UNIVERSITY OF LEEDS

# Visual bias mitigation driven by Bayesian uncertainties

Rebecca S. Stone

Submitted in accordance with the requirements for the degree of
PhD Computer Science

Faculty of Engineering

School of Computing

April 2024

# Intellectual Property

The candidate confirms that the work submitted is her own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

© 2024 The University of Leeds, Rebecca S. Stone

Signed

# Acknowledgements

Firstly, I would like to thank my supervisors Professors David Hogg and Andy Bulpitt at the University of Leeds, for supporting and encouraging me from day one and throughout my studies: from encouraging me to begin the PhD, to changing my research direction, to taking leave for each of my two girls, to navigating through the pandemic. Andy has been particularly helpful with administrative matters and eye for details. And from research to teaching, I have learned so much from David with his attention to detail, integrity, and the opportunities he has given to learn from and work with him.

A big thank you to Nishant Ravikumar for teaching me about research, giving his time generously to talk through problems together, and encouraging me. I am grateful as well to Sharib Ali, who found an opportunity to collaborate, and from whom I have also learned much. I would also like to thank my fellow PhD candidates - Peter, Hanh, and Leo for being good friends - and especially Jose and Mohammed for the many conversations about research and life.

Training for this research was carried out on ARC3 [1] from the research computing cluster at the University of Leeds, and Bede, a facility of the N8 Centre of Excellence in Computationally Intensive Research (N8 CIR) [2], provided and funded by the N8 research partnership and EPSRC (Grant No. EP/T022167/1). The Centre is co-ordinated by the Universities of Durham, Manchester and York. The team at both ARC3 and Bede have been supportive and helpful throughout, with special thanks to Martin Callaghan and his colleagues.

A huge thanks to all my family, for supporting me in many different ways. And last but not least, thank you to my husband Nathanael, for not letting me give up, for helping me as best he could amidst many full days and sleep-deprived nights, and for always reminding me how the PhD began and how it would end: with God.

---

[1] https://arc.leeds.ac.uk/index.html

[2] https://n8cir.org.uk/bede/

# Abstract

Today, intelligent models are used in applications across all of society, from recidivism prediction, identity verification, a vast collection of healthcare tasks from polyp segmentation to cancer grading, and information retrieval, among many more. The vast majority of these intelligent models are variants of deep neural networks trained on large real-world datasets. These datasets reflect our historical and societal biases; in turn, AI learns these correlations during training resulting in predictors and decision-makers exhibiting racism, ableism, sexism and other forms of prejudice.

Visual data contains many potential biases given its richness of features, and the research related to developing fair vision models is a challenging, open problem. In particular, the sub-problem of implicit mitigation – or mitigation when the knowledge of bias sources in the training or testing data is unknown – is relevant to many use cases where metadata for datasets is difficult to collect. This work contributes to this domain of research by leveraging the observation that bias-conflicting samples, or input samples which are not aligned with the majority correlations, tend to have higher uncertainties in the Bayesian paradigm. By using Bayesian deep neural networks, we can both maintain the performance capabilities of a deterministic network, while gaining access to principled uncertainty estimates. Model uncertainties or epistemic uncertainties in particular provide direct insight into the training data distribution and bias landscape, as bias-conflicting samples are under-represented.

We explore two novel strategies driven by the uncertainties of a Bayesian neural network. The first dynamically re-weights samples as a function of their predictive uncertainty estimates during training, encouraging the model to focus on the more difficult bias-conflicting samples. The second approach fine-tunes the posterior estimate of a converged Bayesian neural network, using the uncertainties to adjust the estimates in favour of fairer predictions. The potential of these methods for implicit visual bias mitigation is demonstrated on benchmark classification tasks and then extended to a medical image segmentation problem with known generalisability issues. Our research, while far from a solution to the bias problem, shows potential for improving model fairness and generalisability and contributes to the literature in this challenging domain.

# Contents

**1 Introduction** **1**
1.1 Contributions . . . . . 2
1.2 Thesis outline . . . . . 3

**2 Background** **5**
2.1 Overview . . . . . 5
2.2 Bias: sources and implications . . . . . 7
  2.2.1 Bias from humans and their data . . . . . 8
  2.2.2 Algorithmic bias . . . . . 13
2.3 Bias mitigation in deep learning . . . . . 14
  2.3.1 Dataset collection . . . . . 14
  2.3.2 Augmented and synthetic data . . . . . 16
  2.3.3 Algorithmic mitigation: explicit . . . . . 17
  2.3.4 Algorithmic mitigation: implicit . . . . . 21
2.4 Visual bias challenges . . . . . 25

**3 An uncertainty-weighted loss for bias mitigation** **27**
3.1 Overview . . . . . 27
3.2 Bayesian deep learning . . . . . 27
  3.2.1 The Bayesian paradigm . . . . . 28
  3.2.2 The Bayesian neural network . . . . . 29
  3.2.3 Inference methods . . . . . 32
  3.2.4 Challenges . . . . . 35
3.3 Uncertainties . . . . . 36
  3.3.1 Aleatoric . . . . . 36
  3.3.2 Epistemic . . . . . 36
  3.3.3 Uncertainty disentanglement . . . . . 37
3.4 Uncertainties and bias-conflicting samples . . . . . 37
3.5 Uncertainty-weighted loss . . . . . 39
  3.5.1 Loss-weighting in literature . . . . . 39
  3.5.2 Method . . . . . 39
3.6 Experiments . . . . . 41

3.6.1 Datasets . . . 41
3.6.2 Evaluation metrics . . . 47
3.6.3 Inference . . . 48
3.7 Results . . . 49
3.8 Discussion and conclusion . . . 52

**4 Posterior estimate fine-tuning for bias mitigation 54**
4.1 Overview . . . 54
4.2 Posterior estimate fine-tuning . . . 54
4.2.1 Exploring uncertainty discrepancies . . . 54
4.2.2 Method . . . 59
4.3 Experiments . . . 62
4.3.1 Datasets . . . 62
4.3.2 Evaluation metrics . . . 65
4.3.3 Group predictive uncertainties . . . 67
4.3.4 Inference . . . 68
4.4 Results . . . 69
4.4.1 Bias mitigation evaluation . . . 69
4.4.2 Ablation studies . . . 73
4.5 Hypotheses related to the regularizing loss . . . 73
4.5.1 Hypothesis 1: The model over-fits to minority samples . . . 74
4.5.2 Hypothesis 2: Bias-conflicting sample uncertainties are being driven to zero 75
4.6 Discussion and conclusion . . . 76

**5 Extension to polyp segmentation generalisability 78**
5.1 Overview . . . 78
5.2 Background . . . 79
5.2.1 Clinical motivation and context . . . 79
5.2.2 Generalisability challenges . . . 81
5.3 Method . . . 82
5.3.1 Semantic segmentation framework . . . 82
5.3.2 Bayesian DeepLabV3+ . . . 83
5.3.3 EpiUpWt: uncertainty-weighted cross entropy loss for segmentation . . . 84
5.3.4 $L_{RCE}$: Fine-tuned posteriors for segmentation . . . 85
5.4 Experiments . . . 85
5.4.1 Datasets . . . 85
5.4.2 Inference . . . 87
5.5 Results . . . 88
5.5.1 Predictive uncertainties . . . 88
5.5.2 Generalisability evaluation . . . 88
5.5.3 Bias mitigation evaluation . . . 92
5.6 Discussion and conclusion . . . 93

**6 Conclusion** **97**

**References** **99**

**Appendix** **114**

# List of Figures

2.1 Taiwanese New Zealander gets his proposed passport photo repeatedly rejected by the AI-powered photo-checking system for eyes being “closed" [158]. . . . . . . 6
2.2 Correlations were learned by Google Vision Cloud between skin color and a weapon. 7
2.3 A CNN learned a correlation between females and smiling, finding Marie Curie in the famous 1911 Solvay Conference appears “angry" compared to the “neutral" expressions of most of the men [11]. . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.4 The AI-powered COMPAS recidivism predictor was shown to be biased against people of color. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
2.5 An example of historical biases embedded into public datasets: Google search engine’s autosuggest results when searching the phrase “why are black people so," on January 25, 2013 [122]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
2.6 In 2019 ImageNet classes including people were re-balanced. . . . . . . . . . . . 12
2.7 68 key-points, used to localize 19 facial landmarks (large labelled dots) for computing craniofacial measurements for the 10 coding schemes used in the Diversity in Faces dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.8 The distribution across geographical regions of eight facial datasets. . . . . . . . 14
2.9 The distribution across skin phenotypes of three facial datasets. . . . . . . . . . 15
2.10 The stylization of images from Stylized-ImageNet to deal with the textural bias learned by CNNs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.11 A multiple head de-biasing architecture proposed by Alvi et al. learns the primary task while becoming invariant to bias variables. . . . . . . . . . . . . . . . . . 20
2.12 The latent space of the debiasing variational auto-encoder proposed by Amini et al. implicitly learns the sparser areas of the input feature space. . . . . . . . . . 22
2.13 Nam et al. train two networks simultaneously, with one learning the easier samples first, and the second focusing on the more difficult. . . . . . . . . . . . . . . . 23
2.14 OccamNets proposes an adaptable architecture which allows the network to favor simpler solutions when needed, for bias mitigation. . . . . . . . . . . . . . . . . 24

3.1 As depicted by Gal et al. [53], an arbitrary function $f(x)$ as a function of input data $x$ *(left)* and the softmax, $\sigma(f(x))$ as a function of input $x$ *(right)*. . . . . . . 28

3.2 Figure from [81] showing (a) a deterministic point estimate neural network, (b) stochastic ANN with probability distributions over activations, and (c) stochastic NN with distributions over all weights. . . . . . . . . . . . . . . . . . . . . . . . 30
3.3 Figure from [179] showing the cyclical stepsize schedule (red) compared against the traditional decreasing stepsize schedule (blue) for SG-MCMC algorithms. . . 35
3.4 The weighting scalar of $L(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is a function of sample uncertainty $\sigma_{i,y_i}$ and scalar $\kappa$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
3.5 A sampling of 100 images, 10 from each class from our minority class bias benchmark CIFAR-10M with $p = 0.9$ as an example of minority group bias equally skewed across all classes. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.6 A sampling of 100 images, 10 from each class from CIFAR-10S [167] with $p = 0.9$ as an example of sensitive attribute bias equally distributed across the dataset but skewed across classes. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
3.7 FairFace [84] is balanced across 7 racial groups: White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, and Latino. . . . . . . . . . . . . . . . . . . . 45
3.8 FairFace [84] is also balanced across seven age groups, from “0-2", “3-9", “10-19", “20-29", “30-39", “40-49", “50-59", “60-69", to “more than 70". . . . . . . . . . . . 46
3.9 The distribution of faces across region and gender for two major facial recognition datasets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
3.10 A random selection of 16 CelebA images before transformation *(left)* and the same images *(right)* after applying transformation $T$ which minimizes the average FID score between the CelebA and FairFace datasets. . . . . . . . . . . . . . . . . . 47
3.11 Performance discrepancies between baseline (deterministic model with no debiasing) and Bayesian model with uncertainty-weighted loss for minority subgroups with lowest TPRs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
3.12 True positive rate (TPR) over the entire FairFace dataset as a function of tunable de-biasing parameter kappa $\kappa$, showing how the degree of de-biasing can be controlled by $\kappa$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
3.13 A grid of images from the CelebA training set showing higher predictive uncertainty images are subject to various unlabeled biases. . . . . . . . . . . . . . . . 52

4.1 Four samples based on a single generated Synbols image from class “s" of our toy dataset, each with a different controllable bias variable *(left to right)*: the letter itself, spurious square placed in random corner, resolution of whole image, and gray-scale. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.2 The representation component of a typical CNN architecture (*left*), resulting in flattened features which feed into the classification component (*right*) composed of several linear layers. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.3 The top six training images for which kernel 10, convolution layer 2 from AlexNet most strongly activates, showing that it has learned the spurious feature. . . . . . 56

4.4 The top six training images for which four kernels in the second convolutional layer of AlexNet most strongly activate. . . . . . . . . . . . . . . . . . . . . . . . 57
4.5 Intersection-over-union (IoU) of each kernel $k$ in each of the 5 convolutional layers of the Bayesian AlexNet, plotted against the kernel uncertainties. . . . . . . . . . 58
4.6 The interaction between the two loss gradients $\Delta L_{CE}$ and $\Delta L_R$ under PCGrad [177]. 61
4.7 Bias-aligned samples from Biased MNIST training set with $p_{bias} = 0.95$ (*left*), and uncorrelated, bias-conflicting samples from the test set (*right*). . . . . . . . . 63
4.8 Samples from the training set of COCO-on-Places where foreground objects are all correlated with a background category; the "dog" object is paired with the "staircase" background. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
4.9 COCO-on-Places in-distribution test set samples from the "dog" class, with the same "staircase" background pairing as in 80% of occurrences in the training set. 64
4.10 COCO-on-Places out-of-distribution test set samples from the "dog" class, where foreground objects are paired with backgrounds from categories which do not appear at all in the training set. . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
4.11 COCO-on-Places systematic shift test set samples from the "dog" class, where foreground objects are paired with previously seen but not correlated backgrounds; i.e., not paired with "staircase" backgrounds. . . . . . . . . . . . . . . . . . . . 65
4.12 BAR training samples from the "climbing" action class (*left*) correlated with structural backgrounds and the test set (*right*) with mostly outdoor snowy mountains. 66
4.13 BAR training samples from the "racing" action class (*left*) correlated with race tracks and the test set (*right*) with more rugged terrain. . . . . . . . . . . . . . . 66
4.14 BAR training samples from the "throwing" action class (*left*) correlated with organized sporting fields and the test set (*right*) with urban scenes. . . . . . . . 67
4.15 Mean uncertainties for subgroups of each test dataset, showing bias-induced discrepancies, but also increased uncertainties arising from a variety of other sources, some of which may be unintentional bias present in the data. . . . . . . . . . . 68
4.16 The mean minority subgroup accuracies of each bias variable in the training, validation, and test sets of Biased MNIST suggesting that $L_{CE}$ overfits more than PCGrad$(L_{CE}, L_R)$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
4.17 The mean minority subgroup uncertainties of each bias variable in the training, validation, and test sets of Biased MNIST suggesting that using $L_{CE}$ alone does not drive the bias-conflicting sample uncertainties to zero. . . . . . . . . . . . . 75

5.1 Polyp segmentation during a colonoscopy must occur under challenging conditions. 80
5.2 DeepLabV3+ from [32, 31], the encoder-decoder architecture with atrous convolutions at different rates (filter step-sizes) for capturing multi-scale contextual information and more effective boundary delineation. . . . . . . . . . . . . . . . 82
5.3 Pixel-wise weighting of cross entropy (CE) loss contribution based on predictive uncertainty maps for each training sample; the model is encouraged to focus on regions for which it is more uncertain. . . . . . . . . . . . . . . . . . . . . . . . 84

5.4 cSG-MCMC with a choice of smaller $M$ for comparable performance and uncertainty estimates and lower resource requirement: selecting moments with maximized intervals during the sampling phase. Note that using this sampling size under the regular criteria would mean sampling in the last three epochs of the sampling phase. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.5 Ali et al. presents the 2D t-SNE embedding of the PolypGen training set from deep autoencoder extracted features. . . . . . . . . . . . . . . . . . . . . . . . . . 87
5.6 Samples from the PolypGen demonstrate the variability of sequential data due to the presence of artifacts, occlusions, and polyps with different morphology. . . . . 88
5.7 Frames with ground truth segmentation masks and corresponding predictive uncertainty masks from Center 6 sequence data. . . . . . . . . . . . . . . . . . . . . 89
5.8 Performance gaps of the three models (state-of-the-art deterministic DeepLabV3+, BayDeepLabV3+, BayDeepLabV3+EpiUpWt, BayDeepLabV3+$L_{RCE}$) between the three different test sets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
5.9 Results for the bias mitigation experiments evaluated on the three test datasets. . 94
5.10 Frames with ground truth segmentation masks and corresponding predictive uncertainty masks from Center 1-5 sequence data. . . . . . . . . . . . . . . . . . . 96

6.1 Mean uncertainties of each bias variable in Biased MNIST after fine-tuning with $L_R$ loss alone. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
6.2 Baseline Bayesian BayDeepLabV3+ results for the bias mitigation experiments evaluated on the three test datasets. . . . . . . . . . . . . . . . . . . . . . . . . . 116

# List of Tables

3.1 Notation for an artificial neural network. . . . . . . . . . . . . . . . . . . . . . . . . 29
3.2 Notation for a Bayesian neural network. . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.3 Distribution skew for age, gender, and skin color in the CelebA [107] face image dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
3.4 Our novel uncertainty-weighted bias mitigation method compared with other benchmarks for multi-class classification. . . . . . . . . . . . . . . . . . . . . . . . 49
3.5 The uncertainty-weighted loss reduces the true positive disparity on minority bias dataset CIFAR-10M. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
3.6 Ablation study showing results of a Bayesian cSG-MCMC network with regular unweighted cross-entropy loss. . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

4.1 Notation for a Bayesian neural network dissection. . . . . . . . . . . . . . . . . . 55
4.2 Notation for the fine-tuning procedure. . . . . . . . . . . . . . . . . . . . . . . . . . 60
4.3 The COCO-on-Places composition with unbalanced foreground and background pairings. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
4.4 The Biased Action Recognition dataset composition with unbalanced background and action pairings. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
4.5 Grid search parameters and value ranges with increment size for the hyper-parameter fine-tuning of the baseline Bayesian models. . . . . . . . . . . . . . . . 69
4.6 Unbiased test set accuracies comparing the fine-tuned BayResNet against current debiasing methods. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
4.7 Accuracies for each of the three test splits of COCO-on-Places. . . . . . . . . . . 70
4.8 Accuracies on majority (maj)/minority (min) groups and bias-induced accuracy gaps for each bias variable in BiasedMNIST. . . . . . . . . . . . . . . . . . . . . 71
4.9 Ablation study of fine-tuning procedure losses showing performance on the most challenging test sets for each dataset. . . . . . . . . . . . . . . . . . . . . . . . . 71
4.10 Overall and per-class accuracies on BAR, showing large discrepancies in performance across methods and classes. . . . . . . . . . . . . . . . . . . . . . . . . . 72

5.1 The PolypGen training and three test set image counts, modalities, and collection centres. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86

5.2 Evaluation of the state-of-the-art deterministic DeepLabV3+, BayDeepLabV3+, and our proposed BayDeepLabV3+EpiUpWt, and BayDeepLabV3+EpiUpWt+$L_{RCE}$, showing mean and standard deviations across the respective test dataset samples. 91
5.3 Minority/majority subgroup ratios for the bias mitigation experiments. . . . . . . 93

