselines in addition to Cross-Stitch networks [187] trained end-to-end rather than sequentially for fair comparison. The four baselines considered are: 1) single-task networks, 2) hard-parameter sharing multi-task network (MT-hard sharing), 3) SFG-networks with constant $1/3$ allocated grouping (MT-constant mask) as *per* Fig. 6.3(i), and 4) SFG-networks with constant grouping probabilities (MT-constant $\mathbf{p}$). We train all the baselines in an end-to-end fashion for all the experiments.

We note that all four baselines can be considered special cases of an SFG-network. Two *single-task networks* can be learned when the shared grouping probability of kernels is set to zero. Considering Fig. 6.5, this would remove the diagonal connections and the shared network. This may be important when faced with two unrelated tasks which share no contextual information. A *hard-parameter sharing network* exists when all shared grouping probabilities are maximised to one leading to a scenario where all features are shared within the network up until the task-specific layers. The *MT-constant mask network* is illustrated in Fig. 6.3(i), where $1/3$ of kernels are allocated to the task 1, task 2 and shared groups, yielding uniform splits across layers. This occurs when an equal number of kernels in each layer obtain probabilities of $\mathbf{p}^{(l),k} = [1, 0, 0], [0, 1, 0]$ and $[0, 0, 1]$. Lastly, the *MT-constant* $\boldsymbol{p}$ model represents the situation where the grouping is non-informative and each kernel has equal probability of being specific or shared with probability $\mathbf{p}^{(l),k} = [1/3, 1/3, 1/3]$. Training details for these models, including the hyper-parameter settings, are provided in the supplementary document.

UTKFace network: We used VGG-11 CNN architecture [210] for age and gender prediction. The network consists of a series of 3x3 convolutional layers interleaved with max pooling layers. In contrast to the original architecture, we replaced the final max pooling and fully connected layers with global average pooling (GAP) followed by a fully connected layers for prediction. Our model's version of VGG (SFG-VGG) replaces each convolutional layer in VGG-11 with a SFG layer with max pooling applied to each feature map $F_1^{(l)}$, $F_2^{(l)}$, $F_s^{(l)}$. We applied GAP to each final feature map before the final merging operation and two fully connected layers for each task.

Medical imaging network: We used a high-resolution network architecture (HighResNet) [211] for CT synthesis and organ segmentation. This network has been successfully developed for semantic segmentation in medical imaging and has been used in a variety of medical applications such as CT synthesis [192, 212], brain segmentation [211] and tumour segmentation [213]. It consists of a series of residual blocks, which group two 3x3 convolutional layers with dilated

convolutions. The baseline network is composed of a 3x3 convolutional layer followed by three sets of twice repeated residual blocks with dilated convolutions using factors $d = [1, 2, 4]$. There is a 3x3 convolutional layer between each set of repeated residual blocks. The network ends with two final 3x3 layers and either one or two 1x1 convolutional layers for single and multi-task predictions. In our model, we replace each convolutional layer with an SFG module. After the first SFG layer, three distinct repeated residual blocks are applied to $F_1^{(l=0)}$, $F_2^{(l=0)}$, $F_s^{(l=0)}$. These are then merged according the feature routing methodology followed by a new SFG-layer and subsequent residual layers. Our model concludes with 2 successive SFG-layers followed by 1x1 convolutional layers applied to the merged features $F_1^{(l=L)}$ and $F_2^{(l=L)}$.

(a) Full training data

Method	Age (MAE)	Gender (Accuracy)
One-task (VGG11) [210]	7.32	90.70
MT-hard sharing	7.92	90.60
MT-constant mask	7.67	89.41
MT-constant $\mathbf{p}$=[1/3,1/3,1/3]	6.34	92.10
VGG11 Cross Stitch [187]	6.78	90.30
MT-SFG (ours)	**6.00**	**92.46**

(b) Small training data

Method	Age (MAE)	Gender (Accuracy)
One-task (VGG11) [210]	8.79	85.54
MT-hard sharing	9.19	85.83
MT-constant mask	9.02	85.98
MT-constant $\mathbf{p}$=[1/3,1/3,1/3]	9.15	86.01
VGG11 Cross Stitch [187]	8.85	83.72
MT-SFG (ours)	**8.54**	**87.01**

Table 6.1: Age regression and gender classification results on UTKFace [199] with (a) the full and (b) limited training set. The best and the second best results are shown in red and blue. The mean absolute error (MAE) is reported for the age prediction and classification accuracy for gender prediction. For our model, we performed 50 stochastic forward passes at test time by sampling the kernels from the approximate posterior $q_\phi(\mathcal{W})$. We calculated the average age per subject and obtained gender prediction using the mode of the test-time predictions. We initialised our model with grouping probabilities $\mathbf{p}$=[0.2, 0.6, 0.2] for all convolution kernels.

6.5 Results

6.5.1 Age regression and gender prediction

Results on age prediction and gender classification on both datasets are presented in Tab. 6.1a and 6.1b. Our model (MT-SFG) achieved the best performance in comparison to the baselines in both data regimes. In both sets of experiments, our model outperformed the hard-parameter sharing (*MT-hard sharing*) and constant allocation (*MT-constant mask*). This demonstrates the advantage of learning to allocate kernels. In the *MT-constant mask* model, kernels are equally allocated across groups. In contrast, our model is able to allocate kernels in varying proportions across different layers in the network (Fig. 6.7 - SFG-VGG11) to maximise inductive transfer. Moreover, our methods performed better than a model with constant, non-informative grouping probabilities (*MT-constant* $\mathbf{p}$= $[1/3, 1/3, 1/3]$), displaying the importance of learning structured representations and connectivity across layers to yield good predictions.

6.5.2 Image regression and semantic segmentation

Results on CT image synthesis and organ segmentation from input MRI scans is detailed in Tab. 6.2. Our method obtains equivalent (non-statistically significant different) results to the Cross-Stitch network [187] on both tasks. We have, however, observed best synthesis performance in the bone regions (femur heads and pelvic bone region) in our model when compared

(a) CT Synthesis (PSNR)

Method	Overall	Bones	Organs	Prostate	Bladder	Rectum
One-task (HighResNet) [211]	25.76 (0.80)	30.35 (0.58)	38.04 (0.94)	51.38 (0.79)	33.34 (0.83)	34.19 (0.31)
MT-hard sharing	26.31 (0.76)	31.25 (0.61)	39.19 (0.98)	52.93 (0.95)	34.12 (0.82)	34.15 (0.30)
MT-constant mask	24.43(0.57)	29.10(0.46)	37.24(0.86)	50.48(0.73)	32.29(1.01)	33.44(2.88)
MT-constant $\mathbf{p}$=[1/3,1/3,1/3]	26.64(0.54)	31.05 (0.55)	39.11 (1.00)	53.20 (0.86)	34.34 (1.35)	35.61 (0.35)
Cross Stitch [187]	27.86 (1.05)	32.27 (0.55)	40.45 (1.27)	54.51 (1.01)	36.81 (0.92)	36.35 (0.38)
MT-SFG (ours)	27.74 (0.96)	32.29 (0.59)	39.93 (1.09)	53.01 (1.06)	35.65 (0.44)	35.65 (0.37)

(b) Segmentation (DICE)

Method	Overall	Left Femur Head	Right Femur Head	Prostate	Bladder	Rectum
One-task (HighResNet) [211]	0.848(0.024)	0.931 (0.012)	0.917 (0.013)	0.913 (0.013)	0.739 (0.060)	0.741 (0.011)
MT-hard sharing	0.829(0.023)	0.933 (0.009)	0.889 (0.044)	0.904 (0.016)	0.685 (0.036)	0.732 (0.014)
MT-constant mask	0.774(0.065)	0.908 (0.012)	0.911 (0.015)	0.806 (0.0541)	0.583 (0.178)	0.662 (0.019)
MT-constant $\mathbf{p}$=[1/3,1/3,1/3]	0.752(0.056)	0.917 (0.004)	0.917 (0.01)	0.729 (0.086)	0.560 (0.180)	0.639 (0.012)
Cross Stitch [187]	0.854 (0.036)	0.923 (0.008)	0.915 (0.013)	0.933 (0.009)	0.761 (0.053)	0.737 (0.015)
MT-SFG (ours)	0.852(0.047)	0.935 (0.007)	0.912 (0.013)	0.923 (0.016)	0.750 (0.062)	0.758 (0.011)

Table 6.2: Performance on the medical imaging dataset with best results in red, and the second best results in blue. The PSNR is reported for the CT-synthesis (synCT) across the whole volume (overall), at the bone regions, across all organ labels and individually at the prostate, bladder and rectum. For the segmentation, the average DICE score per patient across all semantic labels is computed. The standard deviations are computed over the test subject cohort. For our model, we perform 50 stochastic forward passes at test-time by sampling the kernels from the approximated posterior distribution $q_\phi(\mathcal{W})$. We compute the average of all passes to obtain the synCT and calculate the mode of the segmentation labels for the final segmentation. We initialised our model with grouping probabilities $\mathbf{p}$=[0.2, 0.6, 0.2]. Red cells indicate best performing and blue cells indicate second best models.

against all the baselines, including Cross-Stitch. The bone voxel intensities are the most difficult to synthesise from an input MR scan as task uncertainty in the MR to CT mapping at the bone is often highest [192]. Our model was able to disentangle features specific to the bone intensity mapping (Fig. 6.6) without supervision of the pelvic location, which allowed it to learn a more accurate mapping of an intrinsically difficult task.

6.5.3 Learned architectures

Analysis of the grouping probabilities of a network embedded with SFG modules permits visualisation of the network connectivity and thus the learned MTL architecture. To analyse the group allocation of kernels at each layer, we computed the sum of class-wise probabilities per layer. Learned groupings for both SFG-VGG11 network trained on UTKFace and the SFG-HighResNet network trained on prostate scans are presented in Fig. 6.7. These figures illustrate increasing task specialisation in the kernels with network depth. At the first layer, all kernels are classified as shared ($\mathbf{p}$= $[0, 1, 0]$) as low-order features such as edge or contrast descriptors are generally learned earlier layers. In deeper layers, higher-order representations are learned, which describe various salient features specific to the tasks. This coincides with our network allocating kernels as task specific, as illustrated in Fig. 6.6, where activations are stratified by allocated class per layer. Density plots of the learned kernel probabilities and trajectory maps displaying training dynamics, along with more examples of feature visualisations, are provided in supplementary materials.

Notably, the learned connectivity of both models shows striking similarities to hard-parameter sharing architectures commonly used in MTL. Generally, there is a set of shared layers, which

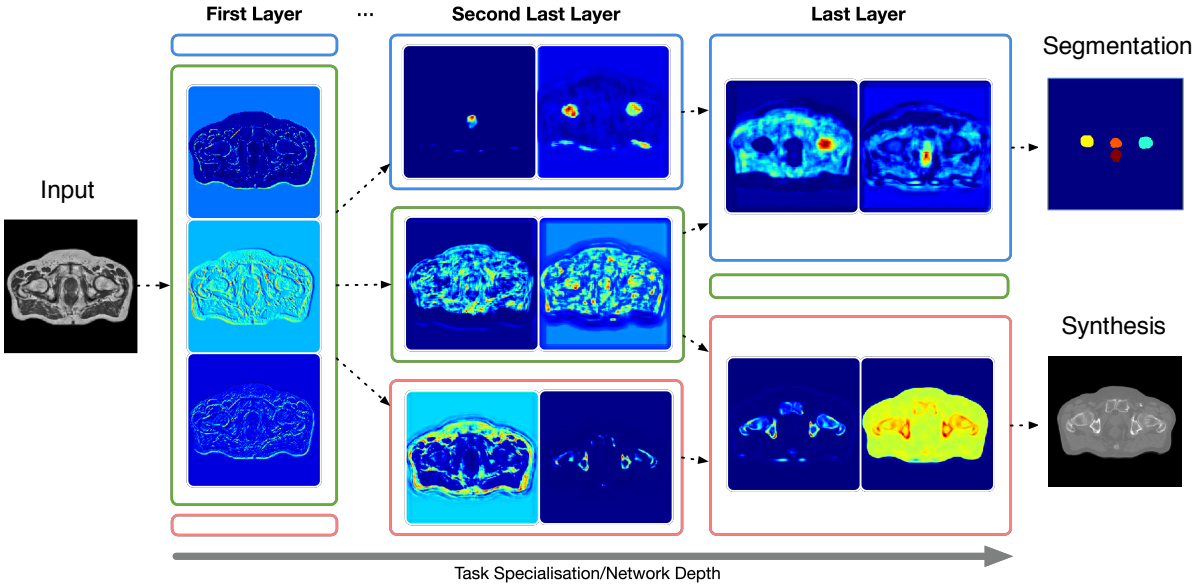


Figure 6.6: Activation maps from example kernels in the learned task-specific and shared filter groups, $G_1^{(l)}, G_2^{(l)}, G_s^{(l)}$ (enclosed in blue, green and pink funnels) in the first, the second last and the last convolution layers in the SFG-HighResNet model trained on the medical imaging dataset. The results from convolution kernels with low entropy (i.e. high "confidence") of group assignment probabilities $\mathbf{p}^{(l)}$ are shown for the respective layers.

aim to learn a feature set common to both tasks. Task-specific branches then learn a mapping from this feature space for task-specific predictions. Our models are able to automatically learn this structure whilst allowing asymmetric allocation of task-specific kernels with no priors on the network structure.