6.1 Numeric values for the true positives rate gaps for the FairFaces test set. . . . . . 115
6.2 Hyperparameters chosen for each reported experiment in Chapter 4. . . . . . . . 116
6.3 Bias mitigation experiment results for the state-of-the-art deterministic DeepLabV3+ for different number of in-distribution samples (IDS) which for the minority group added into the training data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
6.4 Bias mitigation experiment results for the baseline Bayesian model, BayDeepLabV3+ for different number of in-distribution samples (IDS) which for the minority group added into the training data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
6.5 Bias mitigation experiment results for BayDeepLabV3+EpiUpWt for different number of in-distribution samples (IDS) which for the minority group added into the training data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
6.6 Bias mitigation experiment results for BayDeepLabV3+$L_{RCE}$ for different number of in-distribution samples (IDS) which for the minority group added into the training data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118

# Notation

The following list describes the notation used throughout this thesis. The terms are also present near the sections where they appear in the thesis for reference. As per standard notation, scalars are represented in italics as $a$, whereas vectors or matrices of larger dimensions are represented in bold $\boldsymbol{a}$.

| **Parameter** | **Definition** |
|---|---|
| $\boldsymbol{\theta}$ | The set of all learnable parameters of a neural network |
| $\boldsymbol{\Phi}_\theta$ | The function learned by a neural network with parameters $\boldsymbol{\theta}$ |
| $\boldsymbol{x}$ | The input to the network |
| $b$ | A scalar bias parameter included in $\boldsymbol{\theta}$ |
| $h()$ | An activation function |
| $\mathbb{W}$ | The weights of a single layer included in $\boldsymbol{\theta}$ |
| $D$ | A dataset of input and target output pairs, $(X, Y) = (\boldsymbol{x}_i, \boldsymbol{y}_i), i \in 1..N$ |
| $p(\boldsymbol{\theta} \mid D)$ | The posterior distribution |
| $p(\boldsymbol{\theta})$ | The prior distribution |
| $M$ | The Monte Carlo sampling size from the posterior for the posterior estimate of a Bayesian neural network |
| $\boldsymbol{\Theta}$ | The set of posterior samples comprising the posterior estimate $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ...\boldsymbol{\theta}_M\}$ |
| $\boldsymbol{\mu}_i$ | The predictive mean associated with the $i^{th}$ input |
| $\hat{\boldsymbol{y}}_i$ | The class prediction associated with the $i^{th}$ input |
| $\boldsymbol{\sigma}_i$ | The predictive uncertainty associated with the $i^{th}$ input |
| $\hat{A}_k(\boldsymbol{x})$ | The pixel-wise mean activation map across all posterior estimates for the $k^{th}$ convolutional kernel |
| $f$ | An identified feature in the dataset |
| $IoU_{k,f}$ | The intersection-over-union for the $k^{th}$ convolutional kernel and feature $f$ |
| $t$ | The threshold for determining whether kernel $k$ is a high-activator for feature $f$ |
| $\sigma_k(\boldsymbol{x})$ | The kernel uncertainty, the maximum variance across activation maps |
| $\boldsymbol{\theta}^r$ | The parameters of the representation component of the network |
| $\boldsymbol{\theta}^c$ | The parameters of the classification component of the network |
| $\boldsymbol{\Phi}_{\boldsymbol{\theta}_{\boldsymbol{m}}}$ | The $m^{th}$ posterior estimate |

# Chapter 1

# Introduction

Data-driven intelligent models - models which learn from data - have been at the forefront of artificial intelligence advances for the past several decades. In the past decade, largely thanks to the success of neural networks, increasingly impressive accomplishments have captured the attention of the general public: from the classical vision problems including image classification, segmentation, and general understanding, to style transfers [56, 80], image and video generation from text – at qualities beyond what the human eye can detect as fake [129, 135] – AI-enhanced devices [125, 149], self-driving vehicles [168], and most recently the large language models (LLMs) [128, 21] such as ChatGPT, Bard, and others. Intelligent models have become part of our everyday lives, whether we realize it or not.

Yet, barely keeping up with the constant advances of data-driven AI models are the skeptics, the policy makers, the AI practitioners, the educators, and the general public. In March 2023 the controversial letter titled, “Pause Giant AI Experiments: An Open Letter" [124] called for a temporary pause in research developing LLMs, and stirred up a controversy with its famous signatories and opponents passionately defending the future of AI (notable proponents including AI pioneers Yoshua Bengio and Andrew Yang, and objectors Andrew Ng and Yann LeCun among others). Regardless of the merits of each stance, the traction gained by such a letter is indicative of the fact that such models may pose serious risks to society if their shortcomings and the implications of failure are not fully understood.

All neural network-based models learn from data. Neural networks are essentially multi-variate, powerful function approximators which optimize given a cost function and data. The learning is guided by data - *data* - not contextual understanding of the task, not risk awareness, not empathy, and certainly not any internal moral or ethical compass. The data can be trillions of word-text pairs, millions of images, hundreds of thousands of annotations and labels, and petabytes of video clips. At best, this data is a true reflection of human society; at worst, it is a skewed subset which represents the reality for only a very few. Even assuming the dataset is an accurate reflection of society, our society unfortunately *is* extremely biased with innumerable social, economic, ethnic, age, and gender stereotypes. As Mehrabi et al. [112] comprehensively present, datasets are user-generated, and inherent biases in users are often reflected in their data, whether the biases are introduced via sampling processes, populations, social influences, or user behaviour.

Furthermore, models do not know when they are wrong. There is no absolute truth; only the data they have seen, and what patterns are statistically confirmed by that data. There exists an underlying distribution from which that data was sampled, but no guarantee of the sampling procedure. A 2-D linear regression model cannot be faulted for wrongly classifying a test data point far from the range of data it has seen, nor for wrongly classifying a test data point when a portion of its similar training data was misleading. Data-driven neural networks can and do learn misrepresentations of their target tasks - largely thanks to spurious correlations present in the data they were shown. Furthermore, they can be wrong not only in the ground-truth sense; they can also be wrong due to more complex ethical and moral reasons which cannot always be quantified.

Computer vision is particularly susceptible to biases due to the wealth of information present in visual data. This work focuses on bias mitigation in computer vision, whereby deep neural networks are trained in a way that discourages them from learning spurious correlations present in the data they learn from. While we focus specifically on the fundamental tasks of image classification and segmentation, the approaches explored could also be applied to other visual tasks and even to some non-visual tasks.

## 1.1 Contributions

There is a critical need for a better understanding of the biases that pervade large datasets, especially vision datasets, and for developing algorithmic bias mitigation methods. In our research, we argue that the uncertainties of probabilistic models, in particular, Bayesian neural networks which can provide uncertainty estimates in addition to state-of-the-art performance, can be leveraged for the difficult bias mitigation problem. Using the under-explored correlation between Bayesian predictive uncertainties and minority, or bias-conflicting, training samples, we propose two novel mitigation methods leveraging Bayesian uncertainties in deep neural networks. We then modify the methods and demonstrate applicability to a medical imaging segmentation problem.

Some of the thesis contributions can be found in the following peer-reviewed publications:

- Chapter 3: **Stone, R.S.**, Ravikumar, N., Bulpitt, A.J. and Hogg, D.C., 2022. Epistemic uncertainty-weighted loss for visual bias mitigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2898-2905). Workshop on Fair, Data-Efficient, and Trusted Computer Vision 2022.
- Chapter 5: **Stone, R.S.**, Chavarrias-Solano, P.E., Bulpitt, A.J., Hogg, D.C. and Ali, S., 2023. Bayesian uncertainty-weighted loss for improved generalisability on polyp segmentation task. arXiv preprint arXiv:2309.06807. *Volume 14242 of the Lecture Notes in Computer Science, Clinical Image-Based Procedures, Ethical and Philosophical Issues in*

*Medical Imaging, and Fairness of AI in Medical Imaging, MICCAI Workshop 2023.*

And presented at the following meeting:

- Chapter 3: **Stone, R.S.**, Ravikumar, N., Bulpitt, A.J. and Hogg, D.C., 2022. Epistemic uncertainty-weighted loss for uninformed visual bias mitigation. The British Machine Vision Association (BMVA) Symposium, 2022.

The second publication includes contributions from two collaborators: Pedro E. Chavarrias-Solano, who helped process part of the dataset used and generated some figures for the published paper, and Sharib Ali, first author of the papers presenting the PolypGen dataset and presenting benchmarks from the challenge hosted around it [4, 6], who contributed by proposing the use of the dataset, providing information and guidance related to the data, and providing the code repository for the baselines.

## 1.2 Thesis outline

Bias mitigation approaches fall in one of two broad categories: explicit and implicit. Explicit mitigation methods explicitly use and leverage the knowledge of bias sources in the training data in order to de-bias the model. Conversely, implicit methods make no assumptions, and require no bias-related annotations. In this research we do not address issues related to bias benchmark datasets or how to identify and detect biases. While our contributions may be applicable to non-vision domains, we do not address this in the scope of our work. Rather, we present and contribute towards the challenging problem of implicit visual bias mitigation in supervised vision tasks, proposing two novel ways to leverage Bayesian uncertainties in deep neural networks for this purpose.

First, in Chapter 2 we present a background of the significance and implications of the visual bias problem, a survey of the literature of mitigation methods generally across artificial intelligence but with a focus on vision models.

In Chapter 3 we then present Bayesian deep learning as an uncertainty-aware alternative for deep neural networks, and consider the different types of uncertainties and their uses. Finally, we present the correlation between bias-conflicting samples and predictive uncertainties as seen in the literature. Based on this correlation, we propose a simple, novel uncertainty-weighted loss function leveraging the predictive uncertainty and bias correlation, applied to image classification.

Chapter 4 builds on this with a different bias mitigation method which is applied post-training as a fine-tuning procedure to modify the posterior estimate of a Bayesian neural network. The sharpening loss objective motivates the network to focus less on high uncertainty-inducing features, and rather on core features which do not contribute to uncertainty fluctuations. We demonstrate the strengths and weaknesses of this strategy through experiments on several datasets.

In Chapter 5, we demonstrate how the previous two mitigation methods can be modified from classification to an image segmentation task, and then apply them to the challenging, open-ended medical imaging problem of polyp segmentation, where state-of-the-art models are known to struggle with generalisability.

Chapter 6 concludes with a final discussion of the key contributions, limitations of the methods explored, and the potential for further work opened up by our research.

# Chapter 2

# Background

## 2.1 Overview

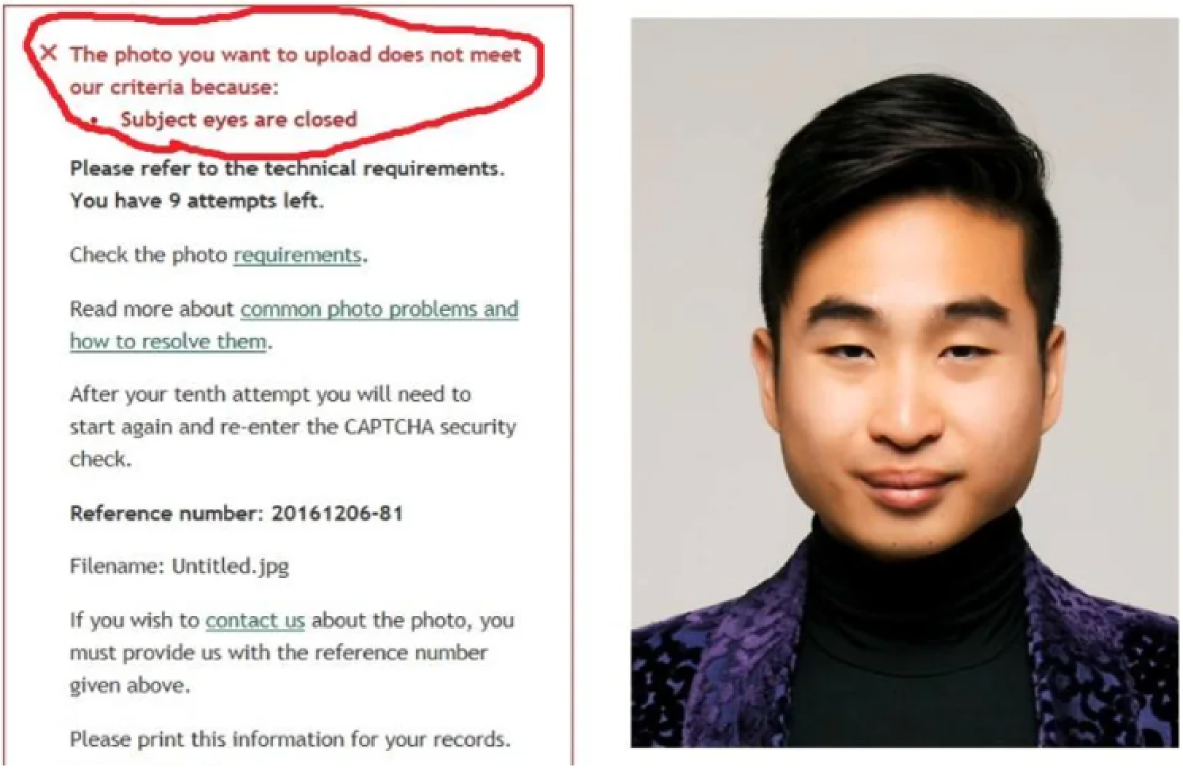
In 2015, Google rolled out an AI-powered “Assistant", an automatic photo tagging feature in its well-used Photos app, giving users the capability to search photos for objects or people and create categories. The feature quickly became widely popular; but only a few months later, users reported concerning results – dark-skinned people were being tagged as “gorillas" [62]. Apologies, promises to do better, and a year later in 2016, as part of Google Cloud Machine Learning API's, Google released Vision Cloud [1] with powerful vision tools including image annotating, object tracking, product search, and more. AlgorithmWatch [2] performed an experiment using simple in-painting over skin colors in images and found biases; simply through the training data, the model believed a dark-skinned subject more likely to be holding a weapon than a light-skinned person [63] (Figure 2.2).

Google, while perhaps with exceptionally large datasets at its disposal, is in no way an exception to the visual bias problem. An AI-powered automatic passport photo checking service used in New Zealand repeatedly rejected Richard Lee's photos due to his eyes being detected as “closed" [158](Figure 2.1). In 2020, researchers found that deep models trained on three of the most widely-used public chest X-ray datasets for research were biased against gender, socioeconomic groups and racial minorities [139], giving lower predictive accuracies for those subgroups, and show corresponding imbalances in the training data. A gender and emotion predictor – a real-time convolutional neural network (CNN) – learned the correlation between the female gender and smiling, notably labelling the men in the 1911 Solvay Conference mostly as having “neutral" emotions, whereas the only female, Marie Curie, is an “angry woman" [11] (Figure 2.3).

---

[1] https://cloud.google.com/vision/

[2] https://algorithmwatch.org/



✕ The photo you want to upload does not meet our criteria because:

- Subject eyes are closed

**Please refer to the technical requirements.**
**You have 9 attempts left.**

Check the photo requirements.

Read more about common photo problems and how to resolve them.

After your tenth attempt you will need to start again and re-enter the CAPTCHA security check.

**Reference number: 20161206-81**

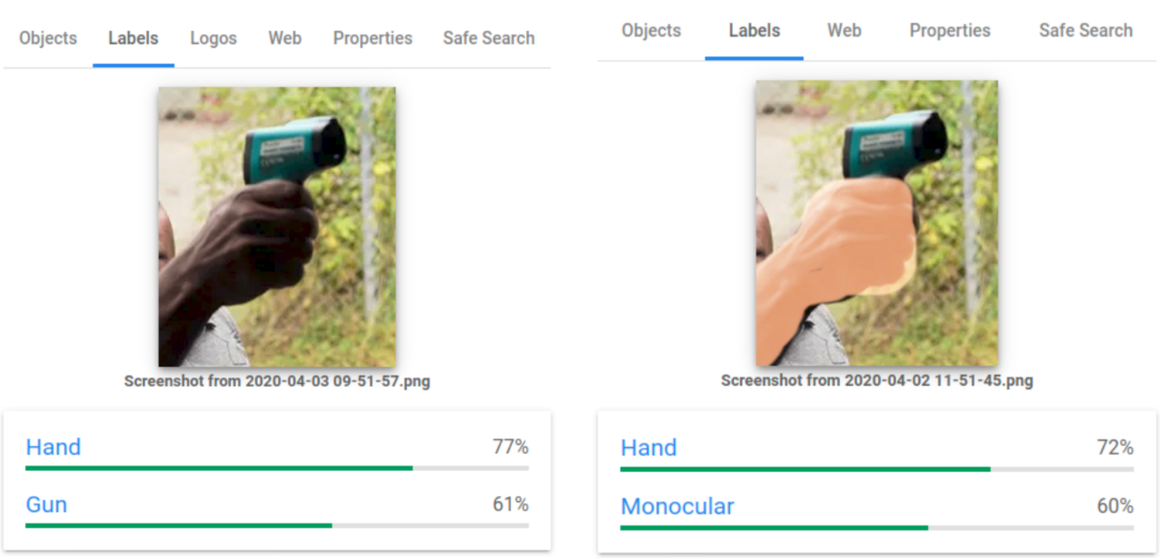
Filename: Untitled.jpg

If you wish to contact us about the photo, you must provide us with the reference number given above.

Please print this information for your records.

Figure 2.1: Taiwanese New Zealander gets his proposed passport photo repeatedly rejected by the AI-powered photo-checking system for eyes being “closed" [158].

Biases in visual datasets are prevalent and pervasive. This chapter presents an overview of biases: (1) their sources, (2) how deep learning models learn them and even systematize and amplify them; and (3) a survey of work in the research community on visual bias mitigation. The first two topics are mostly modality-independent, so we will consider various real-world examples, all relevant to the vision domain. In contrast, the literature survey will focus primarily on visual bias mitigation methods. To conclude, we discuss the present open challenges.



Screenshot from 2020-04-03 09-51-57.png

Hand 77%

Gun 61%

Screenshot from 2020-04-02 11-51-45.png

Hand 72%

Monocular 60%

Figure 2.2: Google Vision Cloud's labelling API annotates the image of a hand holding a thermometer as containing a "hand" and "gun" *(left)*, and the same image but with the hand painted to be light-skinned is a "hand" and "monocular" *(right)*.

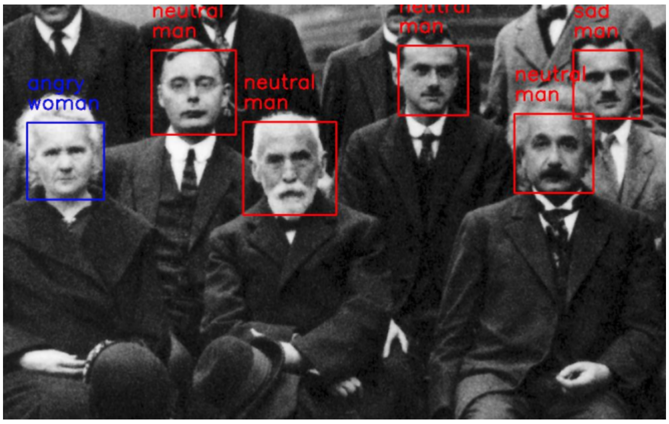


Figure 2.3: A CNN learned a correlation between females and smiling, finding Marie Curie in the famous 1911 Solvay Conference appears "angry" compared to the "neutral" expressions of most of the men [11].

## 2.2 Bias: sources and implications

> "In the context of decision-making, fairness is *the absence of any prejudice or favoritism toward an individual or group based on their inherent or acquired characteristics* [112]; or, for all possible combinations of protected attributes, the probabilities of the outcomes will be similar" – An intersectional definition of fairness, Foulds et al. [52].

Correlations exist in every data set, whether intentional or unintentional. In the simple classification setting, for dataset $D$ with inputs $X = \{x_0, x_1 ... . x_N\}$ and corresponding target classes $Y = \{y_0, y_1 ... y_C\}$, any feature $f$ such that a majority of $x_n \in y_c$ have the same value of $f$ induces a majority bias. Depending on the difficulty of learning $f$, the correlation threshold may vary;

when learning $f$ comes at the expense of learning some other core feature [126], the threshold is lower, whereas in other cases $f$ is not easy to learn so may induce less bias. It may also be noted that not all biases are harmful. For example, a bias in a dataset may never have negative implications in usage and can be defined as a “benign" bias. If, for example, in a dataset of satellite imagery the sky is mostly clear, the bias induced by having very few “cloudy" samples present is a benign bias if we expect to use the resulting model only on sunny days.

In contrast, harmful biases affect the model performance under certain conditions and may result in discriminatory outcomes for certain populations of people. An unfair model, or biased model, is one whose decisions are skewed against a certain sub-group of valid inputs. *Individual bias* is the subset of this scenario, where the skew is against certain individuals.

Though this definition of bias is general, what a fair outcome looks like in any specific application may vary depending on the context and desired outcomes. The outcomes of a biased model, too, vary greatly in application, necessitating a thorough consideration of the intended use-cases and implications.

In this section, we consider the sources from which biases arise. Bias sources fall into three broad categories: biases induced directly by the users generating the data, biases in the data itself, and algorithm-induced or amplified biases [112, 153].

### 2.2.1 Bias from humans and their data

#### Historical bias

Historical bias refers to bias which is historically present in society, and while it may not be a present bias at least in the region of application, may still be reflected in historical data. Many societal biases are present in the world today. In a Google image search for “CEO" in 2018, 95% of the resuling images of CEO’s were men [153], reflecting the fact that there are more male CEO’s than women.

Similarly, in specific neighborhoods in the US, crime has a much higher likelihood than the national average, and there exists a strong correlation between those zip codes and certain demographics. Thus, intelligent models meant to provide risk assessment in criminal sentencing can learn to correlate higher likelihoods of crime to those demographics, putting people of that background at a disadvantage irrespective of the severity of their crime. The Correctional Offender Management Profiling for Alternative Sanction (COMPAS) software [9] used data-dependent machine learning, and found blacks “almost twice as likely as whites to be labeled a higher risk but not actually re-offend" (Figure 2.4). [3] These correlations, from data collected through ideal sampling procedures, are historically and statistically accurate but contain biases that can be very harmful.

---

[3] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

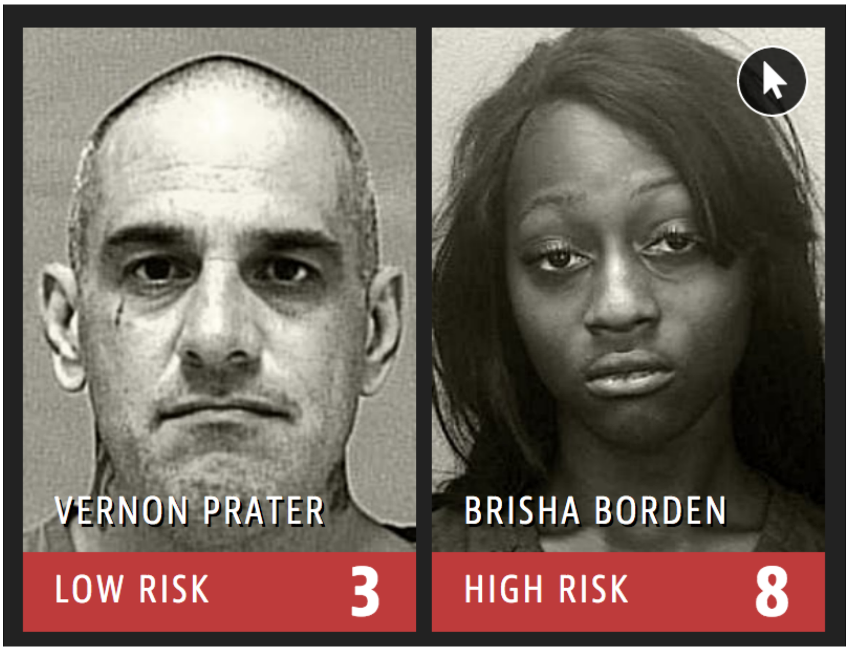
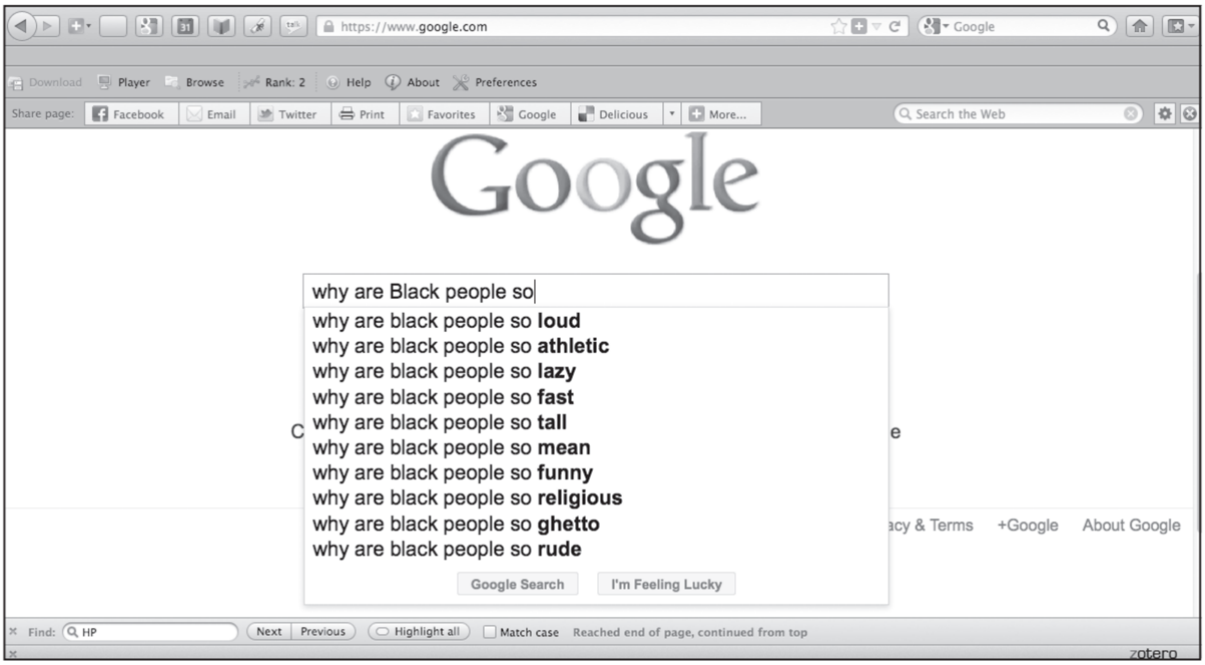


Figure 2.4: The AI-powered COMPAS system rated Borden (*right*) as high risk for future crime, though her initial conviction was for taking a “kid’s bike and scooter that were sitting outside." She never re-offended. Prater (*left*) was rated as low risk, despite being older, having been convicted twice of armed robbery, and having served five years in prison. After release he stole thousands of dollars of electronics from a warehouse. [3]



why are Black people so

why are black people so **loud**
why are black people so **athletic**
why are black people so **lazy**
why are black people so **fast**
why are black people so **tall**
why are black people so **mean**
why are black people so **funny**
why are black people so **religious**
why are black people so **ghetto**
why are black people so **rude**

Figure 2.5: An example of historical biases embedded into public datasets: Google search engine’s autosuggest results when searching the phrase “why are black people so," on January 25, 2013 [122].

> “What we find in search engines about people and culture is important. They oversimplify complex phenomena. They obscure any struggle over understanding, and

> they can mask history. Search results can reframe our thinking and deny us the ability to engage deeply with essential information and knowledge we need, knowledge that has traditionally been learned through teachers, books, history, and experience. Search results, in the context of commercial advertising companies, lay the groundwork, as I have discussed throughout this book, for implicit bias: bias that is buttressed by advertising profits." – Algorithms of Oppression, Safiya Noble [122].

Search engines, the natural source for collecting large-scale data in many domains, are rife with historical biases [122, 111, 85]. Safiya Noble's case studies of bias "Algorithms of Oppression" [122] highlight the algorithmic conceptualizations of categories of people and ideas embedded into the Google search engine, particularly strong around race and sex. Textual searches using Google's auto-completion function such as the search in Figure 2.5 show racist correlations and assumptions. Image searches for "Professor style" returned an overwhelming majority of males in suits, "ugly" returned mostly men, "beautiful" returned mostly white women, and "black girls" returned women of color in heavily sexualized outfits and poses. Makhortykh et al. [111] study in more detail the racial and gender biases in the image search results of the six most popular search engines, including two non-Western engines. They find that while racial diversity is improved for those engines, gendered representations are even stronger. Kay et al. [85] find that for gender and occupation biases - the imbalance of gender ratios for many occupations - image search results actually exaggerate or amplify the existing imbalance ratios; furthermore, their user studies show that users' perceptions of occupational gender proportions after seeing the skewed image search results shifted slightly towards the bias. Despite everything, eye-tracking studies with search engines still confirm that users trust the automated ranking of search results, being biased towards those ranked at the top of the page regardless of relevance [138, 23].

Both search engines and LLMs such as ChatGPT and Bard serve as information intermediaries, filtering and guiding our information seeking process. Biased representations from these intelligent models underscore and perpetuate existing historical bias into the present and future, with dangerous effect.

### Self-selection and sampling bias

Self-selection and sampling bias refer to the bias induced from the populations from which data is collected. Data collection is inherently dependent on user-provided information. The platforms from which data are collected act as a filter for the respondents; collecting data from SnapChat and Instagram means appealing to a generally younger, majority female demographic; soliciting information from Reddit or Twitter includes more men. Any data which comes from a survey is limited to a demographic which is willing to participate - users who have enough time and feel strongly enough about the topic.

Andrej Karpathy conducted a project for personal interest using a CNN to determine the "best" selfie [4]. He scraped publicly available data from social media and concluded that the most

---

[4] http://karpathy.github.io/2015/10/25/selfie/

“popular" selfies, or the ones with the most “likes", were of women with longer hair, and tended to be cropped at around the forehead; an interesting conclusion which says more about the population from which the data was collected than anything else.

Further than just a person’s willingness to provide information, self-selection biases can be proxies for socio-economic factors which contribute to whether users self-select to participate in surveys, polls, or data sharing.

Even when users do not self-select, the data sampling procedure can also be flawed and result in biases. The literature is rife with examples of biased sampling; of 94 widely-used ophthalmology datasets used to train models, only 7 came from regions other than Europe, North America, and China [88]. ImageNet lacks geographical diversity, representing primarily Western cultures. Chest X-ray imaging from three large, prominent datasets under-represents certain insurance types, a proxy for lower economic statuses [139]. Ultimately, both self-selection and sampling biases result in data representative of a sub-group of the true population.

### Labelling bias

Once a dataset has been collected, various forms of annotation may be required for semi-supervised and supervised learning, which can also inadvertently result in biases. ImageNet, the widely-used large-scale dataset of millions of labelled images used as a benchmark in computer vision, uses as labels a vocabulary of categories from WordNet [117], an English language lexical database where each category “synset" is represented by a set of synonyms. In 2009, the ImageNet team queried various search engines and crowdsourced images for each synset. As search engines at the time had no image understanding, the results were based on the textual captions or meta-data associated with each image. Over 50K Amazon Mechanical Turk (MTurk) workers were then hired to remove irrelevant images and to verify labels for retained images.

A decade later, a Princeton research group considered ImageNet from a fairness perspective and took steps to remedy the problems they identified [174]. Firstly, they located and removed over 600K images belonging to WordNet synsets containing offensive words, such as derogatory terms related to sexual orientation and religion. Secondly, they removed synsets which were not necessarily offensive, but were descriptive, rather than definitive. They targeted in particular ethnic or racial categories, such as “Bahamian (a native or inhabitant of the Bahamas)". While most images in the category included people wearing traditional Bahamian costumes, the category by definition is non-imageable, since anyone from the Bahamas belongs, irrespective of their outfit. They used MTurk workers to rate imageability, or how easily each word in a synset brings to mind certain imagery. It is crucial to note here that again, they did not ask MTurk workers to rate how well the image brought to mind represented the *whole* of the synset, meaning that their biases were still reflected in the revised labelling. Lastly, the researchers added images to rebalance certain categories including occupations such as “programmer" (Figure 2.6), “banker", etc., with respect to skin color, gender, and age.

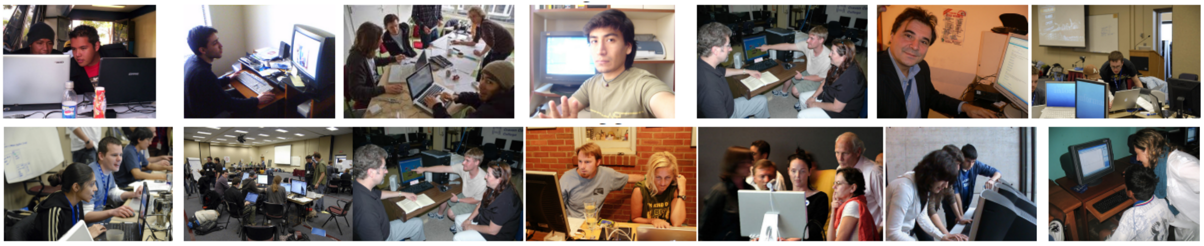


Figure 2.6: Some examples from [174] showing a sampling of images from the original "programmer" class (*above*) and after re-balancing with respect to gender (*bottom*), taken from the 2019 modification to ImageNet towards a fairer dataset.

In a 2021 study, Kate Crawford and Trevor Paglen further analyzed the categorization and labelling by MTurk workers in the "Person" category of ImageNet, finding that there still existed categories for racial groups, sexual preferences, and adjectives describing behaviour including "crazy", "failure", "unskilled person" and "hypocrite", among other examples [39] [5].

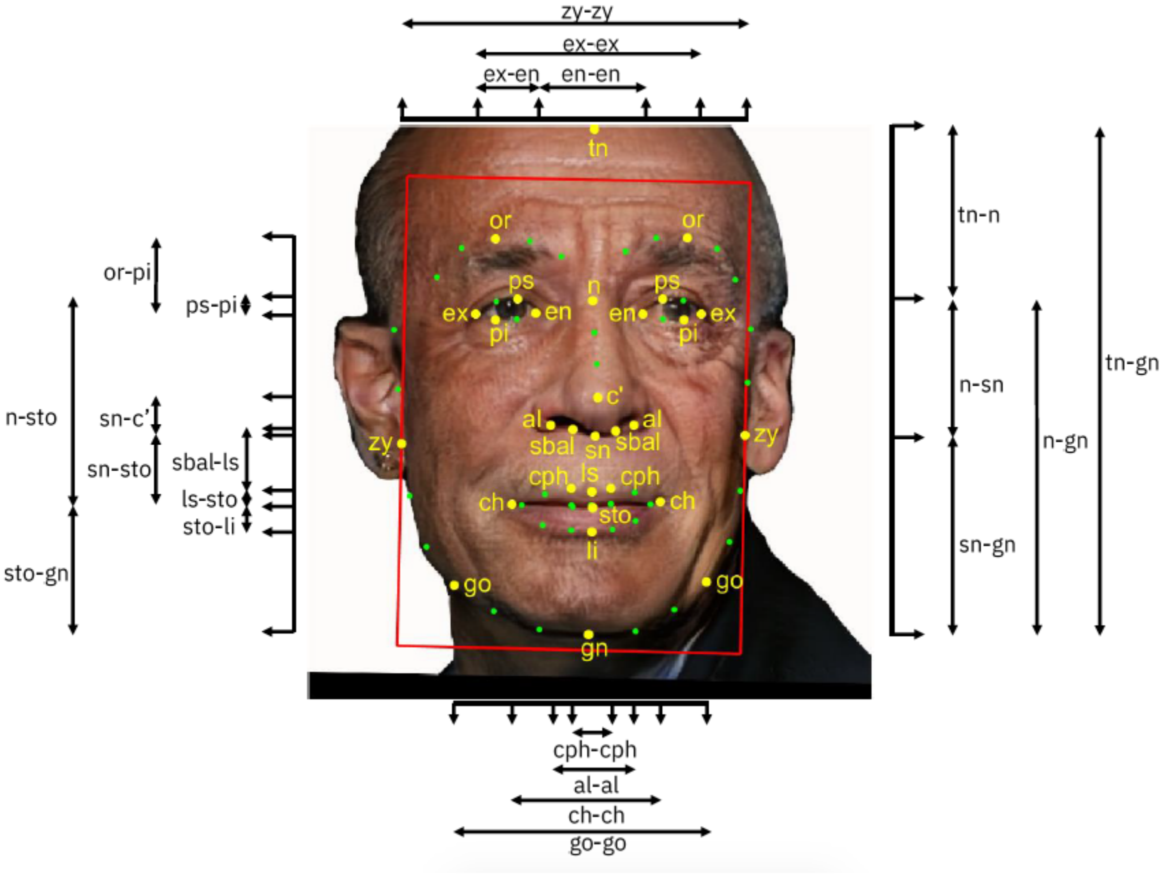


Figure 2.7: 68 key-points, used to localize 19 facial landmarks (large labelled dots) for computing craniofacial measurements for the 10 coding schemes used in the Diversity in Faces dataset [114]. Are facial physiological traits a less biased alternative for determining race than skin color?

Attempts to remove correlations between skin colour and ethnicity have also backfired, showing the difficulty of the task. IBM's Diversity in Faces dataset [114], aimed at improving facial recog-

---

[5] https://excavating.ai/

nition performance on identities of minority skin colour, comprises 1 million face images with a vast array of metadata including craniofacial distances, areas, and ratios, facial symmetries, contrasts, skin color, and other features for a total of 10 facial coding schemes (Figure 2.7).