6.5.4 Effect of p initialisation

Fig. 6.3 shows the layer-wise proportion of the learned kernel groups on the UTKFace dataset for four different initilization schemes of grouping probabilities $\mathbf{p}$: (i) "dominantly shared", with $\mathbf{p} = [0.2, 0.6, 0.2]$, (ii) "dominantly task-specific", with $\mathbf{p} = [0.45, 0.1, 0.45]$, (iii) "random", where $\mathbf{p}$ is drawn from Dirichlet$(1, 1, 1)$, (iv) "start with MT-constant mask", where an equal number of kernels in each layer are set to probabilities of $\mathbf{p} = [1, 0, 0], [0, 1, 0]$ and $[0, 0, 1]$. In all cases, the same set of hyper-parameters, including the annealing rate of the temperature term in GSM approximation and the coefficient of the entropy regularizer $\mathcal{H}(\mathbf{p})$, were used during training. We observe that the kernel grouping of respective layers in (i), (ii) and (iii) all converge to a very similar configuration observed in Sec. 6.5.3, highlighting the robustness of our method to different initialisations of $\mathbf{p}$. In case (iv), the learning of p were much slower than the remaining cases, due to weaker gradients, and we speculate that a higher entropy regularizer is necessary to facilitate its convergence.

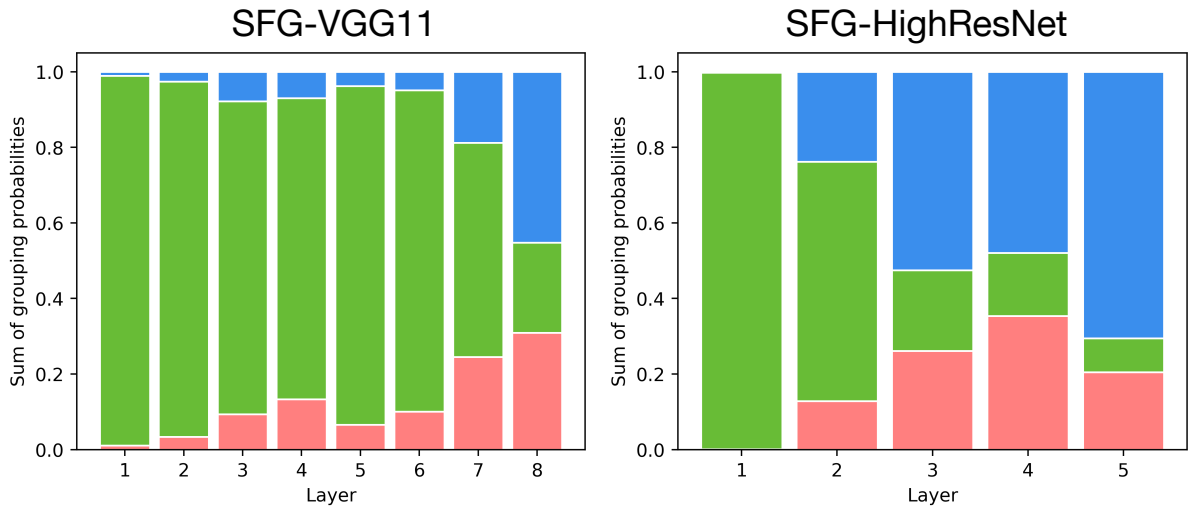


Figure 6.7: Learned kernel grouping in a) SFG-VGG11 network on UTKFace and b) SFG-HighResNet on medical scans. The proportions of task-1, shared and task-2 filter groups are shown in blue, green and pink. Within SFG-VGG11, task-1 age regression and task-2 is gender classification. For SFG-HighResNet, task-1 is CT synthesis and task-2 is organ segmentation.

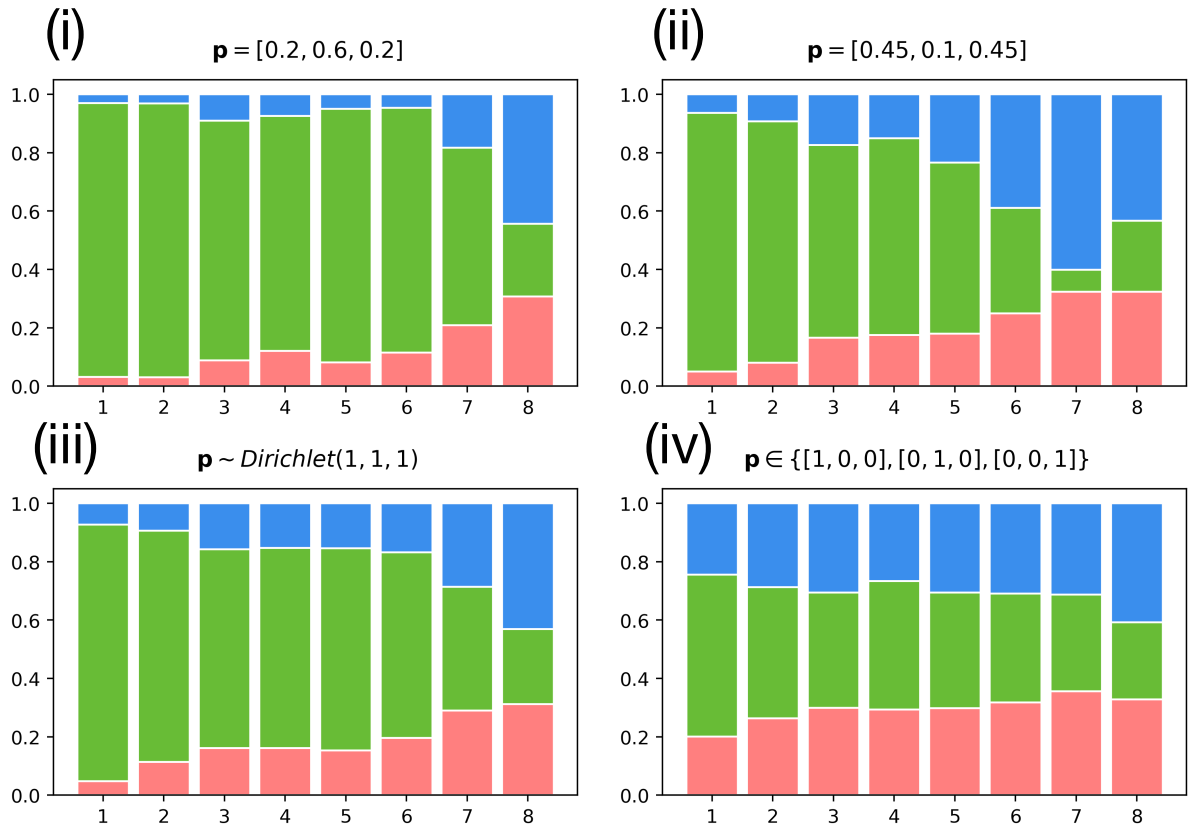


Figure 6.8: Effect of the initial values of grouping probabilities $\mathbf{p}$ on the learned kernel allocation after convergence.

6.6 Discussion

In this paper, we have proposed *stochastic filter groups* (SFGs) to disentangle *task-specific* and *generalist* features. SFGs probabilistically defines the grouping of kernels and thus the connectivity of features in a CNNs. We use variational inference to approximate the distribution over connectivity given training data and sample over possible architectures during training. Our method can be considered as a probabilistic form of multi-task architecture learning [214], as the learned posterior embodies the optimal MTL architecture given the data.

Our model learns structure in the representations. The learned shared (generalist) features may be exploited either in a transfer learning or continual learning scenario. As seen in [215], an effective prior learned from multiple tasks can be a powerful tool for learning new, unrelated tasks. Our model consequently offers the possibility to exploit the learned task-specific and generalist features when faced with situations where a third task is needed, which may suffer from unbalanced or limited training data. This is particularly relevant in the medical field, where training data is expensive to acquire as well as laborious. We will investigate this in further work.

Lastly, a network composed of SFG modules can be seen as a superset of numerous MTL architectures. Depending on the data and the analysed problem, SFGs can recover many different architectures such as single task networks, traditional hard-parameter sharing, equivalent allocation across tasks, and asymmetrical grouping (Fig. 6.3). Note, however, that proposed SFG module only learns connectivity between neighbouring layers. Non-parallel ordering of layers, a crucial concept of MTL models [195, 194], was not investigated. Future work will look to investigate the applicability of SFG modules for learning connections across grouped kernels between non-neighbouring layers.

Chapter 7

How to Combine Decision Trees and Neural Networks

Deep neural networks and decision trees operate on largely separate paradigms; typically, the former performs representation learning with pre-specified architectures, while the latter is characterised by learning hierarchies over pre-specified features with data-driven architectures. We unite the two via *adaptive neural trees* (ANTs) that incorporates representation learning into edges, routing functions and leaf nodes of a decision tree, along with a backpropagation-based training algorithm that adaptively grows the architecture from primitive modules (e.g., convolutional layers). We demonstrate that, whilst achieving competitive performance on classification and regression datasets, ANTs benefit from (i) lightweight inference via conditional computation, (ii) hierarchical separation of features useful to the task e.g. learning meaningful class associations, such as separating natural vs. man-made objects, and (iii) a mechanism to adapt the architecture to the size and complexity of the training dataset.

7.1 Introduction

Neural networks (NNs) and decision trees (DTs) are both powerful classes of machine learning models with proven successes in academic and commercial applications. The two approaches, however, typically come with mutually exclusive benefits and limitations.

NNs are characterised by learning hierarchical representations of data through the composition of nonlinear transformations [216, 217], which has alleviated the need for feature engineering, in contrast with many other machine learning models. In addition, NNs are trained with stochastic optimisers, such as stochastic gradient descent (SGD), allowing training to scale to large datasets. Consequently, with modern hardware, we can train NNs of many layers on large datasets, solving numerous problems ranging from object detection to speech recognition with unprecedented accuracy [218]. However, their architectures typically need to be designed by hand and fixed per task or dataset, requiring domain expertise [219]. Inference can also be heavy-weight for large models, as each sample engages every part of the network, i.e., increasing capacity causes a proportional increase in computation [220].

Alternatively, DTs are characterised by learning hierarchical clusters of data [12]. A DT learns how to split the input space, so that in each subset, linear models suffice to explain the data. In contrast to standard NNs, the architectures of DTs are optimised based on training data, and are particularly advantageous in data-scarce scenarios. DTs also enjoy lightweight inference as only a single root-to-leaf path on the tree is used for each input sample. However, successful applications of DTs often require hand-engineered features of data. We can ascribe the limited expressivity of single DTs to the common use of simplistic routing functions,

such as splitting on axis-aligned features. The loss function for optimising hard partitioning is non-differentiable, which hinders the use of gradient descent-based optimization and thus complex splitting functions. Current techniques for increasing capacity include ensemble methods such as random forests (RFs) [11] and gradient-boosted trees (GBTs) [221], which are known to achieve state-of-the-art performance in various tasks, including medical applications and financial forecasting [222, 223, 224, 225].