The motivation for the dataset was that skin color is a poor predictor of ethnicity, and relying on skin color can result in biased models. Yet inadvertently, by replacing skin color with physiological traits, Diversity in Faces posits that correlations of ethnicity to physiological traits are somehow a better, less biased option. What Yang et al. [174] identified, but failed to remove in their 2019 audit of ImageNet, is that certain labels are non-imageable, and asking other humans to think of what features they rely on to categorize people reflects *their* biases. Ethnicity, for example, is essentially more a question of geographical and cultural origins than of physical appearance [22, 39].

These two high-profile examples illustrate how easily historical biases reflected in the data through the collection process can be further emphasized and even worsened through labelling. Avoiding any kind of labelling bias requires very serious and thorough consideration of the data, context, and use cases.

### 2.2.2 Algorithmic bias

#### Model-amplified bias

Algorithmic bias [164, 52, 181] is bias which is caused, amplified or exaggerated by the algorithm itself, due to the choice of algorithm and training process. Algorithmic bias results in worse performance for certain subgroups than the skews present in the training data. Various factors have been shown to determine when and how bias amplification occurs; Hall et al. [64] perform a systematic study on tightly controlled synthetic bias datasets based on MNIST and CIFAR, considering how bias amplification varies as a result of the degree of minority/majority imbalance, model capacity, training set size, model overconfidence, the training period, and the relative difficulty of learning the target class versus the bias subgroup feature. Among other things, they observe that an unbiased training set results in no bias amplification, and that there are optimal ranges with respect to bias amplification for model capacity, model overconfidence, and training set size. The clearest factor by far is the relative difficulty of learning the bias feature - as the bias feature becomes easier to learn, and the target class more difficult, bias amplification increases rapidly. Unfortunately, this factor is the most difficult to control as it is a function of the data itself and not the algorithm. Thus, bias amplification is a complex problem with no single solution fitting every scenario.

Various metrics have been proposed including Bias Amplification [181] measuring model fairness between predictive scores on a test set and the same scores on the training set (under the assumption that both sets have at least a similar if not less biased distribution), and Directional Biased Amplification [164] which disentangles amplification directions. These methods, among others, require knowing the distribution of biases across the training set, so can only be evaluated in an explicit setting.

## 2.3 Bias mitigation in deep learning

### 2.3.1 Dataset collection

Numerous studies suggest better data collection processes to combat the bias problem [144]. Good dataset collection approaches have one of two aims, towards the same end goal:

1. To avoid learning unwanted correlations via gathering a balanced dataset with a sufficient representation of all possible attributes and their combinations; or,
2. To make protected attributes impossible to learn by excluding them completely from the dataset.

A sufficiently diverse population representation is difficult to obtain, especially in the medical imaging domain with vast disparities across geography. In practice, this problem is often tackled by combining datasets from different collection centers, as in [139] who found lung disease prediction fairer across gender, race, and socio-economic subgroups when models were trained on five chest X-ray datasets combined versus when trained on each dataset independently. Furthermore, in polyp segmentation from laparoscopy procedures, models showed decreases in performance every time one medical centre's data was reserved as a hold-out test set and not seen during training [4].

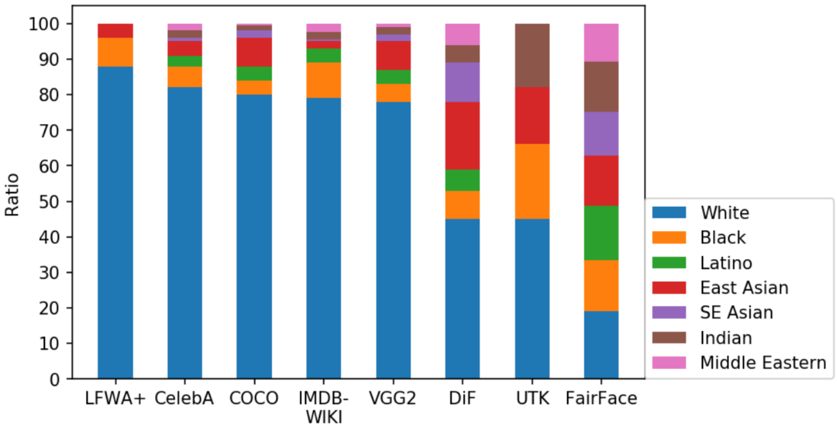


Figure 2.8: Figure from [84] showing the distribution of eight datasets categorized according to the "White", "Black", "Latino", "East Asian", "South East Asian", "Indian", and "Middle Eastern" subgroups with extreme imbalances to the advantage of the "Whites" majority for most of the datasets. FairFaces proposes a facial dataset balanced across what they term as "racial groups".

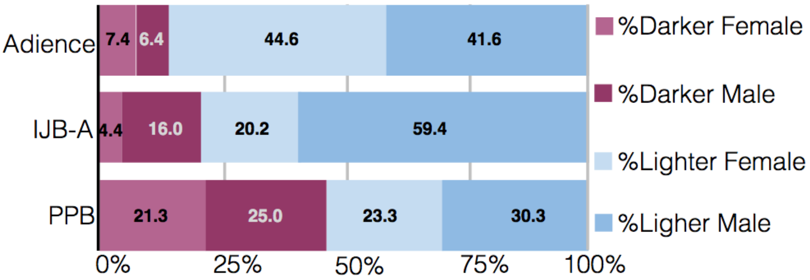


Figure 2.9: Figure from [22] showing the distribution of three datasets when categorized according to skin phenotype: (1) IJB-A for various skin types, (2) Adience for gender and age classification, and (3) PPB showing the benefits of skin phenotypes for balanced dataset compilation. Despite the former two datasets being intended for diverse representation across regions, PPB has the most balanced composition across skin phenotypes.

When possible, datasets with intentional balanced sampling from pre-identified subgroups are preferable. Yet the identification of subgroups is not trivial. For example, the authors of FairFace [84] identify strong bias against non-White racial groups, and define subgroups based on certain geographical regions and dark skin colour. They present a facial dataset with only a slight imbalance towards Whites, and nearly equal presence of “Black”, “Latino”, “East Asian”, “South East Asian”, “Indian”, and “Middle Eastern” subgroups. Figure 2.8 shows how compared to predecessors, FairFace includes a better distribution of samples across racial groups.

Yet the compilers of another facial diversity benchmark, the Pilot Parliaments Benchmark Dataset (PPB Dataset) [22], argue that racial lines cannot be used to subcategorize as one particular racial group; for example “Middle Eastern", is composed of various skin phenotypes, which contains its own majority and minorities. Models trained on balanced ethnic groups may still discriminate towards minorities within those groups. PPB is offered as a fairer alternative to other facial recognition datasets. PPB forms subgroups based on skin phenotypes (shades of colour) instead of ethnic or racial labels. Compared to two other relevant datasets also offering diversity (1) IJB-A, a US government benchmark with Fitzpatrick [155] skin type labels and (2) Adience [47], a gender and age classification benchmark to specifically target gender imbalance. Figure 2.9 illustrates how rating these two datasets according to phenotypes and gender still reveals large disparities, a problem which PPB strives to correct. PPB is, however, still highly biased towards certain age groups and other physiological factors (hair and facial features, for example) and characteristics such as clothing style which are not explicitly balanced. This highlights how collecting a fully unbiased dataset is non-trivial.

The second option of excluding all protected attributes from the dataset is perhaps ideal, but not feasible in most situations. In tasks such as facial recognition, with bias attributes of skin color, hair type and facial features, the core attributes are very difficult to separate from the biased attributes. In other scenarios, even when the attributes are sufficiently isolated and separable, the possible values a bias attribute can take on cannot be comprehensively identified. For example, in medical imaging, the physical presentations in an image are caused by global region/patient origins and backgrounds, equipment, procedures, and many other factors. Knowing beforehand all possible bias sources is thus very difficult. Nonetheless, in certain situations, an ideal dataset for bias unawareness would omit all protected attributes and result in a model not dependent on spurious correlations [144].

### 2.3.2 Augmented and synthetic data

A dataset can also be de-biased via artificial balancing, accomplished by augmenting the existing data to introduce more variants of minority samples, sometimes through synthetically generating data. This creates the illusion of a balanced distribution.

Data augmentation in deep learning has long been proposed to mitigate the overfitting and data shortage problems [141]. These are all applied before training, and typically include transformations which are randomly applied to the training data. Augmentation methods range from simpler geometric transformations such as flips, rotations, and warps, to feature space augmentation and neural network-powered style transfers. Calmon et al. [26] learn an augmentation transformation which specifically targets minimizing discrimination with respect to a protected attribute. A useful regularization technique, augmentation can also be used in parallel with other strategies; one dynamic oversampling bias mitigation method notes that their method is ineffective without augmentation [8].

Data can be synthetically generated from existing data, and added to or used instead of the training set. Geirhos et al. [57] show that ImageNet-trained convolutional neural networks (CNNs) are strongly biased towards texture (but humans are not!), and decrease reliance on this bias by training on a Stylized-ImageNet (SIN), a synthetically modified version of ImageNet where object-related texture information is replaced with randomly selected artistic paintings (Figure 2.10). An AdaIN [56] style transfer via CNN is used to replace the textures in the modified images.

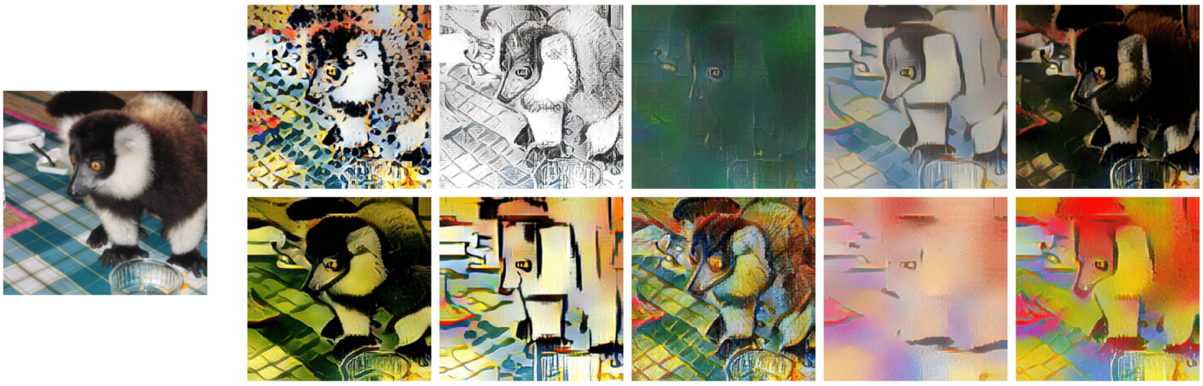


Figure 2.10: Figure from [57] showing one sample image from class “ring-tailed lemur" and ten examples of the synthetic modified versions in Stylized-ImageNet removing the textural bias from CNNs. The styles from paintings are transferred via CNN [56] to the image, replacing the original object texture with a random style.

FairGAN [172] and Fairness GAN [137] use a Generative Adversarial Network (GAN) to either replace or augment the original training data with synthetic data. The generator $G$ generates fake data conditioned on bias attribute $s$ for input $x$ and ground truth label $y$ (Equations 2.1 and 2.2).

$$P_G(s) = P_{data}(s) \tag{2.1}$$

$$P_G(x, y, s) = P_G(x, y \mid s)P_G(s) \tag{2.2}$$

A pair of discriminators are simultaneously trained, one to differentiate between the real and generated data, and the other to ensure no spurious correlation between the inputs, outputs, and bias attribute $s$. Alternately, others propose first creating disentangled representations which allow for synthesizing specific bias-conflicting samples [99], samples which go against the majority correlations. Authors show experimentally that increasing diversity of bias-conflicting samples during training outperforms oversampling minority features. They then distinguish between “intrinsic" (following other literature, we refer to these as *core*) versus bias attributes, train an encoder to embed both separately, then swap the feature vectors among training samples to create bias-conflicting feature combinations.

Synthetic Minority Oversampling Technique (SMOTE) is another balancing method which mimics oversampling by generating synthetic new minority class instances in the vicinity of the existing minority instances [29]. While mostly applicable in the class imbalance problem - whereby one whole class is the minority group - when the bias attribute is known and labelled in the training data, the method can also be extended for addressing imbalance problems in general. SMOTE-based algorithms have also been used in vision tasks [50]. Similarly, methods such as ADASYN [66] augment the training data by adding synthetic samples to it from the minority class, relying on a weighted distribution to determine the degree of classification difficulty for minority class samples (or minority subgroup when known). These weightings determine how much synthetic data is generated to augment each instance.

Where bias variables are identified and known across the data, synthetic data can be generated to target specific correlations. Smith et al. [147] balance the COCO Captions [34] dataset, commonly used for evaluating bias between background context and gender of people in-situ, with respect to gender by adding synthetic variants of the existing data where the gender of the subject is edited. In the medical domain, Generative Adversarial Networks (GANs) have been used to extend a skin lesion dataset with more diverse skin color [116], but the authors also use counterfactual bias insertion to show that GAN-based data generation can amplify biases. In counterfactual bias insertion, bias hypotheses are tested by adding the potential bias to every sample in the dataset in order to evaluate the effect of that potential bias on the performance of the model. Mikolajczyk et al. [116] demonstrate that GAN-based data augmentation must be done carefully in order to avoid bias amplification.

### 2.3.3 Algorithmic mitigation: explicit

Algorithmic mitigation methods which de-bias while simultaneously learning the target task can be categorized as *explicit* or *implicit*. Explicit approaches require knowing the bias variables and their distribution across the training data. In this section, we also include class imbalance mitigation methods, since as long as the labels for the bias variables are known, such methods can be applied in the same way to mitigate bias against underrepresented groups.

**Re-sampling and cost-sensitive learning**

As dataset collection can be expensive and time-consuming in many domains due to privacy and sensitivity constraints, one category of works in the literature have experimented with re-weighting or re-sampling during training, the intuitive approach by which the minority samples are made to appear more often, thus encouraging the model not to leverage undesired correlations. In the explicit setting where bias variables are known, this can be done via two broad methods:

1. By dynamically or statically modifying the sampling probability by either under- or over-sampling; and
2. By cost weighting during training, similar to increasing the learning rate when encountering underrepresented samples [43].

The first modifies the class distribution of the training data, whereas the other imposes non-uniform classification costs; the methods have been shown to be theoretically equivalent under certain conditions [48].

Adjusting sampling probabilities for the majority/minority samples requires pre-identification of “minority" and “majority" groups and corresponding labels for all training samples. Then, the majority class(es) can be under-sampled to create a representative subset, removing the imbalance. The method has the advantage of decreasing the training set size and shortening training time and computational expenses, yet runs the risk of discarding useful information. Strategies on subset selection range from random selection, proximity to class boundary in the class imbalance scenario [78], to more sophisticated methods trying to identify redundant samples to remove [95].

Conversely, the minority samples can be over-sampled either at random or by some selection criteria [78, 169]. Kamiran et al. [83] suggest preferential sampling based on the classification accuracy vs. discrimination trade-off. While effective to a degree, the over-sampling strategy makes the model more prone to overfitting if the exact copies of existing samples are used. Chawla et al. with SMOTE [29] reach the surprising conclusion that both over-sampling the minority via synthetic additions *and* under-sampling the majority group performs better than the over-sampling alone, suggesting potential for combinations of the approaches.

Alternatively, samples can be cost-weighted during training to encourage the model to prioritize loss contributions by minority samples [51, 13]. In addition to the intuitive weighting of minority samples in inverse proportion to the class frequency, various alternatives have been proposed. Cui et al. [40] address the problem of long-tailed class imbalance by computing the *effective* number of samples instead of true count in a subgroup based on group densities. They then cost-weight inversely proportionally to this. Through progressively training on subsamples, Wang et al. [166] modify the loss to transfer knowledge from the majority classes to the minority, and show improvement over the traditional class-inverse loss weighting. Lin et al. [105] tackle the object detection and segmentation task with a focal loss which weights minority samples higher and down-weights majority samples, preventing the model from being overwhelmed by uninformative easier samples. Finally, Huang et al. [75] propose learning a feature embedding to preserve margins between classes, coupled with a weighted loss.

Drummond et al. [44] argue that at least for a decision tree learner, under-sampling has advantages over over-sampling; Weiss et al. [169] note that for larger datasets cost-sensitive learning is preferable to oversampling. While many have compared these methods, the best method depends on the specific use case and numerous factors [184, 29, 78, 169, 83].

### Adversarial de-biasing

Another category of explicit methods focus on discouraging the representation portion of the network from encoding the biases. This can be accomplished through adversarial learning, whereby protected bias feature $b$ is explicitly given to the network along with the input and target output, $(x, y, b)$. A predictor network $P$ is given $(x, y)$ and minimizes loss $L(\hat{y}, y)$. The output of $P$ is passed to an adversary network, whose task it is to predict $b$ given $P(x)$. The update to all the weights is a combination of gradients on both networks, motivating the predictor to keep the adversary from learning the protected feature $b$ while obtaining optimal accuracy [178]. This technique is shown to decrease racial discrimination for predicting recidivism scores compared to the COMPAS [9] score known to be heavily biased towards black inmates [163].

However, as adversarial training requires learning the additional task of biased feature prediction, it can create undesirable confusion for the actual target task, decreasing overall performance [167]. Furthermore, it is at high risk of redundant encoding, whereby another non-identified variable is learned as a proxy instead of the bias variable [46, 65]. Numerous proxies can exist; for example, profession can be a proxy for gender, or insurance type a proxy for socio-economic class.

### Learning through awareness or blindness

Other methods explicitly learn the bias variable, in order to then unlearn it or adjust the model decisions based on this knowledge [167]. Alvi et al. [7] propose “fairness through blindness", simultaneously learning the target task while learning to ignore the bias variables. This requires dividing the training data into general primary data and $N$ secondary datasets, each of which contain instances with a particular bias feature. The primary loss rewards correctly classifying the primary dataset images, while the secondary losses reward *not* learning to classify the secondary dataset images; i.e., by imposing a uniform confusion loss (Figure 2.11). As the losses are opposing, they are optimized in alternating iterations rather than simultaneously.

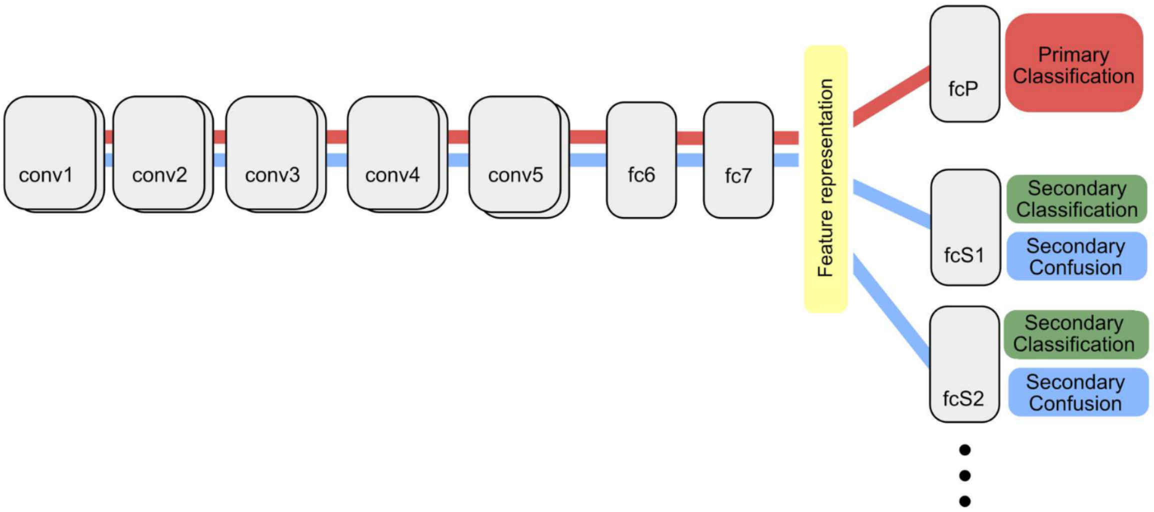


Figure 2.11: Figure from [7] representing the CNN architecture used to jointly learn the feature representation for the primary task ($fcP$) while minimizing a confusion loss which changes feature representations ($fcS_i$) such that they become invariant to the identified bias features.

Other works mitigate bias by manipulating the feature representations. By aggregating class and bias prototypes in feature latent space post-training, [157] remove the bias direction, creating “protected embeddings". Similarly, [154] add a regularizer to entangle the feature vectors of data in the same target class, and disentangle the feature vectors of data of the same class but different bias variable. They do this by learning orthogonal representations for samples of the same bias variable while simultaneously learning similar representations for samples of the same target class. Du et al. [45] show that the features learned from a trained network can be debiased using the classification head fully-connected layers at the end of the network only, using a specific selection of bias-conflicting samples. This allows for a richer learned representation from all the data, yet a fairer model.

“Fairness through awareness" also suggests explicitly learning the bias variable [46]. The most direct approach is to train a classifier with *number of classes* * *number of bias variables* possible outputs. With this setup, during inference the correlation between each bias variable and target class can be removed by considering each softmax output as a probability [133, 132].

### Grouping

The strategy of grouping the training or test data into subgroups is known as Group Distributionally Robust Optimization (Group DRO). Group DRO suggests optimizing such that the model generalizes best over all subgroups during training [134]. They find that adding a L2 penalty to regularize the network results in equivalent worst-group train and test accuracies. In other words, worst-group training error is indicative of test error, so focusing on the former during training results in a fairer model.

Similarly, Arjovsky et al. [10] define this as Invariant Risk Minimization (IRM) whereby a feature representation is learned from the training data such that the corresponding classifier is optimal across all possible environments - environments being subgroups of different combinations of attributes. IRM assumes that training samples do *not* come from the same distribution, but rather from multiple distributions. In bias mitigation terms, the invariance across those distributions are the core features. Subsequent studies find that in a single-bias variable scenario, IRM frameworks are capable of being invariant predictors [36]. Work remains to scale IRM to larger datasets, models, and more complex bias settings.

Predictive group invariance (PGI) [2] also groups data during training and encourages predictive invariance, or similar performance, across pre-selected sub-groups of the training set, assuming access to the majority and minority groups. The network is encouraged to learn group invariant features by learning the class-conditioned feature distributions to have the same softmax distributions on average as training progresses. Minimizing the Kullbeck-Leibler (KL) divergence between the two mean predictive distributions penalizes learning spurious features that do not appear across all groups. While both Group DRO and PGI assume bias variables are specified in the training set, they also propose ways to infer groupings implicitly from the data, which we discuss further below.

Wang et al. [165] propose a bias mitigation method via reinforcement learning whereby deep Q-learning is used to find appropriate margins between subgroups for the network. Majority groups are trained with a fixed margin, while minority groups are trained with adaptive margins, guided by the Q-learning agent. The method is shown to produce more balanced features across identified bias variables.

#### Ensembles

Most explicit ensembling methods employ two networks. The first branch $f_b$ learns to predict the target class $y$ using only the bias variable spuriously correlated with that $y$. The second branch then focuses on the samples which $f_b$ cannot classify correctly (assumed to not have that bias and thus be bias-conflicting, or minority, instances). The two branches can be trained separately [67] or in joint manner [25], and ensembled via learned weights or entropy constraints. Clark et al. [37] train the second model alongside the first in a second stage of training as an ensemble.

Wang et al.[167] modify [46] to train an ensemble of classifiers with shared feature representation. This strategy has reduced complexity compared to having an ensemble of independent networks, yet still allows learning the class-bias boundaries. At test time, the output probabilities of all classifiers are adjusted to essentially average the class decision boundaries across all identified biases.

### 2.3.4 Algorithmic mitigation: implicit

Explicit mitigation methods have the advantage of being able to directly leverage information about the bias variables, whether their density or gradients in latent space, or to formulate various groupings of training or test data to optimize for invariance. While intuitive and often effective, these strategies are at risk of learning other, non-identified bias variables via proxy. In contrast, the more challenging implicit approach does not assume any prior knowledge of bias variables in the data. Therefore, they also have the potential to mitigate more comprehensively.

### Re-sampling and cost-sensitive learning

Implicit re-weighting approaches function the same as explicit re-weighting, but rely on some weighting function which is called either before or during training. One such method by Amini et al. [8] uses a De-Biasing Variational Autoencoder (DB-VAE) to discover the sparse areas of the latent feature space. The densities in that space then adaptively guide the choice of training batches, giving more diverse inputs a higher likelihood of being sampled in a batch (Figure 2.12). A tunable parameter controls the sampling probabilities and how much they rely on the weighting function.

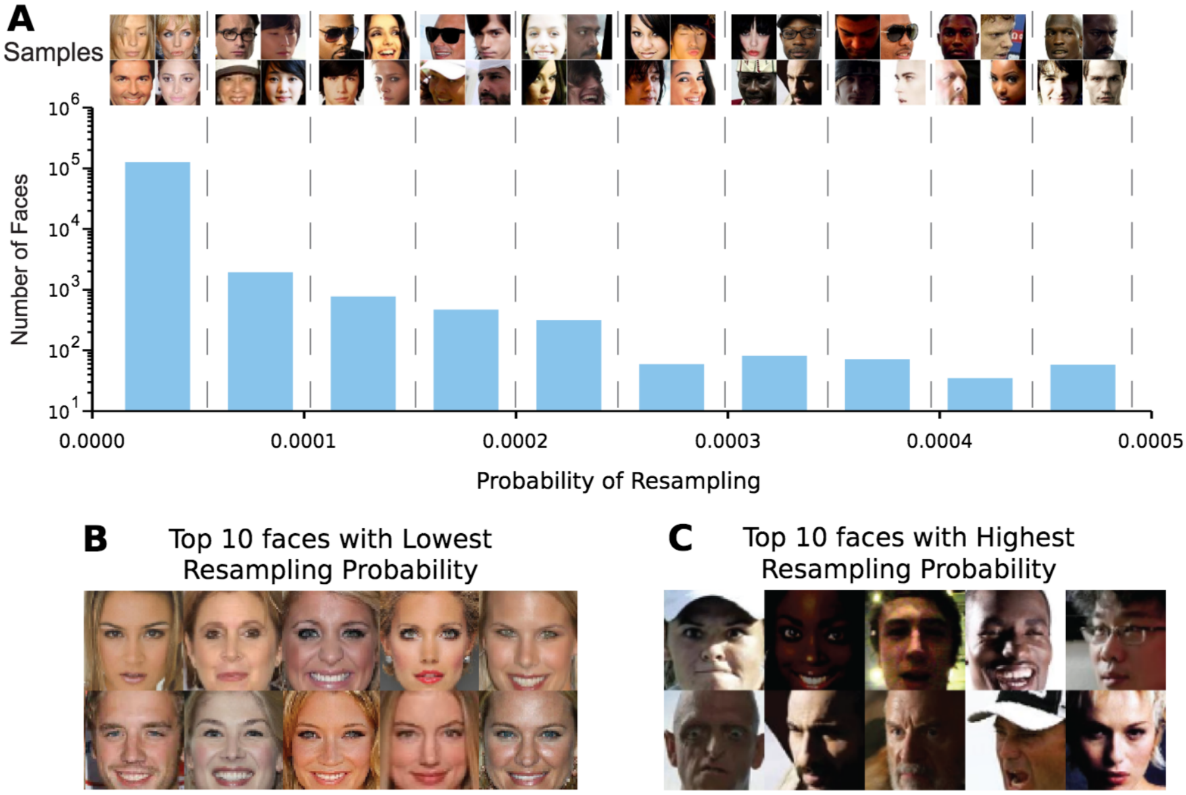


Figure 2.12: Figure from Amini et al. [8] showing higher sampling probability for more diverse instances, learned implicitly through locating the sparser areas of the input feature space. Without having been provided any bias variables, the DB-VAE has learned the underrepresented subgroups with darker skin colors, alternate poses, occlusions, and unclear lighting. In contrast, majority traits such as light skin, blond hair, and female gender have lowest re-sampling probability.

Our novel approach proposed in Chapter 3 and applied to a medical segmentation problem in Chapter 5 also falls under this category of implicit dynamic cost-sensitive methods.

### De-biasing via objective loss

Another category of implicit methods change the loss function to reward learning a fairer model; Xu et al. [173] implement this via a novel loss which penalizes inconsistent false positive rates (FPR). They argue that across different demographic groups in the problem of facial recognition, FPR varies greatly. A fairer model has consistent FPR across all data. The implicit method simply assumes various groups exist but does not require knowing them during training.

Similarly, Pezeshki et al. [126] explore the notion of *gradient starvation*, whereby one gradient is so strong that others' impact is diminished or “starved". Gradient starvation leads to poor generalizability on out-of-distribution and minority data, as well as excessive invariance (over-confidence). They introduce Spectral Decoupling (SD), where the L2 weight decay term in the loss is replaced with an L2 penalty exclusively on the output layer weights. This SD loss promotes balanced learning dynamics across all features, encouraging feature decoupling and penalizing learning features which are learnt at the expense of learning another. SD allows the learning of simple correlations but not unilaterally, resulting in the preservation of minority core features.

### Ensembles

Another category of implicit de-biasing methods leverages the observation that spurious or biased correlations are most easily learnt in the training process as they most quickly lead to a smaller objective loss. Figure 2.13 shows how for two benchmark datasets the model converges more quickly on the bias-aligned data samples, and much slower on the bias-conflicting samples. Those more difficult samples are learned later in the training process. This phenomena can be leveraged for implicit mitigation via ensembles. Learning from Failure (LfF) [119] trains two networks simultaneously; a first which amplifies the early-stage predictions. The samples are dynamically weighted by difficulty, and the most difficult are passed to the second network.

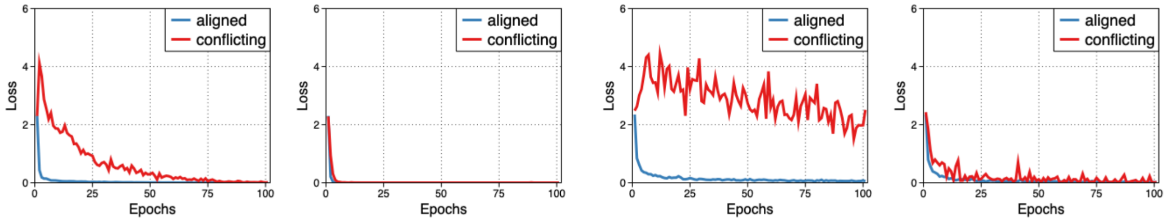


Figure 2.13: Figure from Nam et al. [119] on two different datasets: (*left pair of figures*) Colored MNIST where digits are spuriously correlated with color, and (*right pair of figures*) Corrupted CIFAR-10 where alterations are spuriously correlated with target classes. For both datasets, the left plot shows convergence for the network amplifying the early-stage predictions, while the right shows the learning of the second de-biased network.

Learning with Biased Committee (LWBC) [90] similarly relies on an ensemble or “committee" of networks to learn the bias variables naturally, and a final network which is trained on the instances most difficult for the committee. The networks are all trained simultaneously and the single network passes information back to the committee so that over the course of training, they also are de-biased. Other methods further encourage a first model to learn biases by using a weak model [160] or limited capacity model [136], then train a more robust, higher capacity model on the bias-conflicting samples identified by the weaker model.

Bahng et al. [14] first train an ensemble of models, then train a final model to learn representations orthogonal to the set of representations learned by the ensemble, thus learning a de-biased representation.

### Architectural inductive biases

In the last year, Shrestha et al. [143](Figure 2.14) opened a promising new direction of work, leveraging architectural inductive biases with an adaptable architecture (OccamNets) which allows the network to favor simpler solutions when needed, inline with Occam's razor. They note that neural networks use the same complexity of function for learning all inputs, whereas a less biased model would use the minimum amount of information required for each input, i.e., only the core features.

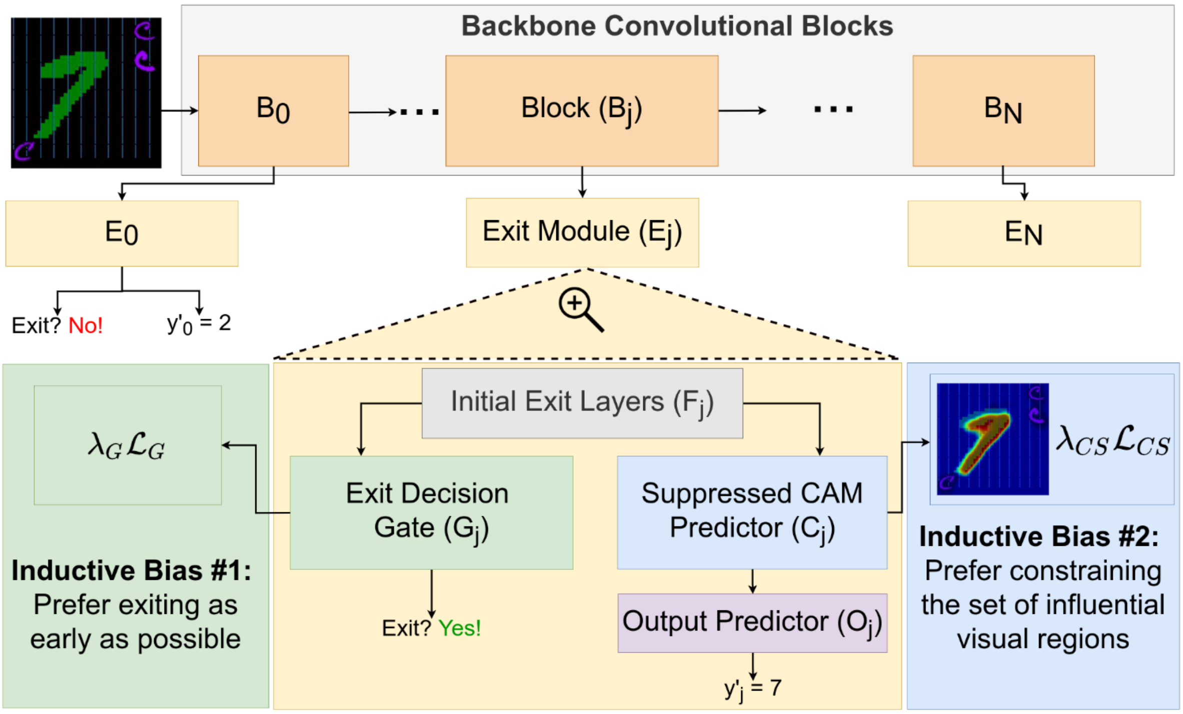

Backbone Convolutional Blocks
$B_0$
Block ($B_j$)
$B_N$
$E_0$
Exit Module ($E_j$)
$E_N$
Exit? No!
$y'_0 = 2$
$\lambda_G \mathcal{L}_G$
Inductive Bias #1: Prefer exiting as early as possible
Initial Exit Layers ($F_j$)
Exit Decision Gate ($G_j$)
Exit? Yes!
Suppressed CAM Predictor ($C_j$)
Output Predictor ($O_j$)
$y'_j = 7$
$\lambda_{CS} \mathcal{L}_{CS}$
Inductive Bias #2: Prefer constraining the set of influential visual regions

Figure 2.14: Figure from OccamNets paper [143] showing the adaptive architecture with early exiting via exit decision gates. During training, there are two inductive biases applied: (1) to prefer exiting as early as possible, and (2) to constrain the size of the influential visual regions.

The proposed OccamNet model adds "early exiting" at each layer, whereby the network is more shallow for an input that can be classified correctly at an early layer of the network. The full depth of the network is reserved for samples that are too difficult to be learned by previous layers. OccamNet prefers focusing on smaller visual regions for predictions yet does not constrain all samples to rely on the same complexity of hypotheses, as optimal “exit" locations in the architecture are learned per sample.

We observe with our experiments comparing this approach to our novel method in Chapter 4 that OccamNets are particularly effective on datasets where core features can be learned quickly, because for the majority of inputs, the model can exit early and thus not rely on the biases. The model can struggle on more visually complex datasets, which we hypothesize is because spatially larger features must be observed in order to make a decision, thus removing the advantage of early exiting. Regardless, the architectural inductive bias line of research opens up a promising, largely unexplored new direction for bias mitigation.

## 2.4 Visual bias challenges

Many challenges still remain, which motivate the need for further contributions in the field. Even with many mitigation methods shown to be successful in experimental scenarios, sample imbalances between minority and majority groups can be too steep to overcome without overfitting [169, 8, 152], learning spurious correlations is often much easier than learning the desired core features [64], and some bias-conflicting samples are simply more difficult to learn than others [165].

Studies of bias in datasets have offered advice on dataset creation, such as selecting datasets automatically rather than curated manually; as much as possible, collecting multi-national, if not global, data; using crowd-sourcing for labelling instead of a small select group of experts (though as discussed in the de-biasing ImageNet case, Amazon Turk workers have also reflected historical biases [174]); and collecting a sufficiently unbiased validation or test set for model de-biasing or evaluation.

Other works, however, show that even if it were always feasible to obtain a completely balanced dataset with respect to identified subgroups, this in itself does not ensure a fair model, as relative quantities are not the sole contributor towards bias [112, 64]. For example, in the context of facial recognition, Wang et. al [165] observe that certain racial subgroups contain more diversity of features than others, so simply balancing the quantities of each racial subgroup in the training set is not enough to produce a fair model.