The goal of this work is to combine NNs and DTs to gain the complementary benefits of both approaches. To this end, we propose *adaptive neural trees* (ANTs), which generalise previous work that attempted the same unification [226, 227, 228, 229, 230, 231, 232] and address their limitations (see Tab. 7.1). ANTs represent routing decisions and root-to-leaf computational paths within the tree structures as NNs, which lets them benefit from hierarchical representation learning, rather than being restricted to partitioning the raw data space. On the other hand, unlike the fully distributed representaion of standard NN models, the tree topology of ANTs acts as a strong structural prior that enforces sparse structures by which features are shared and separated in a hierarchical fashion. In addition, we propose a backpropagation-based training algorithm to grow ANTs based on a series of decisions between making the ANT deeper—the central NN paradigm—or partitioning the data—the central DT paradigm (see Fig. 7.1 (Right)). This allows the architectures of ANTs to adapt to the data available. By our design, ANTs inherit the following desirable properties from both DTs and NNs:

- **Representation learning**: as each root-to-leaf path in an ANT is an NN, features can be learned end-to-end with gradient-based optimisation. Combined with the tree structure, an ANT can learn such features which are hierarchically shared and separated.

- **Architecture learning**: by progressively growing ANTs, the architecture adapts to the availability and complexity of data, embodying Occam's razor. The growth procedure can be viewed as architecture search with a hard constraint over the model class.

- **Lightweight inference**: at inference time, ANTs perform conditional computation, selecting a single root-to-leaf path on the tree on a per-sample basis, activating only a subset of the parameters of the model.

We empirically validate these benefits for regression and classification through experiments on the SARCOS [233], MNIST [234] and CIFAR-10 [235] datasets. The best performing methods on the SARCOS multivariate regression dataset are all tree-based, with ANTs achieving the lowest mean squared error. On the other hand, along with other forms of neural networks, ANTs far outperform state-of-the-art RF [236] and GBT [237] methods on image classification, with architectures achieving over 99% accuracy on MNIST and over 90% accuracy on CIFAR-10. Our ablations on all three datasets consistently show that the combination of feature learning and data partitioning are required for the best predictive performance of ANTs. In addition, we show that ANTs can learn meaningful hierarchical partitionings of data, e.g., grouping man-made and natural objects (see Fig. 7.2) useful to the end task. ANTs also have reduced time and memory requirements during inference, thanks to such hierarchical structure. In one case, we discover an architecture that achieves over 98% accuracy on MNIST using approximately the same number of parameters as a linear classifier on raw image pixels, showing the benefits of tree-shaped hierarchical sharing and separation of features in enhancing both computational and predictive performance. Finally, we demonstrate the benefits of architecture learning by training ANTs on subsets of CIFAR-10 of varying sizes. The method can construct architectures of adequate size, leading to better generalisation, particularly on small datasets.

Table 7.1: Comparison of tree-structured NNs. The first column denotes if each path on the tree is a NN, and the second column denotes if the routers learn features. The last column shows if the method grows an architecture, or uses a pre-specified one.

Method	**Feature learning?**		**Grown?**
	Path	Routers	
SDT [226]	✗	✗	✓
SDT 2 / HME [238]	✗	✓	✗
SDT 3 [227]	✗	✓	✓
SDT 4 [231]	✗	✓	✗
RDT [**?**]	✗	✓	✗
BT [242]	✗	✓	✓
Conv DT [228]	✗	✓	✗
NDT [229]	✗	✓	✓
NDT 2 [232]	✓	✓	✗
NDF [230]	✓	✓	✗
CNet [241]	✓	✓	✗
ANT (ours)	✓	✓	✓

7.2 Related work

Our work is primarily related to research into combining DTs and NNs. Here we explain how ANTs subsume a large body of such prior work as specific cases and address their limitations. We include additional reviews of work in conditional computation and neural architecture search in Sec.B in the supplementary material.

The very first soft decition tree (SDT) introduced in [226] is a specific case where in our terminology the routers are axis-aligned features, the transformers are identity functions, and the routers are static distributions over classes or linear functions. The hierarchical mixture of experts (HMEs) proposed by [238] is a variant of SDTs whose routers are linear classifiers and the tree structure is fixed; [**?**] recently proposed a more computationally efficient training method that is able to directly optimise hard-partitioning by differentiating through stochastic gradient estimators. More modern SDTs in [229, 228, 231] used multilayer perceptrons (MLPs) or convolutional layers in the routers to learn more complex partitionings of the input space. However, the simplicity of identity transformers used in these methods means that input data is never transformed and thus each path on the tree does not perform representation learning, limiting their performance.

More recent work suggested that integrating non-linear transformations of data into DTs would enhance model performance. The neural decision forest (NDF) [230], which held cutting-edge performance on ImageNet [239] in 2015, is an ensemble of DTs, each of which is also an instance of ANTs where the whole GoogLeNet architecture [240] (except for the last linear layer) is used as the root transformer, prior to learning tree-structured classifiers with linear routers. [232] employed a similar approach with a MLP at the root transformer, and is optimised to minimise a differentiable information gain loss. The conditional network proposed in [241] sparsified CNN architectures by distributing computations on hierarchical structures based on directed acyclic graphs with MLP-based routers, and designed models with the same accuracy with reduced compute cost and number of parameters. However, in all cases, the model architectures are pre-specified and fixed.

In contrast, ANTs satisfy all criteria in Tab. 7.1; they provide a general framework for learning tree-structured models with the capacity of representation learning along each path and within routing functions, and a mechanism for learning its architecture.

Architecture growth is a key facet of DTs [12], and typically performed in a greedy fashion

with a termination criteria based on validation set error [226, 227]. Previous works in DT research have made attempts to improve upon this greedy growth strategy. Decision jungles [243] employ a training mechanism to merge partitioned input spaces between different subtrees, and thus to rectify suboptimal "splits" made due to the locality of optimisation. [242] proposes budding trees, which are grown and pruned incrementally based on global optimisation of existing nodes. While our training algorithm, for simplicity, grows the architecture by greedily choosing the best option between going "deeper" and "splitting" the input space (see Fig. 7.1), it is certainly amenable to these advances.

7.3 Adaptive Neural Trees

We now formalise the definition of Adaptive Neural Trees (ANTs), which are a form of DTs enhanced with deep, learned representations. We focus on supervised learning, where the aim is to learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$ from a set of N labelled samples $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{y}^{(N)}) \in \mathcal{X} \times \mathcal{Y}$ as training data.

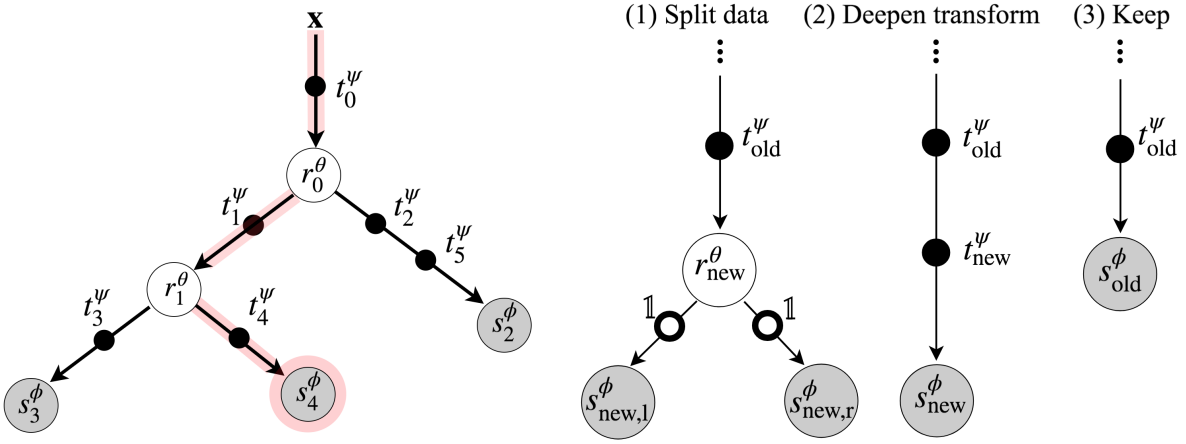


Figure 7.1: **(Left).** An example ANT. Data is passed through transformers (black circles on edges), routers (white circles on internal nodes), and solvers (gray circles on leaf nodes). The red shaded path shows routing of $\mathbf{x}$ to reach leaf node 4. Input $\mathbf{x}$ undergoes a series of selected transformations $\mathbf{x} \rightarrow \mathbf{x}_0^\psi := t_0^\psi(\mathbf{x}) \rightarrow \mathbf{x}_1^\psi := t_1^\psi(\mathbf{x}_0^\psi) \rightarrow \mathbf{x}_4^\psi := t_4^\psi(\mathbf{x}_1^\psi)$ and the solver module yields the predictive distribution $p_4^{\phi,\psi}(\mathbf{y}) := s_4^\phi(\mathbf{x}_4^\psi)$. The probability of selecting this path is given by $\pi_2^{\psi,\boldsymbol{\theta}}(\mathbf{x}) := r_0^{\boldsymbol{\theta}}(\mathbf{x}_0^\psi) \cdot (1 - r_1^{\boldsymbol{\theta}}(\mathbf{x}_1^\psi))$. **(Right).** Three growth options at a given node: *split data, deepen transform & keep.* The small white circles on the edges denote identity transformers.

7.3.1 Model Topology and Operations

In short, an ANT is a tree-structured model, characterized by a set of hierarchical partitions of the input space $\mathcal{X}$, a series of nonlinear transformations, and separate predictive models in the respective component regions. More formally, we define an ANT as a pair $(\mathbb{T}, \mathbb{O})$ where $\mathbb{T}$ defines the model topology, and $\mathbb{O}$ denotes the set of operations on it.

We restrict the model topology $\mathbb{T}$ to be instances of *binary trees*, defined as a set of graphs whose each node is either an internal node or a leaf, and is the child of exactly one parent node, except the root node at the top. We define the topology of a tree as $\mathbb{T} := \{\mathcal{N}, \mathcal{E}\}$ where $\mathcal{N}$ is the set of all nodes, and $\mathcal{E}$ is the set of edges between them. Nodes with no children are leaf nodes, $\mathcal{N}_{leaf}$, and all others are internal nodes, $\mathcal{N}_{int}$. Every internal node $j \in \mathcal{N}_{int}$ has exactly

Table 7.2: Primitive module specifications for MNIST, CIFAR-10 and SARCOS datasets. "conv5-40" denotes a 2D convolution with 40 kernels of spatial size 5×5. "GAP", "FC", "LC" and "LR" stand for global-average-pooling, fully connected layer, linear classifier and linear regressor. "Downsample Freq" denotes the frequency at which 2×2 max-pooling is applied.

Model	Router, $\mathcal{R}$	Transformer, $\mathcal{T}$	Solver, $\mathcal{S}$	Downsample Freq.
ANT-SARCOS	$1 \times$ FC +Sigmoid	$1 \times$FC+ tanh	LR	0
ANT-MNIST-A	$1 \times$ conv5-40 + GAP + $2 \times$FC +Sigmoid	$1 \times$ conv5-40 + ReLU	LC	1
ANT-MNIST-B	$1 \times$ conv3-40 + GAP + $2 \times$FC +Sigmoid	$1 \times$ conv3-40 +ReLU	LC	2
ANT-MNIST-C	$1 \times$ conv5-5 + GAP + $2 \times$FC +Sigmoid	$1 \times$ conv5-5+ReLU	LC	2
ANT-CIFAR10-A	$2 \times$ conv3-128 + GAP + $1 \times$FC +Sigmoid	$2 \times$ conv3-128 +ReLU	LC	1
ANT-CIFAR10-B	$2 \times$ conv3-96 + GAP + $1 \times$FC +Sigmoid	$2 \times$ conv3-96 +ReLU	LC	1
ANT-CIFAR10-C	$2 \times$ conv3-72 + GAP + $1 \times$FC +Sigmoid	$2 \times$ conv3-72 +ReLU	GAP + LC	1

two children nodes, represented by left(j) and right(j). Unlike standard trees, $\mathcal{E}$ contains an edge which connects input data $\mathbf{x}$ with the root node, as shown in Fig.7.1 (Left).

Every node and edge is assigned with operations which acts on the allocated samples of data (Fig.7.1). Starting at the root, each sample gets transformed and traverses the tree according to the set of operations $\mathbb{O}$. An ANT is constructed based on three primitive modules of differentiable operations:

1. **Routers**, $\mathcal{R}$: each internal node $j \in \mathcal{N}_{int}$ holds a *router* module, $r_j^{\boldsymbol{\theta}} : \mathcal{X}_j \rightarrow [0, 1] \in \mathcal{R}$, parametrised by $\boldsymbol{\theta}$, which sends samples from the incoming edge to either the left or right child. Here $\mathcal{X}_j$ denotes the representation at node j. We use *stochastic routing*, where the decision (1 for the left and 0 for the right branch) is sampled from Bernoulli distribution with mean $r_j^{\boldsymbol{\theta}}(\mathbf{x}_j)$ for input $\mathbf{x}_j \in \mathcal{X}_j$. As an example, $r_j^{\boldsymbol{\theta}}$ can be defined as a small CNN.