Furthermore, the concept of fairness, while formally defined in terms of feature subgroups and performance metrics, in practice differs depending on the context and intended outcomes. For example, relying on classification accuracy to choose an optimal model can be misleading for unbalanced classes, and even more for majority/minority subgroups within those classes. Even when relying on a more nuanced fairness metric, model performance under one such metric may do poorly under another, though both are designed to test for fairness. In fact, this can be considered as another type of bias, evaluation bias. The choice of an evaluation metric and benchmark in themselves can result in discrimination [22, 153, 112]. The lack of a unified, comprehensive fairness definition and metric also contribute to the challenge of the bias mitigation problem. The appropriate metric still remains dependent on the task, data, and desired outcome.

Mehrabi et al. [112] also identify a further problem, assuming that via some mitigation method we could guarantee fairness of equality across all possible subgroups: *equity* [61]. Equality refers to giving each subgroup equal outcome, but equity factors in that not all subgroups *need* the same outcome in order to succeed. This opens up a slew of additional considerations and to date has been under-explored.

Finally, literature comparing results for multiple methods on more than one visual bias benchmark dataset demonstrate that novel methods tend to work well on certain but not all datasets [142, 167, 64], highlighting the lack of a comprehensive solution to all forms of bias. Though the research and AI community has made progress towards understanding the problem, identifying causes, and proposing a diverse collection of mitigation methods, the bias mitigation problem is far from solved.

# Chapter 3

# An uncertainty-weighted loss for bias mitigation

## 3.1 Overview

Various methods have been proposed to mitigate visual bias, the majority requiring explicit knowledge of the biases present in the training data. In this chapter, we explore a novel implicit bias mitigation method (we will refer to it as EpiUpWt - for Epistemic Uncertainty Up-Weighting) which dynamically leverages the relationship between predictive uncertainties and bias-conflicting training samples. Firstly, we establish the connection between predictive uncertainties of Bayesian neural networks and bias-conflicting samples in the literature. We then define a simple approach which employs a Bayesian neural network in order to dynamically approximate predictive epistemic uncertainties for samples during training, identifies potential bias in individual training samples, and weights the loss function accordingly. Intuitively, this motivates the model to pay more attention to minority samples. We select two bias benchmark datasets, one a benchmark in the literature and a second generated to complement it, and demonstrate the method's potential to successfully identify bias-conflicting samples, and to mitigate bias.

Finally, we evaluate the approach with a challenging real-world face detection dataset where the test data is significantly more diverse than the training data. Some of the contents of this chapter are in our published work in [152].

## 3.2 Bayesian deep learning

Deep neural networks have in the past decade quickly risen to the forefront of artificial intelligence. The universal approximation theorem states that a feed-forward network with a single hidden layer can approximate any continuous function for inputs within a particular range. In application, this ability to model is evidenced in the state-of-the-art results given by neural networks in most artificial intelligence problems including classification, detection, segmentation and most recently, language, image, and video generation.

However, neural networks are also the subject of criticism. They exhibit “black box" behaviour, lacking human-understandable interpretability, are prone to learning spurious correlations, and perhaps most critically, cannot convey when they are not certain, fostering a public distrust in many critical domains.

The final output of a neural network in the standard classification problem is a normalised prediction vector, often erroneously interpreted as a probability. In reality, this output is not a true probability but rather a softmax normalisation of the final layer output. While the values may correspond to how strongly the model feels about a prediction if the input is within the range of inputs it has seen during training, they become irrelevant and meaningless for an input which is outside that range. Unfortunately, the model itself has no knowledge of which values are within the range of inputs it has seen during training.

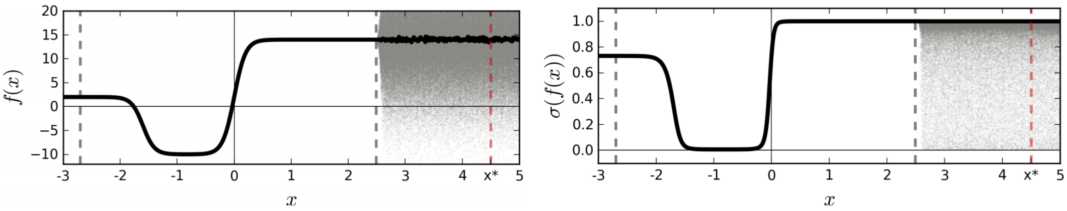


Figure 3.1: As depicted by Gal et al. [53], an arbitrary function $f(x)$ as a function of input data $x$ *(left)* and the softmax, $\sigma(f(x))$ as a function of input $x$ *(right)*.

In Figure 3.1, training data is given between the dashed gray lines, and function uncertainty is shown in the shaded area. Input point x*, far outside the range of training inputs, is classified as class 1 with high probability, completely disregarding the function uncertainty. Hein et al. [68] further show that for such inputs, a neural network is *always* confident.

### 3.2.1 The Bayesian paradigm

Bayesian neural networks are an alternative to the deterministic approach which offer uncertainty estimates through the Bayesian paradigm without sacrificing the learning capabilities of deep neural networks.

The Bayesian paradigm is one of three ways of thinking about probabilities. The *classical* approach views all outcomes as equally likely irrespective of any other information. The *frequentist* approach requires an infinite number of simulations of inputs and outputs, and then makes a decision based on the observed outcomes. Of course, an infinite number of simulations is intractable, so in practice, some finite number is chosen. A larger number clearly is more likely to result in a better decision, but is more costly. Often the difficulty of this approach lies in choosing a suitable sampling number which balances accuracy versus the sampling cost.

The Bayesian approach provides a middle ground between classical and frequentist approaches: the probability of an event expresses the degree of belief held in the event’s likelihood based on prior knowledge. Consider the simple example of a given woman *woman*. What is $p(engineer \mid$

*woman*), the likelihood that she is a engineer? The classicist will predict that both outcomes, that the woman *is* or *is not* an engineer, are equally likely. The frequentist will find $N$ other women, with $N$ as large as feasible, count how many of them are engineers, and then predict using this percentage. A Bayesian approach, in contrast, relies on prior knowledge. Using Bayes' Theorem (Equation 3.1), $p(engineer \mid woman)$ is the probability of $e$ given $w$ and vice versa, and $p(engineer)$, $p(woman)$ are the independent likelihoods of those events.

| **Parameter** | **Definition** |
|---|---|
| $\boldsymbol{\theta}$ | The set of all learnable parameters of a neural network |
| $\boldsymbol{\Phi}_\theta$ | The function learned by a neural network with parameters $\boldsymbol{\theta}$ |
| $\boldsymbol{x}$ | The input to the network |
| $\boldsymbol{y}$ | The target label for the input $\boldsymbol{x}$ |
| $b$ | A scalar bias parameter included in $\boldsymbol{\theta}$ |
| $w_{jk}$ | The $j^{th}$ weight parameter in the $k^{th}$ layer, where J is the total number of outputs from the previous layer |
| $h()$ | An activation function |
| $\mathbb{W}$ | The weights of a single layer included in $\boldsymbol{\theta}$ |
| $D$ | The dataset $(X, Y) = (\boldsymbol{x}_i, \boldsymbol{y}_i), i \in 1..N$ |

Table 3.1: Notation for an artificial neural network.

$$p(engineer \mid woman) = \frac{p(woman \mid engineer)\, p(engineer)}{p(woman)} \tag{3.1}$$

$p(engineer \mid woman)$ represents the posterior, the belief in the woman being an engineer given the observations. The prior $p(engineer)$ reflects our prior knowledge about the likelihood of any person being an engineer. Suppose it is known that 18% of all adults are engineers ($p(engineer) = 0.18$) and that 49% of all adults are women, ($p(woman) = 0.49$). Furthermore, 26% of engineers are women ($p(woman \mid engineer) = 0.26$. This prior knowledge directly affects the likelihood $p(engineer \mid woman) = 0.10$.

### 3.2.2 The Bayesian neural network

#### The foundation: a deterministic neural network

A standard, deterministic artificial neural network (ANN) consists of a set of learnable parameters $\boldsymbol{\theta}$ and is represented by the function $\boldsymbol{\Phi}_\theta$ (Table 3.1 for notation). The fundamental building block of the ANN is a layer, consisting of input $\boldsymbol{x}$, a bias parameter $b$, a non-linear activation function $h()$ and input weights $\mathbb{W}$. For the $k^{th}$ layer with input $\boldsymbol{x}$, and summing over each of the J weighted inputs into the layer:

$$f_k(\boldsymbol{x}) = h(b_k + \sum_{j=1}^{J} w_{jk} x) \tag{3.2}$$

Thus, each output value is a weighted sum plus a bias, passed through an activation function. These layers, stacked together end-to-end, form a neural network. Variations on the computation of $f_k(x)$ and the ways by which the layers are stacked together form different variants of networks, still categorized broadly as ANNs.

Given training data set $D = (X, Y)$ composed of pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i), i \in 1..N$, a cost function computes a loss by comparing $\boldsymbol{\Phi}_\theta(\boldsymbol{x}_i)$ with $\boldsymbol{y}_i$. This loss is used to compute gradients, back-propagated layer by layer through the ANN. $\boldsymbol{\theta}$ is updated via gradient descent, such that the loss is minimized. Training involves iterating through $D$ until some stopping point. $D$ is usually processed in mini-batches for efficiency, better convergence, and regularization. The final $\boldsymbol{\theta}^*$ is the maximum likelihood estimation (MLE) or when regularised, the maximum a posteriori estimation (MAP). $\boldsymbol{\theta}^*$ is a *fixed point estimate* of the optimal $\boldsymbol{\theta}$ given $D$.

### A Bayesian neural network

A Bayesian neural network (BNN) is a stochastic ANN trained using Bayesian inference, whereby probability distributions are placed over parameters in the network and given prior distributions which are then updated with data during inference (Table 3.2 for notation). The stochasticity allows for approximating the posterior distribution $p(\boldsymbol{\theta} \mid D)$. Stochasticity can be introduced into the network by (1) placing probability distributions over $\boldsymbol{\theta}$ directly, or by (2) placing distributions over the neuron activations (Figure 3.2). By expressing the parameters as distributions, $\boldsymbol{\theta}$ is sampled from a high dimensional distribution $p(\boldsymbol{\theta})$, which can be used to compute predictive distributions with a mean and variance.

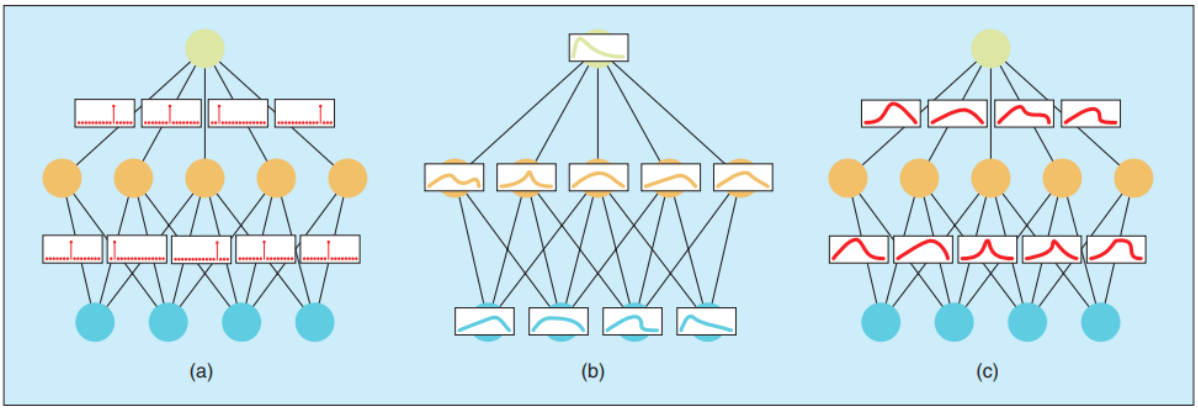


Figure 3.2: Figure from [81] showing (a) a deterministic point estimate neural network, (b) stochastic ANN with probability distributions over activations, and (c) stochastic NN with distributions over all weights.

Regardless of the choice of stochastic model, a prior distribution $p(\boldsymbol{\theta})$ must be chosen over each parameter - typically a simple distribution such as a Bernoulli or Gaussian. According to the Bayesian paradigm, the probability of the posterior $p(\boldsymbol{\theta} \mid D)$ depends on a prior belief about the current hypothesis $\boldsymbol{\theta}$, and the seen data $D$ which is used to update the belief in $\theta$. Bayes' theorem formalizes this relationship as follows in Equation 3.3:

$$p(\boldsymbol{\theta} \mid D) = \frac{p(D \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int_{\theta} p(D \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \propto p(D \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{3.3}$$

As computing the evidence $\int_{\theta} p(D \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is intractable for a neural network, the posterior cannot be directly computed. Instead, approximation methods are used, which can be summarized by two families of methods: Markov Chain Monte Carlo (MCMC) and variational inference (VI).

| **Parameter** | **Definition** |
|---|---|
| $p(\boldsymbol{\theta} \mid D)$ | The posterior distribution |
| $p(\boldsymbol{\theta})$ | The prior distribution |
| $M$ | The Monte Carlo sampling size from the posterior for the posterior estimate of a Bayesian neural network |
| $\boldsymbol{\Theta}$ | The set of posterior samples comprising the posterior estimate $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ...\boldsymbol{\theta}_M\}$ |
| $\boldsymbol{\mu}_i$ | The predictive mean associated with the $i^{th}$ input |
| $\hat{\boldsymbol{y}}_i$ | The class prediction associated with the $i^{th}$ input |
| $\boldsymbol{\sigma}_i$ | The predictive uncertainty associated with the $i^{th}$ input |

Table 3.2: Notation for a Bayesian neural network.

After inference, the final posterior estimate of the neural network is obtained by Monte Carlo sampling from $\boldsymbol{\Theta} \approx \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ...\boldsymbol{\theta}_M\}$, where each $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid D)$. A sampling size $M$ can be determined using a validation set or some other means. From this, the final BNN prediction for input $\boldsymbol{x}_i$, $\hat{\boldsymbol{y}}_i$, is derived from the predictive mean $\boldsymbol{\mu}_i$, or averaged prediction (Equation 3.4). $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the maximum a posteriori (MAP) estimation. For a classification task, the final prediction is the class index from among all classes C with maximum likelihood.

$$\boldsymbol{\mu}_i \approx \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i) \tag{3.4}$$

$$\hat{\boldsymbol{y}}_i = \underset{C}{\operatorname{argmax}} \left(\boldsymbol{\mu}_i\right) \tag{3.5}$$

This can be interpreted as an equally-weighted average from an ensemble, or the mean of the predictive distribution over outputs for a given input. In addition to the benefits of having this regularized model, the Bayesian paradigm also provides uncertainty estimates (further discussion of uncertainties and approximation methods in Section 3.3). We can approximate the model uncertainty associated with the prediction $\hat{\boldsymbol{y}}_i$ for input $\boldsymbol{x}_{\boldsymbol{i}}$, $\boldsymbol{\sigma}_i$ (Equation 3.6) as the predictive standard deviation. Many choose to use Bessel's correction to the standard deviation and compute the variance using $\frac{1}{M-1}$ to correct the bias in the estimation of the population variance. In application, we also approximate the predictive uncertainty as the standard deviation of the predictive distribution (Equation 3.6) throughout this research. Here, as in throughout this research, a $\mathbb{R} \to \mathbb{R}$ vector operation is to be interpreted as the function applied element-wise.

$$\boldsymbol{\sigma}_i \approx \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\boldsymbol{\Phi}_{\boldsymbol{\theta}_m}(\boldsymbol{x}_i) - \boldsymbol{\mu}_i\right)^2} \tag{3.6}$$

### 3.2.3 Inference methods

**Markov Chain Monte Carlo (MCMC)**

MCMC is a method for approximating a multi-variate probability distribution. Monte Carlo by itself involves taking independent random samples from the probability distribution and then using these samples to approximate the true distribution by computing the mean or variance of the samples. However, in a multi-variate sampling space, Monte Carlo is not effective nor computationally feasible due to the high dimensionality of the sample space. Furthermore, the assumption that each sample is independent and can be independently drawn is not true for neural networks.

A Markov chain is a way of generating dependent samples; each sample $i+1$ is probabilistically dependent on the previous sample $i$. While the first sample may be drawn from the prior, the consecutive samples eventually get closer to the desired posterior. The chain of samples eventually arrives at a stationary distribution, at which point the chain has converged.

MCMC algorithms attempt to construct chains efficiently through Monte Carlo sampling. While other sampling methods use rejection sampling - proposing possible next samples and then rejecting or accepting based on some criteria - MCMC algorithms require a “burn-in" stage during which the chain has not yet converged. Once at a stationary distribution, $M$ samples are taken as the final posterior estimate. Two MCMC algorithms in particular are suitable for Bayesian deep learning [1]: the Metropolis-Hastings algorithm [35] and the Hamiltonian Monte Carlo (HMC) algorithm [120].

The Metropolis-Hastings algorithm begins with an initial sample $\boldsymbol{\theta_0}$, and the choice of a proposal distribution $Q(\boldsymbol{\theta'} \mid \boldsymbol{\theta})$ which defines the probability of proposing new sample $\boldsymbol{\theta'}$ given the previous sample $\boldsymbol{\theta}$. The new sample is accepted with a certain acceptance probability $k$ (fixed or drawn from a distribution - typically Bernoulli or Normal) if it matches the desired target distribution better than the previous sample (i.e., if Equation 3.7 holds true).

$$\log\left(k\frac{Q(\boldsymbol{\theta'} \mid \boldsymbol{\theta})p_0(\boldsymbol{\theta})}{Q(\boldsymbol{\theta} \mid \boldsymbol{\theta'})p_0(\boldsymbol{\theta'})}\right) < \sum_{i=1}^{N}\log\frac{p(\boldsymbol{x_i} \mid \boldsymbol{\theta'})}{p(\boldsymbol{x_i} \mid \boldsymbol{\theta})} \tag{3.7}$$

Choosing $Q$ can be difficult, as too wide a distribution will result in high rejection rates, and

---

[1] Gibbs sampling, while highly effective in other settings, does not scale well to high-dimensions and suffers from a long convergence time.

too narrow in highly correlated $\boldsymbol{\theta}$ and $\boldsymbol{\theta'}$ samples. The Hamiltonian Monte Carlo algorithm, another variant of Metropolis-Hastings, addresses these issues by using principles of Hamiltonian dynamics to propose good samples. The Hamiltonian function provides a sampling mechanism defined by potential and kinetic energy, and parameterized by momentum variables which are updated with each sample. HMC methods can explore the distribution space more quickly and with lower rejection rates.

### Variational inference

An alternative inference method from MCMC is variational inference (VI). The VI premise considers that while it is not tractable to compute the data likelihood, one can create a second distribution, known as the variational distribution $q_{\boldsymbol{\theta}}$, and then minimize the Kullback-Leibler (KL) distance between $q_{\boldsymbol{\theta}}$ and the true posterior $p(\boldsymbol{\theta} \mid D)$. This turns the inference problem into an optimisation problem.

Kullback-Leibler divergence, or relative entropy, has its origins in information theory and measures the difference in entropy between two probability distributions. In information theory, entropy represents the amount of information present in given data. KL divergence between two distributions $p$ and $q$ over a set of parameters $\theta$ is defined in Equation 3.8:

$$KL(q \mid\mid p) = \sum_{i}^{N} q(\boldsymbol{\theta_i}) \cdot \frac{\log q(\boldsymbol{\theta})}{\log p(\boldsymbol{\theta})} \tag{3.8}$$

Optimal parameters $\boldsymbol{\theta}^{opt}$ minimize the divergence between the true posterior and our approximation $q(\boldsymbol{\theta})$ (Equation 3.9):

$$\boldsymbol{\theta}^{opt} = \arg\min_{\theta} \mathrm{KL}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta} \mid D)] = \int q(\boldsymbol{\theta}) \cdot \frac{\log q(\boldsymbol{\theta})}{\log p(\boldsymbol{\theta} \mid D)}\, d\theta \tag{3.9}$$

Unfortunately, this contains the true posterior $p(\boldsymbol{\theta} \mid D)$ which cannot be computed. However, by restricting the variational distribution $q(\boldsymbol{\theta})$ to some form of a normal distribution, minimising KL divergence is equivalent to maximising the Estimate Lower Bound (ELBO), the estimated lower bound on $\log p(D)$. In literature this is denoted as the function ELBO. The transformed optimisation problem is shown in Equation 3.10:

$$\boldsymbol{\theta}^{opt} = \arg\min_{\theta} \mathrm{KL}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta} \mid D)] \sim \arg\max_{\theta} \mathrm{ELBO}(\boldsymbol{\theta}) = \arg\max_{\theta} \int q(\boldsymbol{\theta}) \log \frac{p(D, \boldsymbol{\theta})}{q(\boldsymbol{\theta})}\, d\theta \tag{3.10}$$

The variational distribution $q(\boldsymbol{\theta})$ can be modeled via mean field or parametric approximation.

Mean field approximation expresses $q(\boldsymbol{\theta})$ as a factorized family of variational distributions, and is more costly. Parametric approximation restricts each parameter in $\boldsymbol{\theta}$ to a parametric family of distributions. One concern is that a simple parametric family may be unable to adequately model $q$, whereas a complex family may easily overfit the data. The Gaussian family of distributions is most commonly used.

To get a differentiable estimate of ELBO which allows for gradient descent, the reparameterisation trick for variational autoencoders [91] is applied. The trick moves the stochasticity of a node out into a random variable $\epsilon$, making it possible to take the derivative of the function $\boldsymbol{\theta}_i$ with respect to the mean and variance of each distribution. This allows for the standard backpropagation for training the Bayesian neural network.

### Stochastic gradients

While Bayesian approximation via dropout [53] has made Bayesian deep neural networks applicable to many domains due to ease of use and scalability, Markov chain Monte Carlo (MCMC) algorithms [19] are widely considered the gold standard for Bayesian inference. However, both MCMC and VI methods are computationally intractable for large vision datasets or high-dimensional data frequently encountered in real-world computer vision applications, requiring computing gradients over the entire dataset per iteration. Noisy estimates for gradients based on mini-batches, however, is a more scalable option. Such stochastic gradient methods exist for both sampling (MCMC) and optimization (VI) inference methods.

Stochastic gradients for sampling-based methods include stochastic gradient Langevin dynamics (SGLD) [170] and stochastic gradient Hamiltonian Monte Carlo [33]. SGLD is based on diffusion processes such as the Langevin diffusion, a discrete-time approximation of a continuous-time process, formulated as stochastic differential equations (SDEs) to describe the time evolution of a moving object subject to both random and non-random forces. These approaches compute the likelihood over a mini-batch of data, then add an additional noise term which acts as an upper bound on the error of the approximation.

Given model parameters $\boldsymbol{\theta}$, dataset $D$, prior $p(\boldsymbol{\theta})$, and potential energy $U(\boldsymbol{\theta})$, the posterior distribution is $p(\boldsymbol{\theta} \mid D) \propto exp(-U(\boldsymbol{\theta})) = -\log p(D \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$. As computing $U(\boldsymbol{\theta})$ is not feasible for all $D$, SG-MCMC methods approximate $U(\boldsymbol{\theta})$ via mini-batch learning. The gradient of the log-posterior density is substituted by the stochastic gradient over the minibatch and an additive Gaussian noise term that acts as an upper bound on the error.

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \alpha_i \Delta \tilde{U}(\boldsymbol{\theta}_i) + \sqrt{2\alpha_i \epsilon_i} \tag{3.11}$$

The update to parameters is shown in Equation 3.11, at iteration $i$ of the algorithm, for normal distribution $\epsilon_i$, stepsize $\alpha_i$ and minibatch approximation of the potential energy $\tilde{U}$.

While convergence is in practice slower than for other MCMC algorithms, the parameter update process closely resembles stochastic gradient descent and is generalisable to any neural network.

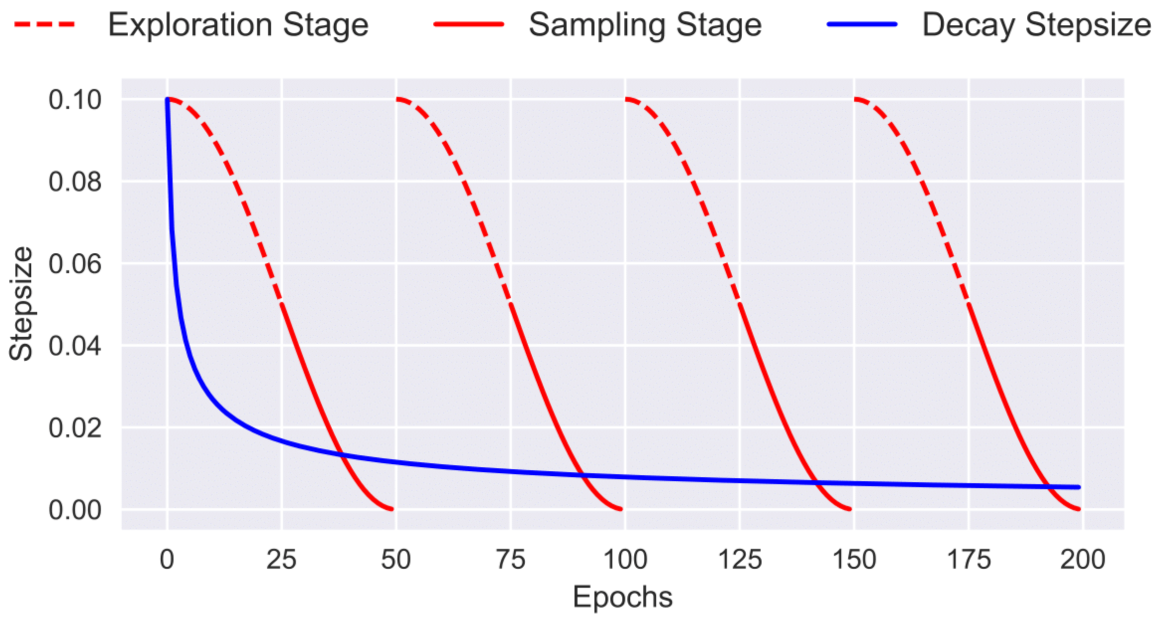


Figure 3.3: Figure from [179] showing the cyclical stepsize schedule (red) compared against the traditional decreasing stepsize schedule (blue) for SG-MCMC algorithms.

To speed up convergence and better explore complex multimodal distributions common for deep neural networks, Zhang et al. [179] propose cyclical SG-MCMC (cSG-MCMC), where a cyclical stepsize schedule allows for quicker discovery of new modes. For each cycle of the learning rate schedule, an initial larger step size allows for exploration, and the subsequent smaller step sizes allow for sampling.

Optimization-based approaches include stochastic variational inference [72] and stochastic approximation for optimization [121]. Such methods adopt an inference procedure very similar to that of standard stochastic optimization (Algorithm 1):

**Algorithm 1** Stochastic variational inference for learning on very large datasets.

```
Require: Training data $D$, Bayesian neural network $\mathbf{\Phi}$, posterior $p(\boldsymbol{\theta} \mid D)$, current variational parameters $\boldsymbol{\theta}_i$
for each iteration $i$ do
    Select batch of one or more samples $\boldsymbol{x}_b \in D$
    Analyze $p(\boldsymbol{x}_b \mid \boldsymbol{\theta_i})$
    Update $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{i+1}$
end for
```

### 3.2.4 Challenges

BNNs, while attractive due to their uncertainty estimates and built-in regularization, pose challenges which still deter general usage as replacements for their deterministic counterparts. They can be sensitive to the choice of prior [145], do not perform optimally using the true Bayesian posterior [171, 77], do not necessarily provide as good uncertainty estimates as desired [70], and irrespective of choice of inference method, have greater computational complexity and longer training times as compared to deterministic networks.

Various works look at combating these issues. In application, most use cold or tempered posteriors [171] for better performance. To deal with the computational complexity, simpler Bayesian formulations, such as BNNs through dropout turned on at test time [53] make training and inference more accessible to practitioners. Others further reduce the training complexity by using Bayesian Last Layer Networks [93] or a trained feature representation network and a small BNN of one or more linear layers only [123, 20, 58] and show evidence that a cheaper Bayesian neural network with only a few probabilistic last layers still results in comparable uncertainty estimates and regularization.

Others propose methods of exploring the multi-modal posterior landscape for better and faster convergence, via adaptive and cyclical sampling step sizes [55, 179]. These allow for inference on large datasets such as ImageNet where training was previously too expensive.

## 3.3 Uncertainties

Uncertainties related to model risk assessment are broadly defined as aleatoric and epistemic. Aleatoric uncertainty, or data uncertainty, measures the level of inherent noise in the data and is dependent on the input data. Epistemic uncertainty is the model uncertainty, and depends on the distribution over model parameters. The two are combined in predictive uncertainty, which depends on the input to the model, the model itself, and the data the model has seen.

### 3.3.1 Aleatoric

Aleatoric uncertainty refers to the data-dependent uncertainty. Aleatoric uncertainty captures the noise in the data, which can be subcategorized into homoscedastic and heteroscedastic uncertainties [76]. The homoscedastic uncertainty is constant irrespective of inputs, whereas the heteroscedastic uncertainty may vary with inputs. Kendall et al. [86] demonstrate the usefulness of heteroscedastic aleatoric uncertainty combined with epistemic uncertainty in depth regression, where object boundaries and objects far from the camera have higher uncertainties. They demonstrate how input-dependent uncertainties can be used to automatically identify more challenging samples.

### 3.3.2 Epistemic

In contrast, epistemic uncertainty refers to model uncertainty, the uncertainty about the model hypothesis and how well it fits the data. This uncertainty corresponds to how likely it is that the given model generated the training data, and reflects the model's uncertainty about how good the model itself is. While aleatoric uncertainty cannot be reduced by increasing the amount of training data, the epistemic uncertainty is directly associated with inputs. The epistemic uncertainties for specific samples *can* be reduced given more data, as additional data can explain away the model uncertainty in the hypothesis.

For a specific input sample, the uncertainty in the model parameters induces an uncertainty in prediction which corresponds to that sample. This is the sample epistemic uncertainty, which is highly useful in application.

In the depth regression setting, epistemic uncertainty is high for pixels which are semantically and visually challenging in a scene [86]; for example, a pavement made of concrete tiles of different colours. Unless such pavements appear frequently in the training data, the model cannot easily interpret their semantic meaning. In the case of the first self-driving car fatality, the extremely rare circumstances of a white truck, the sun straight ahead creating extreme overexposure for the visual sensors, and the position of the nearby cars, led the automated system to believe there was no truck ahead of it [108, 162]. As this area of input space was very sparse, an uncertainty-aware model would have provided high predictive epistemic uncertainty for such a scenario, which could trigger an alert.

### 3.3.3 Uncertainty disentanglement

For certain applications, disentangling the two uncertainty components is useful. Kendall and Gal [86] propose a disentangling model relying on the MC-Dropout Bayesian neural network formulation. The epistemic uncertainty for a particular sample is estimated using Monte Carlo samplings of the posterior and the variance of the predictive distribution, approximated by the sum of the noise of the predictive variance plus a term measuring parameter uncertainty. The heteroscedastic uncertainty is computed by corrupting the model outputs (logits) with a noise parameter which is learned during training. This noise value is dependent on inputs and not the model parameters.

Valdenegro et al. [161] generalize a disentangling method applicable to other Bayesian neural network formulations not only MC-Dropout, by computing the sample entropies of the softmax outputs from posterior samples - the variance of the means for epistemic uncertainty, and a function of the input variances for aleatoric. They find that aleatoric and epistemic uncertainties are entangled irrespective of uncertainty quantification method, and despite aleatoric uncertainties in theory being model-independent. Depeweg et al. [42] compute the total uncertainty for each sample and then subtract the aleatoric component to obtain the epistemic component.

Predictive uncertainty refers to the sample uncertainty comprising both the epistemic and aleatoric components.

## 3.4 Uncertainties and bias-conflicting samples

The predictive uncertainty of a Bayesian neural network is naturally higher in sparser training data regions, which has been noted in early experiments for active learning [109]. Active learning is the learning procedure by which an algorithm chooses which samples from the available dataset to label. This is motivated by the idea that models can achieve better performance with a smaller – but more intelligently selected – labelled subset of data, and by the fact that in many applications, labelled data is expensive and difficult to obtain. Some querying function must be chosen which determines how to choose which data to label, and when.

Uncertainty sampling is a commonly used query framework [101] in active learning, supported by the simple reasoning that choosing samples which maximize information gain results in a better-informed model. While the framework can work in a non-probabilistic setting, for example in support vector machines by choosing samples closest to the decision boundary [38], or by choosing samples with highest entropy [73], it is naturally suited for probabilistic uncertainty-aware models. In a classification setting, the querying function returns the least informative sample for all possible labels. Alternatively, it can return the sample with the highest predictive uncertainty (See Equation 3.12 for the most likely class label $\boldsymbol{y}^*$).

$$\boldsymbol{x}^* = \mathrm{argmin}_x \; p(\boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{\theta}) \tag{3.12}$$

Bayesian Active Learning by Disagreement (BALD) [74] seeks to choose the sample for which the various posterior estimates disagree the most; in other words, the sample for which the variance of the estimated posterior predictive distribution is the largest. Gal et al. [54] compare the performances of BALD, entropy maximization acquisition functions, and others, on the MNIST logit classification image dataset. They note that entropy-based methods which capture and leverage aleatoric uncertainty are not as effective as a Bayesian neural network with an uncertainty-based acquisition function which actively minimizes the epistemic uncertainty (greedily chooses to label samples with maximum epistemic uncertainty) throughout training.

Samples whose combination of attributes are not aligned with the majority biases are called bias-conflicting samples. Irrespective of source, a bias-conflicting sample belongs to one of the following scenarios:

1. **Minority attribute bias**. When a subgroup of the data has a particular attribute or combination of attributes which are relatively uncommon compared to the rest of the dataset, they form a minority group. A model is less likely to correctly predict for samples from a minority group than for those of the majority. These minorities, when exclusively correlated with a target label, are equivalent to class imbalances - having fewer of certain class examples than others. However, in most cases they are not directly correlated with target classes.
2. **Sensitive attribute bias**. A sensitive attribute (also referred to as “protected") is one which should not be used by the model to perform the target task, but which provides an unwanted “shortcut" which is easily learned, and results in an unfair model. These attributes are not exclusive to one target label but are usually correlated to one or more labels.

By definition, bias-conflicting samples are in sparser areas of the input space, and thus have higher epistemic uncertainties [109]. Branchaud et al. [18] explore experimentally whether BALD can mitigate fairness issues for active learning. In particular, they test whether uncertainty-aware models with a BALD acquisition function can improve model fairness with minority group and sensitive attribute biases. BALD increases the accuracy and predictive parity of the bias-conflicting groups while reducing the group epistemic uncertainty compared to the bias-aligned groups.

## 3.5 Uncertainty-weighted loss

### 3.5.1 Loss-weighting in literature

Cross-entropy (CE) loss is the most common choice for multi-class classification loss for deep neural networks. In standard form, cross entropy loss computes the sum of squared errors, treating each sample as equally important in the loss calculation. It is widely known that this leads to skewed performance in the presence of class imbalance, so a simple tactic is to weight each class inversely proportional to the class frequency [96]. Alternately, more difficult instances can be down-weighted [105] or up-weighted [51] via a weighting parameter correlated to the estimated class probability. Aurelio et al. [13] incorporate prior class probabilities into a cost-sensitive CE loss; and Ren et al. [130] assign sample-level weights based on their gradient direction compared to a “clean" unbiased validation set.

Homoscedastic uncertainties, not dependent on the input but rather on task, from MC-Dropout formulated BNNs are used in a multi-task learning setting to weight losses for scene geometry and semantics [87]. In multi-task learning, the losses associated with multiple target tasks are combined with some weighting. Kendall et al. propose a principled method of determining the optimal loss weightings using homoscedastic uncertainty, and show that it can improve performance.

### 3.5.2 Method

Given the relationship between sample predictive uncertainty and bias (Section 3.4), we propose a simple dynamic sample-level cost function which we call Epistemic Uncertainty Up-Weighted cross-entropy loss (EpiUpWt). We use the variance of the predictive distribution to estimate sample uncertainty, and while we do not isolate the epistemic uncertainty from aleatoric in our model, though various disentangling methods have been proposed [86, 161] all contributors to the uncertainty which vary depending on input (epistemic and heteroscedastic aleatoric) are of interest to us, particularly the epistemic component. The novel approach uses dynamic uncertainty estimates during training to weight the objective loss for bias mitigation against minority samples.

Equation 3.13 shows the weighted loss used in Equation 3.14 to compute the loss for training sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, weighted cross entropy loss $L_{CE}$ and the sample-wise weighting function with tunable parameter $\kappa$ controlling the degree of weighting, especially for high-uncertainty samples. $\sigma_{i,y_i}$ indicates that the uncertainty value corresponding to the ground truth prediction $\boldsymbol{y}_i$ is used.

$$w_i = (1.0 + \sigma_{i,y_i})^{\kappa} \tag{3.13}$$

$$L_{CE} = L(\boldsymbol{y}_i, \boldsymbol{\Phi}(\boldsymbol{x}_i), w_i) = -w_i \sum_{C} \boldsymbol{y}_i \cdot \log(\boldsymbol{\Phi}(\boldsymbol{x}_i)) \tag{3.14}$$

So that normally weighted samples have weight 1.0, we shift the distribution such that lowest uncertainty samples are never irrelevant to the loss term. We compute $\hat{L}$ sample-wise and then reduce over the mini-batch. $\kappa = 1$ is equivalent to a normal weighting, whereas $\kappa \to \infty$ increases the importance of high-uncertainty samples. In our fully bias-unaware setup, $\kappa$ is selected using validation loss tuned via grid search. The weighting value $w_i$ used to scale the loss is shown in Figure 3.4 as a function of $\kappa$.