2. **Transformers**, $\mathcal{T}$: every edge $e \in \mathcal{E}$ of the tree has one or a composition of multiple *transformer* module(s). Each transformer $t_e^{\boldsymbol{\psi}} \in \mathcal{T}$ is a nonlinear function, parametrised by $\boldsymbol{\psi}$, that transforms samples from the previous module and passes them to the next one. For example, $t_e^{\boldsymbol{\psi}}$ can be a single convolutional layer followed by ReLU [244]. Unlike in standard DTs, edges transform data and are allowed to "grow" by adding more operations (Sec. 7.4), learning "deeper" representations as needed.

3. **Solvers**, $\mathcal{S}$: each leaf node $l \in \mathcal{N}_{leaf}$ is assigned to a *solver* module, $s_l^{\boldsymbol{\phi}} : \mathcal{X}_l \rightarrow \mathcal{Y} \in \mathcal{S}$, parametrised by $\boldsymbol{\phi}$, which operates on the transformed input data and outputs an estimate for the conditional distribution $p(\mathbf{y}|\mathbf{x})$. For classification tasks, we can define, for example, $s^{\boldsymbol{\phi}}$ as a linear classifier on the feature space $\mathcal{X}_l$, which outputs a distribution over classes.

Defining operations on the graph $\mathbb{T}$ amounts to a specification of the triplet $\mathbb{O} = (\mathcal{R}, \mathcal{T}, \mathcal{S})$. For example, given image inputs, we would choose the operations of each module to be from the set of operations commonly used in CNNs (examples are given in Tab. 7.2). In this case, every computational path on the resultant ANT, as well as the set of routers that guide inputs to one of these paths, are given by CNNs. Lastly, many existing tree-structured models [226, 227, 228, 229, 230, 231, 232] are instantiations of ANTs with limitations which we will address with our model (see Sec. 7.2 for a more detailed discussion).

7.3.2 Probabilistic Model and Inference

An ANT $(\mathbb{T}, \mathbb{O})$ models the conditional distribution $p(\mathbf{y}|\mathbf{x})$ as a hierarchical mixture of experts (HMEs) [238], each of which is defined as an NN and is a root-to-leaf path in the tree. Standard HMEs are a special case of ANTs where transformers are the identity function. As a result, the representations within experts are hierarchically shared between similar experts, unlike the

independent representations within experts in standard HMEs. In addition, ANTs come with a growth mechanism to determine the number of needed experts and their complexity, as discussed in Sec. 7.4.

Each input to the ANT, $\mathbf{x}$, stochastically traverses the tree based on decisions of routers and undergoes a sequence of transformations until it reaches a leaf node where the corresponding solver predicts the label $\mathbf{y}$. Suppose we have L leaf nodes, the full predictive distribution, with parameters $\Theta = (\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi})$, is given by

$$p(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{l=1}^{L} \underbrace{p(z_l = 1|\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\psi})}_{\text{Leaf-assignment prob. } \pi_l^{\boldsymbol{\theta},\boldsymbol{\psi}}} \underbrace{p(\mathbf{y}|\mathbf{x}, z_l = 1, \boldsymbol{\phi}, \boldsymbol{\psi})}_{\text{Leaf-specific prediction. } p_l^{\boldsymbol{\phi},\boldsymbol{\psi}}} \tag{7.1}$$

where $\mathbf{z} \in \{0, 1\}^L$ is an L-dimensional binary latent variable such that $\sum_{l=1}^{L} z_l = 1$, which describes the choice of leaf node (e.g. $z_l = 1$ means that leaf l is used). Here $\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}$ summarise the parameters of router, transformer and solver modules in the tree. The mixing coefficient $\pi_l^{\boldsymbol{\theta},\boldsymbol{\psi}}(\mathbf{x}) := p(z_l = 1|\mathbf{x}, \boldsymbol{\psi}, \boldsymbol{\theta})$ quantifies the probability that $\mathbf{x}$ is assigned to leaf l and is given by a product of decision probabilities over all router modules on the unique path $\mathcal{P}_l$ from the root to leaf node l:

$$\pi_l^{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}) = \prod_{r_j^{\boldsymbol{\theta}} \in \mathcal{P}_l} r_j^{\boldsymbol{\theta}}(\mathbf{x}_j^{\boldsymbol{\psi}})^{\mathbb{1}_{l \swarrow j}} \cdot \left(1 - r_j^{\boldsymbol{\theta}}(\mathbf{x}_j^{\boldsymbol{\psi}})\right)^{1 - \mathbb{1}_{l \swarrow j}} \tag{7.2}$$

where $l \swarrow j$ is a binary relation and is only true if leaf l is in the left subtree of internal node j, and $\mathbf{x}_j^{\boldsymbol{\psi}}$ is the feature representation of $\mathbf{x}$ at node j. Let $\mathcal{T}_j = \{t_{e_1}^{\boldsymbol{\psi}}, ..., t_{e_n}^{\boldsymbol{\psi}}\}$ denote the ordered set of the n transformer modules on the path from the root to node j, the feature vector $\mathbf{x}_j^{\boldsymbol{\psi}}$ is given by

$$\mathbf{x}_j^{\boldsymbol{\psi}} := \left(t_{e_n}^{\boldsymbol{\psi}} \circ ... \circ t_{e_2}^{\boldsymbol{\psi}} \circ t_{e_1}^{\boldsymbol{\psi}}\right)(\mathbf{x}).$$

On the other hand, the leaf-specific conditional distribution $p_l^{\boldsymbol{\phi},\boldsymbol{\psi}}(\mathbf{y}) := p(\mathbf{y}|\mathbf{x}, z_l = 1, \boldsymbol{\phi}, \boldsymbol{\psi})$ in (7.1) yields an estimate for the distribution over target $\mathbf{y}$ for leaf node l and is given by its solver's output $s_l^{\boldsymbol{\phi}}(\mathbf{x}_{\text{parent}(l)}^{\boldsymbol{\psi}})$.

We consider two inference schemes based on a trade-off between accuracy and computation, which we refer to as *multi-path* and *single-path* inference. The multi-path inference uses the *full predictive distribution* given in (7.1) as estimate for $p(\mathbf{y}|\mathbf{x})$. However, computing this quantity requires averaging the distributions over all the leaves involving computing all operations at all nodes and edges of the tree, which is expensive for a large ANT. On the other hand, the single-path inference scheme only uses the predictive distribution at the leaf node chosen by greedily traversing the tree in the directions of highest confidence of the routers. This approximation constrains computations to a single path, allowing for more memory- and time-efficient inference.

7.4 Optimisation

Training of an ANT proceeds in two stages: 1) *growth phase* during which the model architecture is learned based on *local* optimisation, and 2) *refinement phase* which further tunes the parameters of the model discovered in the first phase based on *global* optimisation. We include a pseudocode of the training algorithm in Supp. Sec. A.

7.4.1 Loss function: optimising parameters of $\mathbb{O}$

For both phases, we use the negative log-likelihood (NLL) as the common objective function to minimise:

$$-\log p(\mathbf{Y}|\mathbf{X},\Theta) = -\sum_{n=1}^{N} \log \left(\sum_{l=1}^{L} \pi_l^{\boldsymbol{\theta},\boldsymbol{\psi}}(\mathbf{x}^{(n)})\, p_l^{\boldsymbol{\phi},\boldsymbol{\psi}}(\mathbf{y}^{(n)})\right)$$

where $\mathbf{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$, $\mathbf{Y} = \{\mathbf{y}^{(1)}, ..., \mathbf{y}^{(N)}\}$ denote the training inputs and targets. As all component modules (routers, transformers and solvers) are differentiable with respect to their parameters $\Theta = (\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi})$, we can use gradient-based optimisation. Given an ANT with fixed topology $\mathbb{T}$, we use backpropagation [245] for gradient computation and use gradient descent to minimise the NLL for learning the parameters.

7.4.2 Growth phase: learning architecture $\mathbb{T}$

We next describe our proposed method for growing the tree $\mathbb{T}$ to an architecture of adequate complexity for the given training data. Starting from the root, we choose one of the leaf nodes in breadth-first order and incrementally modify the architecture by adding computational modules to it. In particular, we evaluate 3 choices (Fig. 7.1 (Right)) at each leaf node; (1)"split data" extends the current model by splitting the node with an addition of a new router; (2) "deepen transform" increases the depth of the incoming edge by adding a new transformer; (3) "keep" retains the current model. We then locally optimise the parameters of the newly added modules in the architectures of (1) and (2) by minimising NLL via gradient descent, while fixing the parameters of the previous part of the computational graph. Lastly, we select the model with the lowest validation NLL if it improves on the previously observed lowest NLL, otherwise we execute (3). This process is repeated to all new nodes level-by-level until no more "split data" or "deepen transform" operations pass the validation test.

The rationale for evaluating the two choices is to give the model a freedom to choose the most effective option between "going deeper" or splitting the data space. Splitting a node is equivalent to a soft partitioning of the feature space of incoming data, and gives birth to two new leaf nodes (left and right children solvers). In this case, the added transformer modules on the two branches are identity functions. Deepening an edge on the other hand seeks to learn richer representation via an extra nonlinear transformation, and replaces the old solver with a new one. Local optimisation is efficient in time and space; gradients only need to be computed for the parameters of the new parts of the architecture, reducing computation, while forward activations prior to the new parts do not need to be stored in memory, saving space.

7.4.3 Refinement phase: global tuning of $\mathbb{O}$

Once the model topology is determined in the growth phase, we finish by performing global optimisation to refine the parameters of the model, now with a fixed architecture. This time, we perform gradient descent on the NLL with respect to the parameters of all modules in the graph, jointly optimising the hierarchical grouping of data to paths on the tree and the associated expert NNs. The refinement phase can correct suboptimal decisions made during the local optimisation of the growth phase, and empirically improves the generalisation error (see Sec. 7.5.3).

7.5 Experiments

We evaluate ANTs using the SARCOS multivariate regression dataset [233], and the MNIST [234] and CIFAR-10 [235] classification datasets. We run ablation studies to show that our different components are vital for the best performance. We then assess the ability of ANTs to automatically learn meaningful hierarchical structures in data. Next, we examine the effects

of refinement phase on ANTs, and show that it can automatically prune the tree. Finally, we demonstrate that our proposed training procedure adapts the model size appropriately under varying amounts of labelled data. All of our models are implemented in PyTorch [255] and is available at `https://github.com/rtanno21609/AdaptiveNeuralTrees`. Full training details, including training times, are provided in Supp. Sec. C and D.

7.5.1 Model Performance

We compare the performance of ANTs (Tab. 7.2) against a range of DT and NN models (Tab. 7.3), where notably the relative performance of these two classes of models differs between datasets. ANTs inherit from both and achieve the lowest error on SARCOS, and perform favourably on MNIST and CIFAR-10. In general, DT methods without feature learning, such as RFs [11, 236] and GBTs [237], perform poorly on image classification tasks [235]. In comparison with CNNs without shortcut connections [234, 250, 251, 252], different ANTs balance between stronger performance with comparable numbers of trainable parameters, and comparable performance with smaller amount of parameters. At the other end of the spectrum, state-of-the-art NNs [249, 254] contain significantly more parameters.

Conditional computation: Tab.7.3 compares the errors and number of parameters of different ANTs for both multi-path and single-path inference schemes. While reducing the number of parameters (from Params (multi-path) to Params (single-path)) across all ANT models, we observe only a small difference in error (between Error (multi-path) and Error (single-path)), with the largest deviations being 0.06% for classification and 0.158 for regression. In addition, Supp.Sec. H shows that the single-path inference reduces FLOPS. This means that single-path inference gives an accurate approximation of the multi-path inference, while being more efficient to compute. This close approximation comes from the confident splitting probabilities of routers, being close to 0 or 1 (see blue histograms in Fig. 7.2(b)).

Ablation study: we compare the predictive errors of different variants of ANTs in cases where the options for adding transformer or router modules are disabled (see Tab. 7.4). In the first case, the resulting models are equivalent to SDTs [226] or HMEs [238] with locally grown architectures, while the second case is equivalent to standard CNNs, grown adaptively layer by layer. We observe that either ablation consistently leads to higher errors across different module configurations on all three datasets, justifying the combination of feature learning and hierarchical partitioning in ANTs.

SARCOS multivariate regression: Tab. 7.3 shows that ANT-SARCOS outperforms all other methods in mean squared error (MSE) with the full set of parameters. With the single-path inference, GBTs performs slightly better than a single ANT while requiring fewer parameters. We note that the top 3 methods are all tree-based, with the third best method being an SDT (with MLP routers). On the other hand, ANT and GBTs outperform the best standard NN model with less than a half of the parameter count. This highlights the value of hierarchical clustering for predictive performance and inference speed. Meanwhile, we still reap the benefits of representation learning, as shown by both ANT-SARCOS and the SDT (which is a specific form of ANT with identity transformers) requiring fewer parameters than the best-performing GBT configuration. Finally, we note that deeper NNs (5 vs. 3 hidden layers) can overfit on this small dataset, which makes the adaptive growth procedure of tree-based methods ideal for finding a model that exhibits good generalisation.

MNIST digit classification: we observe that ANT-MNIST-A outperforms state-of-the-art GBT [237] and RF [236] methods in accuracy. This performance is attained despite the use of a single tree, while RF methods operate with ensembles of classifiers (the size shown in Tab. 7.2). In particular, the NDF [230] has a pre-specified architecture where LeNet-5 [234] is used as the root transformer module, and 10 trees of fixed depth 5 are built on this base features. On the other hand, ANT-MNIST-A is constructed in a data-driven manner from

primitive modules, and displays an improvement over the NDF both in terms of accuracy and number of parameters. In addition, reducing the size of convolution kernels (ANT-MNIST-B) reduces the total number of parameters by 25% and the path-wise average by almost 40% while only increasing the error by $< 0.1\%$.

We also compare against the LeNet-5 CNN [234], comprised of the same types of operations used in our primitive modules (i.e. convolutional, max-pooling and FC layers). For a fair comparison, the network is trained with the same protocol as that of the ANT refinement phase, achieving an error rate of 0.82%. Both ANT-MNIST-A and ANT-MNIST-B attain better accuracy with a smaller number of parameters than LeNet-5. The current state-of-the-art, capsule networks (CapsNets) [249], have more parameters than ANT-MNIST-A by almost two orders of magnitude.[1] By ensembling ANTs, we can reach similar performance (0.29% versus 0.25%) with an order of magnitude less parameters (see Supp.Sec. H).

Lastly, we highlight the observation that ANT-MNIST-C, with the simplest primitive modules, achieves an error rate of 1.68% with single-path inference, which is significantly better than that of the linear classifier (7.91%), while engaging almost the same number of parameters ($7,956$ vs. $7,840$) on average. To isolate the benefit of convolutions, we took one of the root-to-path CNNs on ANT-MNIST-C and increased the number of kernels to adjust the number of parameters to the same value. We observe a higher error rate of 3.55%, which indicates that while convolutions are beneficial, data partitioning has additional benefits in improving accuracy. This result demonstrates the potential of ANT growth protocol for constructing performant models with lightweight inference. See Sec. G in the supplementary materials for the architecture of ANT-MNIST-C.

CIFAR-10 object recognition: we see that ANTs largely outperform the state-of-the-art DT method, gcForest [236], achieving over 90% accuracy, demonstrating the benefit of representation learning in tree-structured models. Secondly, with fewer number of parameters in single-path inference, ANT-CIFAR-A achieves higher accuracy than CNN models without shortcut connections [250, 251, 252] that held the state-of-the-art performance in respective years. With simpler primitive modules we learn more compact models (ANT-MNIST-B and -C) with a marginal compromise in accuracy. In addition, initialising the parameters of transformers and routers from a pre-trained single-path CNN further reduced the error rate of ANT-MNIST-A by 20% (see ANT-MNIST-A* in Tab. 7.3), indicating room for improvement in our proposed optimisation method.

Shortcut connections [256] have recently lead to leaps in performance in deep CNNs [253, 254]. We observe that our best network, ANT-MNIST-A*, has a comparable error rate and half the parameter count (with single-path inference) to the best-performing residual network, ResNet-110 [253]. Densely connected networks leads to better accuracy, but with an order of magnitude more parameters [254]. We expect shortcut connections to improve ANT performance, and leave integrating them to future work.

7.5.2 Interpretability

The growth procedure of ANTs is capable of discovering hierarchical structures in the data that are useful to the end task. Without any regularization imposed on routers, the learned hierarchies often display strong specialisation of paths to certain classes or categories of data on both the MNIST and CIFAR-10 datasets. Fig. 7.2 (a) displays an example with particularly "human-interpretable" partitions e.g. man-made versus natural objects, and road vehicles versus other types of vehicles. It should, however, be noted that human intuitions on relevant hierarchical structures do not necessarily equate to optimal representations, particularly as datasets may not necessarily have an underlying hierarchical structure, e.g., MNIST. Rather, what needs to

[1] Notably, CapsNets also feature a routing mechanism, but with a significantly different mechanism and motivation.

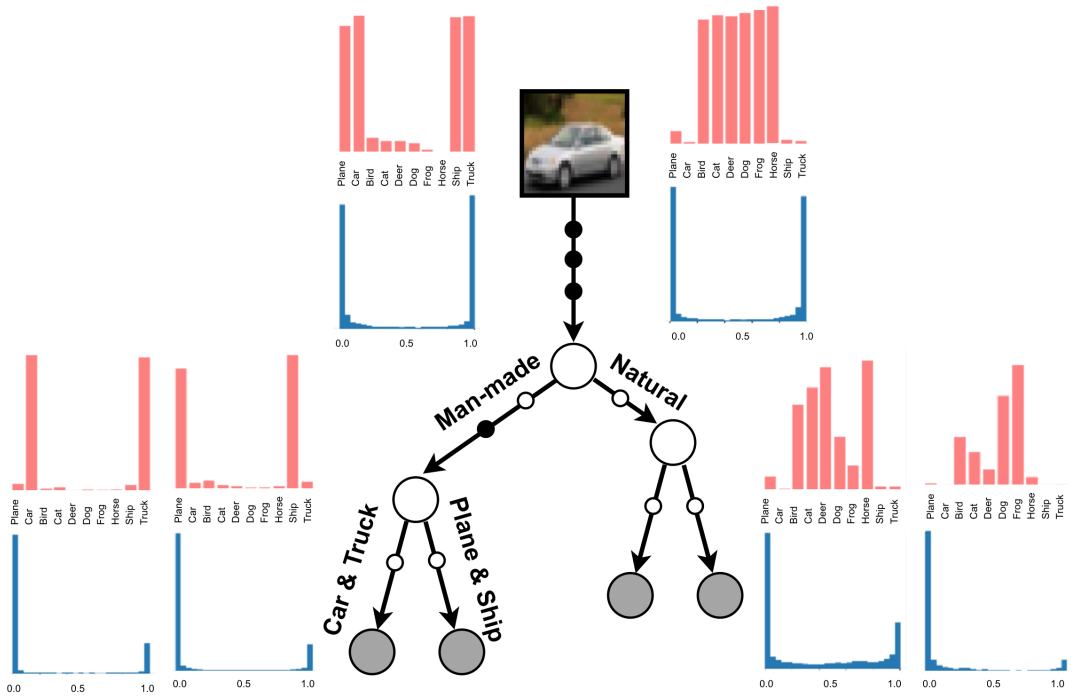
be highlighted is the ability of ANTs to learn when to share or separate the representation of data to optimise end-task performance, which gives rise to automatically discovering such hierarchies. To further attest that the model learns a meaningful routing strategy, we also present the test accuracy of the predictions from the leaf node with the smallest reaching probability in Supp. Sec. F. We observe that using the least likely "expert" leads to a substantial drop in classification accuracy. In addition, most learned trees are unbalanced. This property of adaptive computation is plausible since certain types of images may be easier to classify than others, as seen in prior work [257].

7.5.3 Effect of global refinement

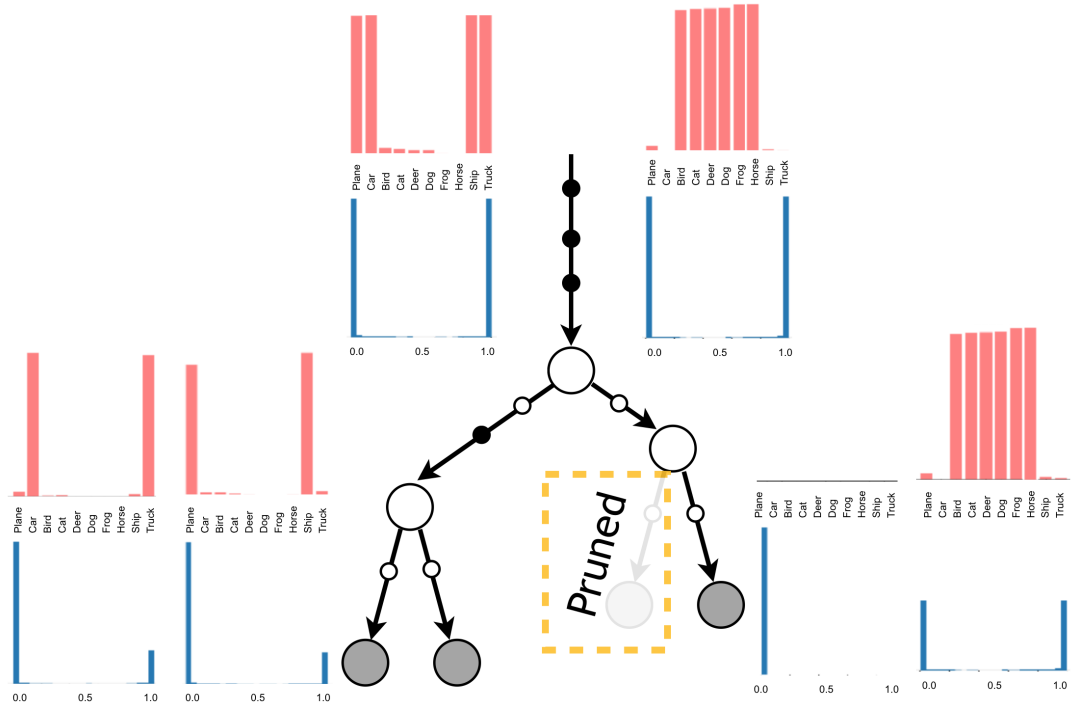
We observe that global refinement phase improves the generalisation error. Fig. 7.3 (Right) shows the generalisation error of various ANT models on CIFAR-10, with vertical dotted lines indicating the epoch when the models enter the refinement phase. As we switch from optimising parts of the ANT in isolation to optimising all parameters, we shift the optimisation landscape, resulting in an initial drop in performance. However, they all consistently converge to higher test accuracy than the best value attained during the growth phase. This provides evidence that refinement phase remedies suboptimal decisions made during the locally-optimised growth phase. In many cases, we observed that global optimisation polarises the decision probability of routers, which occasionally leads to the effective "pruning" of some branches. For example, in the case of the tree shown in Fig. 7.2(b), we observe that the decision probability of routers are more concentrated near 0 or 1 after global refinement, and as a result, the empirical probability of visiting one of the leaf nodes, calculated over the validation set, reduces to 0.09%—meaning that the corresponding branch could be pruned without a negligible change in the network's accuracy. The resultant model attains lower generalisation error, showing the pruning has resolved a suboptimal partioning of data.

7.5.4 Adaptive model complexity

Overparametrised models, trained without regularization, are vulnerable to overfitting on small datasets. Here we assess the ability of our proposed ANT training method to adapt the model complexity to varying amounts of labelled data. We run classfication experiments on CIFAR-10 and train three variants of ANTs, All-CNN [252] and linear classifier on subsets of the dataset of sizes 50, 250, 500, 2.5k, 5k, 25k and 45k (the full training set). Here we choose All-CNN as the baseline as it has similar number of parameters when trained on the full dataset and is the closest in terms of constituent operations (convolutional, GAP and FC layers). Fig.7.3 (Left) shows the corresponding test performances. The best model is picked based on the performance on the same validation set of 5k examples as before. As the dataset gets smaller, the margin between the test accuracy of the ANT models and All-CNN/linear classifier increases (up to 13%). Fig. 7.3 (Middle) shows the model size of discovered ANTs as the dataset size varies. For different settings of primitive modules, the number of parameters generally increases as a function of the dataset size. All-CNN has a fixed number of parameters, consistently larger than the discovered ANTs, and suffers from overfitting, particularly on small datasets. The linear classifier, on the other hand, underfits to the data. Our method constructs models of adequate complexity, leading to better generalisation. This shows the value of our tree-building algorithm over using models of fixed-size structures.



(a) Before refinement



(b) After refinement

Figure 7.2: Visualisation of class distributions (red) and path probabilities (blue) computed over the whole test set at respective nodes of an example ANT (a) before and (b) after the refinement phase. (a) shows that the model captures an interpretable hierarchy, grouping semantically similar images on the same branches. (b) shows that the refinement phase polarises path probabilities, pruning a branch.

Table 7.3: Comparison of performance of different models on SARCOS, MNIST and CIFAR-10. The columns "Error (multi-path)" and "Error (single-path)" indicate the classification (%) or regression (MSE) errors of predictions based on the multi-path and the single-path inference. The columns "Params. (multi-path)" and "Params. (single-path)" respectively show the total number of parameters in the model and the average number of parameters used during single-path inference. "Ensemble Size" indicates the size of ensemble used. An entry of "–" indicates that no value was reported. Methods marked with † are from our implementations trained in the same experimental setup. * indicates that the parameters are initialised with a pre-trained CNN.

	Method	Error (multi-path)	Error (single-path)	Params. (multi-path)	Params. (single-path)	Ensemble Size
SARCOS	Linear regression	10.693	N/A	154	N/A	1
	MLP with 2 hidden layers [246]	5.111	N/A	31,804	N/A	1
	Decision tree	3.708	3.708	319,591	25	1
	MLP with 1 hidden layer	2.835	N/A	7,431	N/A	1
	Gradient boosted trees	2.661	2.661	391,324	2,083	7 × 30
	MLP with 5 hidden layers	2.657	N/A	270,599	N/A	1
	Random forest	2.426	2.426	40,436,840	4,791	200
	Random forest	2.394	2.394	141,540,436	16,771	700
	MLP with 3 hidden layers	2.129	N/A	139,015	N/A	1
	SDT (with MLP routers)	2.118	2.246	28,045	10,167	1
	Gradient boosted trees	1.444	1.444	988,256	6,808	7 × 100
	ANT-SARCOS	1.384	1.542	103,823	61,640	1
	ANT-SARCOS (ensemble)	1.226	1.372	598,280	360,766	8
MNIST	Linear classifier	7.91	N/A	7,840	N/A	1
	RDT [**?**]	5.41	–	–	–	1
	Random Forests [11]	3.21	3.21	–	–	200
	Compact Multi-Class Boosted Trees [237]	2.88	–	–	–	100
	Alternating Decision Forest [247]	2.71	2.71	–	–	20
	Neural Decision Tree [232]	2.10	–	1,773,130	502,170	1
	ANT-MNIST-C	1.62	1.68	39,670	7,956	1
	MLP with 2 hidden layers [248]	1.40	N/A	1,275,200	N/A	1
	LeNet-5† [234]	0.82	N/A	431,000	N/A	1
	gcForest [236]	0.74	0.74	–	–	500
	ANT-MNIST-B	0.72	0.73	76,703	50,653	1
	Neural Decision Forest [230]	0.70	–	544,600	463,180	10
	ANT-MNIST-A	0.64	0.69	100,596	84,935	1
	ANT-MNIST-A (ensemble)	0.29	0.30	850,775	655,449	8
	CapsNet [249]	0.25	–	8.2M	N/A	1
CIFAR-10	Compact Multi-Class Boosted Trees [237]	52.31	–	–	–	100
	Random Forests [11]	50.17	50.17	–	–	2000
	gcForest [236]	38.22	38.22	–	–	500
	MaxOut [250]	9.38	N/A	6M	N/A	1
	ANT-CIFAR10-C	9.31	9.34	0.7M	0.5M	1
	ANT-CIFAR10-B	9.15	9.18	0.9M	0.6M	1
	Network in Network [251]	8.81	N/A	1M	N/A	1
	All-CNN†[252]	8.71	N/A	1.4M	N/A	1
	ANT-CIFAR10-A	8.31	8.32	1.4M	1.0M	1
	ANT-CIFAR10-A (ensemble)	7.71	7.79	8.7M	7.4M	8
	ANT-CIFAR10-A*	6.72	6.74	1.3M	0.8M	1
	ResNet-110 [253]	6.43	N/A	1.7M	N/A	1
	DenseNet-BC (k=24) [254]	3.74	N/A	27.2M	N/A	1

Table 7.4: Ablation study on regression (MSE) and classification (%) errors. "CNN" refers to the case where the ANT is grown without routers while "SDT/HME" refers to the case where transformer modules on the edges are disabled.

Model	**Error (multi-path)**			**Error (single-path)**		
	ANT (default)	CNN (no $\mathcal{R}$)	HME (no $\mathcal{T}$)	ANT (default)	CNN (no $\mathcal{R}$)	HME (no $\mathcal{T}$)
SARCOS	1.38	2.51	2.12	1.54	2.51	2.25
MNIST-A	0.64	0.74	3.18	0.69	0.74	4.19
MNIST-B	0.72	0.80	4.63	0.73	0.80	3.62
MNIST-C	1.62	3.71	5.70	1.68	3.71	6.96
CIFAR10-A	8.31	9.29	39.29	8.32	9.29	40.33
CIFAR10-B	9.15	11.08	43.09	9.18	11.08	44.25
CIFAR10-C	9.31	11.61	48.59	9.34	11.61	50.02

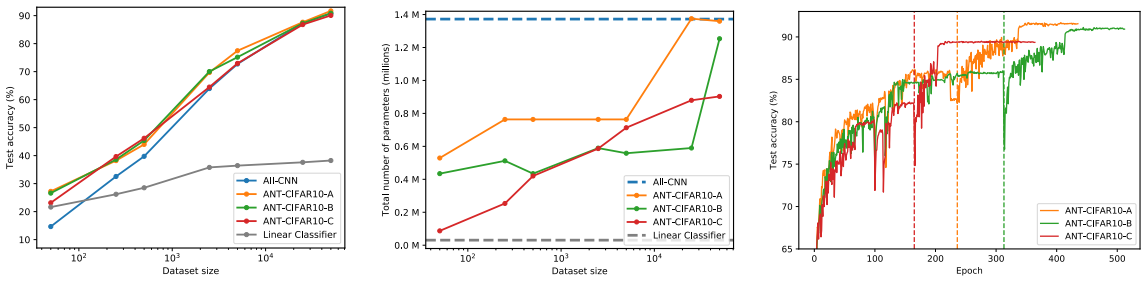


Figure 7.3: (Left). Test accuracy on CIFAR-10 of ANTs for varying amounts of training data. (Middle) The complexity of the grown ANTs increases with dataset size. (Right) Refinement improves generalisation; the dotted lines show where the refinement phase starts.

Bibliography

[1] Ryutaro Tanno, Aurobrata Ghosh, Francesco Grussu, Enrico Kaden, Antonio Criminisi, and Daniel C Alexander. Bayesian image quality transfer. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2016.

[2] François Rousseau. Brain hallucination. In *ECCV 2008*, pages 497–508. Springer, 2008.

[3] Pierrick Coupé and et al. Collaborative patch-based super-resolution for diffusion-weighted images. *NeuroImage*, 83:245–261, 2013.

[4] Andrea Rueda, Norberto Malpica, and Eduardo Romero. Single-image super-resolution of brain MR images using overcomplete dictionaries. *MIA*, 17(1):113–132, 2013.

[5] Yun-Heng Wang, Jiaqing Qiao, Jun-Bao Li, Ping Fu, Shu-Chuan Chu, and John F Roddick. Sparse representation-based MRI super-resolution reconstruction. *Measurement*, 47, 2014.

[6] Dong Hye Ye, Darko Zikic, Ben Glocker, Antonio Criminisi, and Ender Konukoglu. Modality propagation: coherent synthesis of subject-specific scans with data-driven regularization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 606–613. Springer, 2013.

[7] Amod Jog, Aaron Carass, Snehashis Roy, Dzung L Pham, and Jerry L Prince. MR image synthesis by contrast learning on neighborhood ensembles. *Medical image analysis*, 24(1):63–76, 2015.

[8] Ninon Burgos, M Jorge Cardoso, Filipa Guerreiro, Catarina Veiga, Marc Modat, Jamie McClelland, Antje-Christin Knopf, Shonit Punwani, David Atkinson, Simon R Arridge, et al. Robust CT synthesis for radiotherapy planning: Application to the head and neck region. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 476–484. Springer, 2015.

[9] Wolfgang Wein, Shelby Brunke, Ali Khamene, Matthew R Callstrom, and Nassir Navab. Automatic CT-ultrasound registration for diagnostic imaging and image-guided intervention. *Medical image analysis*, 12(5):577–585, 2008.

[10] Stamatios N Sotiropoulos and et al. Advances in diffusion MRI acquisition and processing in the Human Connectome Project. *Neuroimage*, 80:125–143, 2013.

[11] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013.

[13] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

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

[15] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Proc*, 13(4), 2004.

[16] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[18] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[19] Holger R Roth, Le Lu, Ari Seff, Kevin M Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M Summers. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 520–527. Springer, 2014.

[20] Teresa Araújo, Guilherme Aresta, Eduardo Castro, José Rouco, Paulo Aguiar, Catarina Eloy, António Polónia, and Aurélio Campilho. Classification of breast cancer histology images using Convolutional Neural Networks. *PloS one*, 12(6):e0177544, 2017.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[22] Ozan Oktay, Wenjia Bai, Matthew Lee, Ricardo Guerrero, Konstantinos Kamnitsas, Jose Caballero, Antonio de Marvao, Stuart Cook, Declan O'Regan, and Daniel Rueckert. Multi-input Cardiac Image Super-Resolution Using Convolutional Neural Networks. In *MICCAI*. Springer, 2016.

[23] Yuhua Chen, Feng Shi, Anthony G Christodoulou, Yibin Xie, Zhengwei Zhou, and Debiao Li. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 91–99. Springer, 2018.

[24] Daniele Ravì, Agnieszka Barbara Szczotka, Stephen P Pereira, and Tom Vercauteren. Adversarial training with cycle consistency for unsupervised super-resolution in endomicroscopy. *Medical image analysis*, 53:123–131, 2019.

[25] Dong Nie, Xiaohuan Cao, Yaozong Gao, Li Wang, and Dinggang Shen. Estimating CT image from MRI data using 3D fully convolutional networks. In *Deep Learning and Data Labeling for Medical Applications*, pages 170–178. Springer, 2016.

[26] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical physics*, 44(10), 2017.

[27] A Benou, R Veksler, A Friedman, and T Riklin Raviv. Ensemble of expert deep neural networks for spatio-temporal denoising of contrast-enhanced MRI sequences. *Medical image analysis*, 42:145–159, 2017.

[28] Hu Chen, Yi Zhang, Weihua Zhang, Peixi Liao, Ke Li, Jiliu Zhou, and Ge Wang. Low-dose CT via convolutional neural network. *Biomedical optics express*, 8(2):679–694, 2017.

[29] Suheyla Cetin Karayumak, Marek Kubicki, and Yogesh Rathi. Harmonizing Diffusion MRI Data Across Magnetic Field Strengths. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 116–124. Springer, 2018.

[30] Chantal MW Tax, Francesco Grussu, Enrico Kaden, Lipeng Ning, Umesh Rudrapatna, John Evans, Samuel St-Jean, Alexander Leemans, Simon Koppers, Dorit Merhof, et al. Cross-scanner and cross-protocol diffusion MRI data harmonisation: A benchmark database and evaluation of algorithms. *NeuroImage*, 2019.

[31] Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing MRI. In *Advances in neural information processing systems*, pages 10–18, 2016.

[32] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[33] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[34] Jo Schlemper, Jose Caballero, Joseph V Hajnal, Anthony N Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE transactions on medical imaging*, 37(2):491–503, 2018.

[35] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487, 2018.

[36] Guang Yang, Simiao Yu, Hao Dong, Greg Slabaugh, Pier Luigi Dragotti, Xujiong Ye, Fangde Liu, Simon Arridge, Jennifer Keegan, Yike Guo, et al. Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE transactions on medical imaging*, 37(6):1310–1321, 2018.

[37] Yeo Hun Yoon, Shujaat Khan, Jaeyoung Huh, and Jong Chul Ye. Efficient b-mode ultrasound image reconstruction from sub-sampled rf data using deep learning. *IEEE transactions on medical imaging*, 38(2):325–336, 2019.

[38] Hessam Sokooti, Bob de Vos, Floris Berendsen, Boudewijn PF Lelieveldt, Ivana Išgum, and Marius Staring. Nonrigid image registration using multi-scale 3D convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 232–239. Springer, 2017.

[39] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An Unsupervised Learning Model for Deformable Medical Image Registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9252–9260, 2018.

[40] Lingyun Wu, Jie-Zhi Cheng, Shengli Li, Baiying Lei, Tianfu Wang, and Dong Ni. FUIQA: Fetal ultrasound image quality assessment with deep convolutional networks. *IEEE transactions on cybernetics*, 47(5):1336–1349, 2017.

[41] Steven J Esses, Xiaoguang Lu, Tiejun Zhao, Krishna Shanbhogue, Bari Dane, Mary Bruno, and Hersh Chandarana. Automated image quality evaluation of T2-weighted liver MRI utilizing deep learning architecture. *Journal of Magnetic Resonance Imaging*, 47(3):723–728, 2018.

[42] Joseph Paul Cohen, Margaux Luck, and Sina Honari. Distribution Matching Losses Can Hallucinate Features in Medical Image Translation. *arXiv preprint arXiv:1805.08841*, 2018.

[43] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20, 2019.

[44] Stephen C Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223, 1996.

[45] Hong Wang, Dennis M Levi, and Stanley A Klein. Intrinsic uncertainty and integration efficiency in bisection acuity. *Vision research*, 36(5):717–739, 1996.

[46] David Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):45–70, 1995.

[47] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2):105–112, 2009.

[48] Ryutaro Tanno, Daniel E Worrall, Aurobrata Ghosh, Enrico Kaden, Stamatios N Sotiropoulos, Antonio Criminisi, and Daniel C Alexander. Bayesian image quality transfer with cnns: Exploring uncertainty in dmri super-resolution. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 611–619. Springer, 2017.

[49] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2017.

[50] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[51] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *IEEE WCCI*, volume 1, pages 55–60. IEEE, 1994.

[52] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NIPS*, pages 2575–2583, 2015.

[53] Daniel C Alexander and et al. Image quality transfer via random forest regression: applications in diffusion MRI. In *MICCAI 2014*, pages 225–232. Springer, 2014.

[54] Daniel C Alexander, Darko Zikic, Aurobrata Ghosh, Ryutaro Tanno, Viktor Wottschel, Jiaying Zhang, Enrico Kaden, Tim B Dyrby, Stamatios N Sotiropoulos, Hui Zhang, et al. Image quality transfer and applications in diffusion MRI. *Neuroimage*, 152:283–298, 2017.

[55] Stefano B Blumberg, Ryutaro Tanno, Iasonas Kokkinos, and Daniel C Alexander. Deeper image quality transfer: Training low-memory neural networks for 3D images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 118–125. Springer, 2018.

[56] Michael P Harms, Leah H Somerville, Beau M Ances, Jesper Andersson, Deanna M Barch, Matteo Bastiani, Susan Y Bookheimer, Timothy B Brown, Randy L Buckner, Gregory C Burgess, et al. Extending the Human Connectome Project across ages: Imaging protocols for the Lifespan Development and Aging projects. *NeuroImage*, 183:972–984, 2018.

[57] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio de Marvao, Timothy Dawes, Declan P O?Regan, et al. Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation. *IEEE transactions on medical imaging*, 37(2):384–395, 2018.

[58] Can Zhao, Aaron Carass, Blake E Dewey, Jonghye Woo, Jiwon Oh, Peter A Calabresi, Daniel S Reich, Pascal Sati, Dzung L Pham, and Jerry L Prince. A deep learning based anti-aliasing self super-resolution algorithm for MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 100–108. Springer, 2018.

[59] Dwarikanath Mahapatra, Behzad Bozorgtabar, Sajini Hewavitharanage, and Rahil Garnavi. Image super resolution using generative adversarial networks and local saliency maps for retinal image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 382–390. Springer, 2017.

[60] Haichao Yu, Ding Liu, Honghui Shi, Hanchao Yu, Zhangyang Wang, Xinchao Wang, Brent Cross, Matthew Bramler, and Thomas S Huang. Computed tomography super-resolution using convolutional neural networks. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3944–3948. IEEE, 2017.

[61] Dong Nie, Roger Trullo, Jun Lian, Li Wang, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical Image Synthesis with Deep Convolutional Adversarial Networks. *IEEE Transactions on Biomedical Engineering*, 2018.

[62] Jelmer M Wolterink, Anna M Dinkla, Mark HF Savenije, Peter R Seevinck, Cornelis AT van den Berg, and Ivana Išgum. Deep MR to CT synthesis using unpaired data. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 14–23. Springer, 2017.

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[64] Khosro Bahrami, Feng Shi, Islem Rekik, and Dinggang Shen. Convolutional Neural Network for Reconstruction of 7T-like Images from 3T MRI Using Appearance and Anatomical Features. In *MICCAI DLDLM workshop*, pages 39–47. Springer, 2016.

[65] Stefano B Blumberg, Marco Palombo, Can Son Khoo, Chantal Tax, Ryutaro Tanno, and Daniel C Alexander. Multi-Stage Prediction Networks for Data Harmonization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019.

[66] Hu Shi, Daniel Worrall, Bas Veeling, Henkjan Huisman, and Max Welling. Supervised Uncertainty Quantification for Segmentation with Multiple Annotations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

[67] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised Learning for Fast Probabilistic Diffeomorphic Registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[68] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[69] Jo Schlemper, Guang Yang, Pedro Ferreira, Andrew Scott, Laura-Ann McGill, Zohya Khalique, Margarita Gorodezky, Malte Roehl, Jennifer Keegan, Dudley Pennell, et al. Stochastic Deep Compressive Sensing for the Reconstruction of Diffusion Tensor Cardiac MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[70] Felix JS Bragman, Ryutaro Tanno, Zach Eaton-Rosen, Wenqi Li, David J Hawkes, Sebastien Ourselin, Daniel C Alexander, Jamie R McClelland, and M Jorge Cardoso. Uncertainty in multitask learning: joint representations for probabilistic MR-only radiotherapy planning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[71] Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring Uncertainty Measures in Deep Networks for Multiple Sclerosis Lesion Detection and Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 655–663. Springer, 2018.

[72] Zach Eaton-Rosen, Felix Bragman, Sotirios Bisdas, Sébastien Ourselin, and M Jorge Cardoso. Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 691–699. Springer, 2018.

[73] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Insights and applications. In *Deep Learning Workshop, ICML*, 2015.

[74] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer's Disease Neuroimaging Initiative, et al. Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage*, 2019.

[75] Daniel E Worrall, Clare M Wilson, and Gabriel J Brostow. Automated Retinopathy of Prematurity Case Detection with Convolutional Neural Networks. In *MICCAI DLDLM Workshop*, pages 68–76. Springer, 2016.

[76] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):17816, 2017.

[77] Murat Seckin Ayhan and Philipp Berens. Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks. 2018.

[78] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Sendhil Mullainathan, and Jon M. Kleinberg. Direct Uncertainty Prediction for Medical Second Opinions. 2018.

[79] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems*, pages 6965–6975, 2018.

[80] Christian F. Baumgartner, Kerem C. Tezcan, Krishna Chaitanya, Andreas M. Hötker, Urs J. Muehlematter, Khoschy Schawkat, Anton S. Becker, Olivio Donati, and Ender Konukoglu. PHiSeg: Capturing Uncertainty in Medical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

[81] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[82] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning From Noisy Labels By Regularized Estimation Of Annotator Confusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[83] Xiao Yang, Roland Kwitt, and Marc Niethammer. Fast Predictive Image Registration. In *MICCAI DLDLM Workshop*, pages 48–57. Springer, 2016.

[84] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016.

[85] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE PAMI*, 38(2):295–307, 2016.

[86] Steven McDonagh, Benjamin Hou, Konstantinos Kamnitsas, Ozan Oktay, Amir Alansary, and Bernhard Kainz. Context-Sensitive Super-Resolution for Fast Fetal Magnetic Resonance Imaging. *arXiv preprint arXiv:1703.00035*, 2017.

[87] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[88] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and Checkerboard Artifacts. *Distill*, 2016.

[89] C Radhakrishna Rao. Estimation of heteroscedastic variances in linear models. *Journal of the American Statistical Association*, 65(329):161–172, 1970.

[90] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[91] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[92] Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482, 1993.

[93] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[94] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

[95] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

[96] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. *arXiv preprint arXiv:1701.05369*, 2017.

[97] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[98] Neil A Weiss. *A course in probability*. Addison-Wesley, 2006.

[99] Clive G Bowsher and Peter S Swain. Identifying sources of variation and the flow of information in biochemical networks. *Proceedings of the National Academy of Sciences*, 2012.

[100] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The WU-Minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[101] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*, 80:105–124, 2013.

[102] Matteo Figini, Marco Riva, Mark Graham, Gian Marco Castelli, Bethania Fernandes, Marco Grimaldi, Giuseppe Baselli, Federico Pessina, Lorenzo Bello, Hui Zhang, et al. Prediction of isocitrate dehydrogenase genotype in brain gliomas with MRI: single-shell versus multishell diffusion models. *Radiology*, 289(3):788–796, 2018.

[103] Peter J Basser, James Mattiello, and Denis LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, 1994.

[104] Evren Özarslan, Cheng Guan Koay, Timothy M Shepherd, Michal E Komlosh, M Okan İrfanoğlu, Carlo Pierpaoli, and Peter J Basser. Mean apparent propagator MRI: a novel diffusion imaging method for mapping tissue microstructure. *NeuroImage*, 78:16–32, 2013.

[105] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[106] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[107] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017.

[108] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[109] Enrico Kaden, Frithjof Kruggel, and Daniel C Alexander. Quantitative mapping of the per-axon diffusion coefficients in brain white matter. *Magnetic resonance in medicine*, 75(4):1752–1763, 2016.

[110] JD Tournier, F Calamante, and A Connelly. Improved probabilistic streamlines tractography by 2nd order integration over fibre orientation distributions. In *ISMRM*, page 1670, 2010.

[111] J-Donald Tournier, Fernando Calamante, and Alan Connelly. MRtrix: diffusion tractography in crossing fiber regions. *International journal of imaging systems and technology*, 22(1):53–66, 2012.

[112] Christopher M Bishop. Mixture density networks. Technical report, Citeseer, 1994.

[113] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2012.

[114] Diane Bouchacourt, Pawan K Mudigonda, and Sebastian Nowozin. Disco nets: Dissimilarity coefficients networks. In *Advances in Neural Information Processing Systems*, pages 352–360, 2016.

[115] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018.

[116] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

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

[118] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[119] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[120] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *European Conference on Computer Vision*, pages 402–418. Springer, 2016.

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

[122] Changyong Oh, Kamil Adamczewski, and Mijung Park. Radial and Directional Posteriors for Bayesian Neural Networks. *arXiv preprint arXiv:1902.02603*, 2019.

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

[124] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. *arXiv preprint arXiv:1902.03932*, 2019.

[125] Nick Pawlowski, Andrew Brock, Matthew CH Lee, Martin Rajchl, and Ben Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

[126] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org, 2017.

[127] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[128] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[129] Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

[130] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[131] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *International conference on information processing in medical imaging*, pages 597–609. Springer, 2017.

[132] Neerav Karani, Krishna Chaitanya, Christian Baumgartner, and Ender Konukoglu. A lifelong learning approach to brain mr segmentation across scanners and protocols. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 476–484. Springer, 2018.

[133] Chaitanya Baweja, Ben Glocker, and Konstantinos Kamnitsas. Towards continual learning in medical imaging. *arXiv preprint arXiv:1811.02496*, 2018.

[134] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[135] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

[136] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[137] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.

[138] Dan Ma, Vikas Gulani, Nicole Seiberlich, Kecheng Liu, Jeffrey L Sunshine, Jeffrey L Duerk, and Mark A Griswold. Magnetic resonance fingerprinting. *Nature*, 495(7440):187–192, 2013.

[139] Ouri Cohen, Bo Zhu, and Matthew S Rosen. MR fingerprinting deep reconstruction network (DRONE). *Magnetic resonance in medicine*, 80(3):885–894, 2018.

[140] Yoseob Han, Jaejun Yoo, Hak Hee Kim, Hee Jung Shin, Kyunghyun Sung, and Jong Chul Ye. Deep learning with domain adaptation for accelerated projection-reconstruction MR. *Magnetic resonance in medicine*, 80(3):1189–1205, 2018.

[141] Hengameh Mirzaalian, Lipeng Ning, Peter Savadjiev, Ofer Pasternak, Sylvain Bouix, O Michailovich, G Grant, CE Marx, Rajendra A Morey, LA Flashman, et al. Inter-site and inter-scanner diffusion MRI data harmonization. *NeuroImage*, 135:311–323, 2016.

[142] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[143] Takeyuki Watadani, Fumikazu Sakai, Takeshi Johkoh, Satoshi Noma, Masanori Akira, Kiminori Fujimoto, Alexander A Bankier, Kyung Soo Lee, Nestor L Müller, Jae-Woo Song, et al. Interobserver variability in the CT assessment of honeycombing in the lungs. *Radiology*, 266(3):936–944, 2013.

[144] Andrew B Rosenkrantz, Ruth P Lim, Mershad Haghighi, Molly B Somberg, James S Babb, and Samir S Taneja. Comparison of interreader reproducibility of the prostate imaging reporting and data system and likert scales for evaluation of multiparametric prostate MRI. *American Journal of Roentgenology*, 201(4):W612–W618, 2013.

[145] Elizabeth Lazarus, Martha B Mainiero, Barbara Schepps, Susan L Koelliker, and Linda S Livingston. BI-RADS lexicon for US and mammography: interobserver variability and positive predictive value. *Radiology*, 239(2):385–391, 2006.

[146] Simon K Warfield, Kelly H Zou, and William M Wells. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903–921, 2004.

[147] Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010.

[148] Chengjiang Long, Gang Hua, and Ashish Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3000–3007, 2013.

[149] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2839–2847, 2015.

[150] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics*, pages 20–28, 1979.

[151] Padhraic Smyth, Usama M Fayyad, Michael C Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in neural information processing systems*, pages 1085–1092, 1995.

[152] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[153] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[154] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters*, 34(12):1428–1436, 2013.

[155] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual international conference on machine learning*, pages 889–896. ACM, 2009.

[156] Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *AISTATs*, pages 932–939, 2010.

[157] Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7474–7483, 2017.

[158] Grant Van Horn, Steve Branson, Scott Loarie, Serge Belongie, Cornell Tech, and Pietro Perona. Lean Multiclass Crowdsourcing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2723, 2018.

[159] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. Learning From Noisy Singly-labeled Data. In *International Conference on Learning Representations*, 2018.

[160] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[161] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

[162] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*, 2017.

[163] Melody Y Guan, Varun Gulshan, Andrew M Dai, and Geoffrey E Hinton. Who said what: Modeling individual labelers improves classification. *AAAI*, 2018.

[164] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

[165] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[166] Jakramate Bootkrajang and Ata Kabán. Label-noise robust logistic regression and its applications. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 143–158. Springer, 2012.

[167] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[168] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *International conference on machine learning*, pages 567–574, 2012.

[169] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2233–2241, 2017.

[170] Jacob Goldberger and Ehud Ben-Reuven. Training Deep Neural-networks Using a Noise Adaptation Layer. In *ICLR*, 2017.

[171] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J Belongie. Learning From Noisy Large-Scale Datasets With Minimal Supervision. In *CVPR*, pages 6575–6583, 2017.

[172] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, 2018.

[173] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *International Conference on Machine Learning*, pages 2309–2318, 2018.

[174] Ardavan Saeedi, Matthew D Hoffman, Stephen J DiVerdi, Asma Ghandeharioun, Matthew J Johnson, and Ryan P Adams. Multimodal Prediction and Personalization of Photo Edits with Deep Generative Models. *arXiv preprint arXiv:1704.04997*, 2017.

[175] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[176] N. Burgos et al. Iterative framework for the joint segmentation and CT synthesis of MR images: application to MRI-only radiotherapy treatment planning. *Phys. Med. Biol.*, 62, 2017.

[177] D. Nie et al. Medical image synthesis with context-aware generative adversarial networks. *arXiv:1612.05362*.

[178] J. Wolterink et al. Deep MR to CT Synthesis Using Unpaired Data. In *SASHIMI*, pages 14–22, 2017.

[179] A. Kendall et al. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*, 2018.

[180] P. Moeskops et al. Deep learning for multi-task medical image segmentation in multiple modalities. In *MICCAI*, pages 478–486, 2016.

[181] Rich Caruana. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *ICML*, 1993.

[182] W. Li et al. On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task. In *IPMI*, pages 348–360, 2017.

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

[184] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[185] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[186] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[187] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch Networks for Multi-task Learning. In *CVPR*, 2016.

[188] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

[189] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2019.

[190] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Advances in neural information processing systems*, pages 235–243, 2016.

[191] Sihong Chen, Dong Ni, Jing Qin, Baiying Lei, Tianfu Wang, and Jie-Zhi Cheng. Bridging computational features toward multiple semantic features with multi-task regression: A study of CT pulmonary nodules. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 53–60. Springer, 2016.

[192] Felix Bragman, Ryu Tanno, Zach Eaton-Rosen, Wenqi Li, David Hawkes, Sebastien Ourselin, Daniel Alexander, Jamie McClelland, and M. Jorge Cardoso. Uncertainty in multitask learning: joint representations for probabilistic MR-only radiotherapy planning. In *Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, 2018.

[193] Ryutaro Tanno, Antonios Makropoulos, Salim Arslan, Ozan Oktay, Sven Mischkewitz, Fouad Al-Noor, Jonas Oppenheimer, Ramin Mandegaran, Bernhard Kainz, and Mattias P Heinrich. AutoDVT: Joint Real-Time Classification for Vein Compressibility Analysis in Deep Vein Thrombosis Ultrasound Diagnostics. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 905–912. Springer, 2018.

[194] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent Multi-task Architecture Learning. 2019.

[195] Elliot Meyerson and Risto Miikkulainen. Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering. In *International Conference on Learning Representations*, 2018.

[196] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision*, pages 1062–1070, 2015.

[197] Brendan Jou and Shih-Fu Chang. Deep cross residual learning for multitask visual recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 998–1007. ACM, 2016.

[198] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[199] Song Yang Zhang, Zhifei and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[200] Mingsheng Long and Jianmin Wang. Learning multiple tasks with deep relationship networks. In *Advances in Neural Information Processing Systems*, 2017.

[201] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan):35–63, 2007.

[202] Laurent Jacob, Jean philippe Vert, and Francis R. Bach. Clustered Multi-Task Learning: A Convex Formulation. In *Advances in Neural Information Processing Systems 21*, 2009.

[203] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with Whom to Share in Multi-task Feature Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 521–528, USA, 2011. Omnipress.

[204] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Schmidt Feris. Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. In *CVPR*, volume 1, page 6, 2017.

[205] Youssef A Mejjati, Darren Cosker, and Kwang In Kim. Multi-Task Learning by Maximizing Statistical Dependence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3465–3473, 2018.

[206] Yani Ioannou, Duncan Robertson, Roberto Cipolla, Antonio Criminisi, et al. Deep roots: Improving CNN efficiency with hierarchical filter groups. 2017.

[207] Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 2016.

[208] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[209] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[210] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014.

[211] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the Compactness, Efficiency, and Representation of 3D Convolutional Networks: Brain Parcellation as a Pretext Task. 2017.

[212] Kerstin Kläser, Pawel Markiewicz, Marta Ranzini, Wenqi Li, Marc Modat, Brian F Hutton, David Atkinson, Kris Thielemans, M Jorge Cardoso, and Sebastien Ourselin. Deep Boosted Regression for MR to CT Synthesis, 2018.

[213] Zach Eaton-Rosen, Felix Bragman, Sotirios Bisdas, Sebastien Ourselin, and M. Jorge Cardoso. Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions, 2018.

[214] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–473. ACM, 2018.

[215] Alexandre Lacoste, Boris Oreshkin, Wonchang Chung, Thomas Boquet, Negar Rostamzadeh, and David Krueger. Uncertainty in Multitask Transfer Learning. In *arXiv:1806.07528*, 2018.

[216] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[217] Yoshua Bengio. Deep learning of representations: Looking forward. In *International Conference on Statistical Language and Speech Processing*, pages 1–37. Springer, 2013.

[218] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[219] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *ICLR*, 2017.

[220] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, 2013.

[221] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[222] Vlad Sandulescu and Mihai Chiru. Predicting the future relevance of research institutions-The winning solution of the KDD Cup 2016. *CoRR*, 2016.

[223] Kaggle.com. Two Sigma Financial Modeling Challenge, 2017.

[224] Loic Le Folgoc, Aditya V Nori, Siddharth Ancha, and Antonio Criminisi. Lifted auto-context forests for brain tumour segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 171–183. Springer, 2016.

[225] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. Content-based Neighbor Models for Cold Start in Recommender Systems. In *Proceedings of the Recommender Systems Challenge 2017*, page 7. ACM, 2017.

[226] Alberto Suárez and James F Lutsko. Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions. PAMI*, 21(12):1297–1311, 1999.

[227] Ozan İrsoy, Olcay Taner Yıldız, and Ethem Alpaydın. Soft decision trees. In *ICPR*, pages 1819–1822. IEEE, 2012.

[228] Dmitry Laptev and Joachim M Buhmann. Convolutional decision trees for feature learning and segmentation. In *German Conference on Pattern Recognition*, pages 95–106. Springer, 2014.

[229] Samuel Rota Bulo and Peter Kontschieder. Neural decision forests for semantic image labelling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 81–88, 2014.

[230] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *ICCV*, pages 1467–1475, 2015.

[231] Nicholas Frosst and Geoffrey E Hinton. Distilling a Neural Network Into a Soft Decision Tree. *CoRR*, 2017.

[232] Han Xiao. NDT: Neual Decision Tree Towards Fully Functioned Neural Graph. *arXiv preprint arXiv:1712.05934*, 2017.

[233] Sethu Vijayakumar and Stefan Schaal. Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In *ICML*, volume 1, pages 288–293, 2000.

[234] 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.

[235] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[236] Zhi-Hua Zhou and Ji Feng. Deep Forest: Towards An Alternative to Deep Neural Networks. In *IJCAI*, 2017.

[237] Natalia Ponomareva, Thomas Colthurst, Gilbert Hendry, Salem Haykal, and Soroush Radpour. Compact multi-class boosted trees. In *International Conference on Big Data*, pages 47–56, 2017.

[238] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 1994.

[239] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[240] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. In *CVPR*, 2015.

[241] Yani Ioannou, Duncan Robertson, Darko Zikic, Peter Kontschieder, Jamie Shotton, Matthew Brown, and Antonio Criminisi. Decision forests, convolutional networks and the models in-between. *CoRR*, 2016.

[242] Ozan İrsoy, Olcay Taner Yildiz, and Ethem Alpaydın. Budding trees. In *ICPR*, pages 3582–3587. IEEE, 2014.

[243] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in Neural Information Processing Systems*, pages 234–242, 2013.

[244] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[245] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[246] Han Zhao, Otilia Stretcu, Renato Negrinho, Alex Smola, and Geoff Gordon. Efficient Multi-task Feature and Relationship Learning. *arXiv preprint arXiv:1702.04423*, 2017.

[247] Samuel Schulter, Paul Wohlhart, Christian Leistner, Amir Saffari, Peter M Roth, and Horst Bischof. Alternating decision forests. In *CVPR, 2013*, 2013.

[248] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[249] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[250] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*, 2013.

[251] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014.

[252] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, 2015.

[253] 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.

[254] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[255] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[256] Scott E Fahlman and Christian Lebiere. The cascade-correlation learning architecture. In *Advances in neural information processing systems*, pages 524–532, 1990.

[257] Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry P. Vetrov, and Ruslan Salakhutdinov. Spatially Adaptive Computation Time for Residual Networks. *CVPR*, pages 1790–1799, 2017.