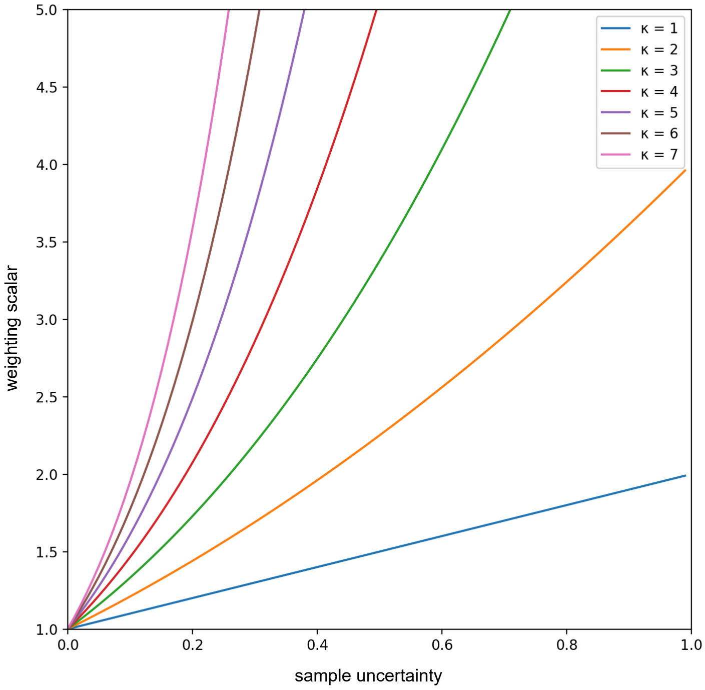


Figure 3.4: The weighting scalar of $L(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is a function of sample uncertainty $\boldsymbol{\sigma}_{i,y_i}$ and scalar $\kappa$. Horizontal line $y = 1$ along the x-axis represents $\kappa = 0$, or no de-biasing. A larger $\kappa$ means that smaller discrepancies in uncertainty have a larger impact on sample weighting.

The sample predictive uncertainties in Algorithm 2 are estimated using the posterior samples from each cycle of the MCMC sampling. If $M = 15$ for example, and there are 3 sampling cycles, the earliest moment at which we can compute $\boldsymbol{\sigma}$ over the training data is after the sampling phase of the first cycle, after the first five samples have been taken (the posterior estimate is $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4\}$, and $M = 5$). Uncertainty values are updated at each consecutive cycle to reflect the developing posterior, requiring a total of $M(C-1)$ samples from the posterior for total number of cycles $C$.

In implementation, this can be sped up during training by storing each training sample's predictive distribution for each $\boldsymbol{\theta}_m$ (Algorithm 2 line 6). Once the sample-wise model uncertainties have been computed from the mean predictions (line 11), all predictions can be discarded. We

**Algorithm 2** Training loop using uncertainty-weighted loss

```
Require: Training data X, Y, neural network $\mathbf{\Phi}$, weighting parameter $\kappa$, and stepsize $\epsilon$
Initialize parameters $\boldsymbol{\theta}$
for each cycle, $c$ do
    for each epoch $e$ in $c$, $\forall \boldsymbol{x}_i \in X$ do
        if $e$ in sampling phase then                                        ▷ sampling phase
            $\boldsymbol{\theta}_m \sim P(\boldsymbol{\theta} \mid D)$                 ▷ take a posterior sample
            $\mathbf{\Phi}_{\theta_m}(\boldsymbol{x}_i)$      ▷ save all predictions using that posterior sample
        end if
        $L_{CE} = L(\boldsymbol{y}_i, \mathbf{\Phi}_\theta(\boldsymbol{x}_i))$                 ▷ normal loss
        if $c > 1$ then                                  ▷ all cycles after the first cycle
            $\boldsymbol{\mu}_i = \frac{1}{M} \sum_{m=1}^{M} \mathbf{\Phi}_{\theta_m}(\boldsymbol{x}_i)$
            $\boldsymbol{\sigma}_i = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\mathbf{\Phi}_{\boldsymbol{\theta}_m}(\boldsymbol{x}_i) - \boldsymbol{\mu}_i\right)^2}$     ▷ compute sample uncertainties
            $w_i = (1.0 + \sigma_{i,y_i})^\kappa$
            $L_{CE} = L(\boldsymbol{y}_i, \mathbf{\Phi}(\boldsymbol{x}_i), w_i)$                 ▷ weighted loss
        end if
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla L_{CE}$                 ▷ update to parameters
    end for
end for
```

set the length of the sampling phase to be 5 epochs, the shortest length for which uncertainty estimates are consistently stable for a fixed seed.

## 3.6 Experiments

### 3.6.1 Datasets

Most, if not all visual datasets contain biases, but suitable visual bias benchmarking datasets must (1) have known, realistic, controllable bias(es), (2) not contain other challenges that could distract from the challenge of overcoming the biases, and preferably (3) be of manageable size. MNIST [98], the popular image dataset for digit classification, broadly meets these criteria and thus has been modified by researchers for benchmarking purposes. These variants include Biased MNIST [14] with a background color bias highly correlated with digits, MNIST with Colored Squares on the Corners [12] correlating the added square feature with digits, Colored MNIST [103, 89] with high correlation between each digit's color and its label, C-MNIST [10] and Extended C-MNIST [92]. Shrestha et al. add complexity to the dataset by placing each MNIST digit in a 3 x 3 grid with seven bias variables and 10 possible values each variable can take [142]. An updated version of this Biased MNIST enlarges the grid to 5 x 5 [143]. In most cases, the degree of bias $p_{bias}$ can be controlled. Furthermore, some frame the problem as binary classification [12] while others as multi-class 10-digit classification [14, 142, 103, 89].

Others turn to synthetic datasets such as Synbols [97] with injected biases [18] or the object classification CIFAR-10 or CIFAR-100 datasets [94] for benchmarking. CIFAR-10 consists of 60k 32x32 color images equally labelled as one of ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Wang et al. [167] propose CIFAR-10S or CIFAR-10 Skewed which is carefully constructed to be contain sensitive attribute bias across 10 output classes. We choose this dataset for our initial EpiUpWt benchmark for its non-trivial complexity, existing benchmarks, and strictly controllable synthetic bias variable. We contribute our own complement to it, CIFAR-10M or CIFAR-10 Minority.

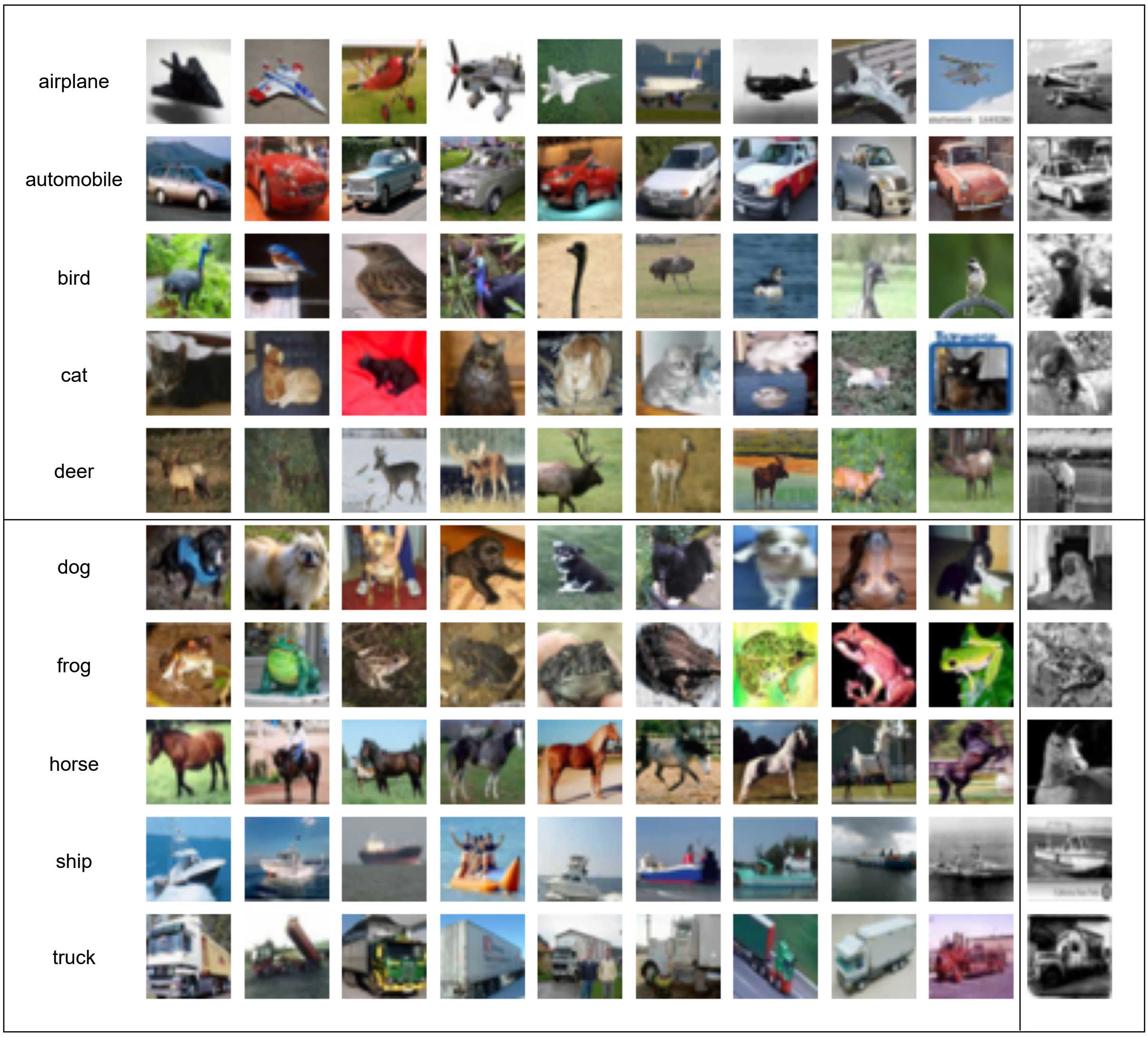


Figure 3.5: A sampling of 100 images, 10 from each class from our minority class bias benchmark CIFAR-10M with $p = 0.9$ as an example of minority group bias equally skewed across all classes. Note that our experiments use $p = 0.95$.

Our new dataset CIFAR-10M corresponds to minority group bias, whereas CIFAR-10S [167] includes a sensitive attribute bias. For both, we use the official CIFAR-10 test set, duplicated for one COLOR copy and one GRAYSCALE copy, and a 5:1 training-validation split for the total of 50k training images and 10k test images.

### Visual bias benchmark: CIFAR-10 Minority

CIFAR-10M (Figure 3.5) is formulated to capture minority group bias, with a single bias variable with two values: color and gray-scale. For each of the ten target classes, 5% of samples are converted to gray-scale. The remaining 95% of each class remain in colour. All images retain the same dimensions. Following the formulation of CIFAR-10S, in CIFAR-10M the gray-scale images are represented as one single color channel duplicated across all three channels.

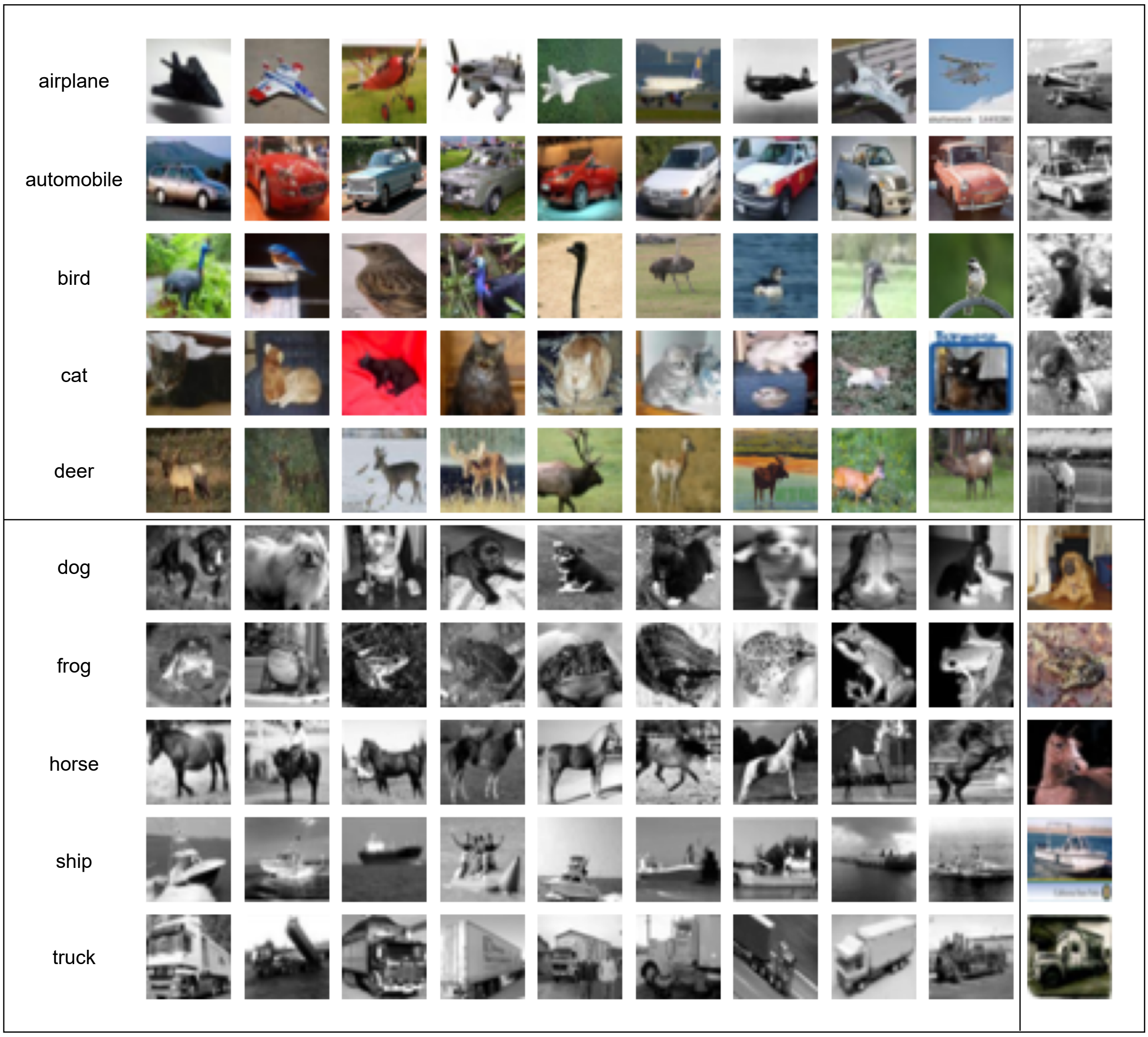


Figure 3.6: A sampling of 100 images, 10 from each class from CIFAR-10S [167] with $p = 0.9$ as an example of sensitive attribute bias equally distributed across the dataset but skewed across classes. Note that our experiments use $p = 0.95$.

### Visual bias benchmarks: CIFAR-10 Skewed

CIFAR-10S (Figure 3.6) proposed by Wang et al. [167] represents an example of sensitive attribute bias. Both values for the bias variable – presence or absence of color – are equally present in the dataset (50/50%). But for five out of ten target classes, 95% of the images are in gray-scale, resulting in an overall balanced dataset with respect to colour but a strong skew within each class. As a model can learn the presence or absence of colour as a class indicator, this is an instance of a sensitive attribute bias.

| **Age** | | **Gender** | | **Skin Color/Type** | |
|---|---|---|---|---|---|
| $\leq 45$ | $> 45$ | Female | Male | Lighter | Darker |
| 77.8% | 22.1% | 58.1% | 42.0% | 85.8% | 14.2% |

Table 3.3: The distribution of age, gender, and skin color/type for the widely used face image dataset CelebA [107] as found by [114], showing bias against older subjects, males, and darker skin tones. For the purpose of showing general trends, binary disjoint categories are shown for each attribute.

### Face detection dataset

Facial detection, or the binary classification task of deciding whether an image contains a face or not, is a real-world problem with significant implications. Alongside the goal of maximising task accuracy, it is also desirable that the model will perform equally across all subgroups. Subgroups are naturally present in facial detection images due to ethnic traits such as skin and hair color and facial features, in addition to other potentially correlated features including occlusions, accessories, and variations in pose and lighting.

Following a similar setup as Amini et al. [8], we create a face vs. no-face detection dataset using 20k instances of faces from CelebA [107] and 20k non-face samples from a variety of different object classes from ImageNet [41], for a training set of 40k images. CelebA includes a labeled selection of images of 10k celebrities collected from the internet. The labels for CelebA comprise 40 different identified physical attributes including “arched eyebrows", “attractive", “5-o-clock-shadow", “bangs", “goatee", “wavy hair", “smiling" and more, with boolean values for each. Yet despite the diversity in attributes, studies show a steep skew away from older subjects (an age bias), darker skinned subjects, and even a slight bias towards females (Table 3.3).

The bias mitigation problem is designed as following: the model is trained on the CelebA + ImageNet non-face subset, and tested on a much more diverse test set to check the model's discrimination against minority groups present in the test set. Amini et al. use the Pilot Parliaments Benchmark [22] for the test set; due to the inaccessibility of that dataset because of data privacy issues, and its lack of age diversity as all the subjects are government members and thus adults, we have chosen to evaluate instead on FairFace [84]. FairFace consists of 108,501 images collected primarily from the YFCC-100M FlickrDataset [156], a public curated dataset of over 100 million images and videos. The FairFace images also include gender, race, and age annotations, balanced across 7 race groups (Figure 3.7): White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, and Latino, and 9 age subgroups from “0-2" to “over 70". The high level of diversity is visible in Figure 3.7. It is also one of the first diverse facial recognition datasets to distinguish between various regions of Asia.

FairFace images, in addition to their inherent diversity, also differ from the training set in other ways: the photos are cropped closely around the subjects' face, often excluding the neck and parts of the protruding facial extremities. Very little of the background behind the face is visible. In contrast, CelebA images show the upper chest and neck, and leave a considerable margin between the face and the edge of the image, revealing the background behind the subject.

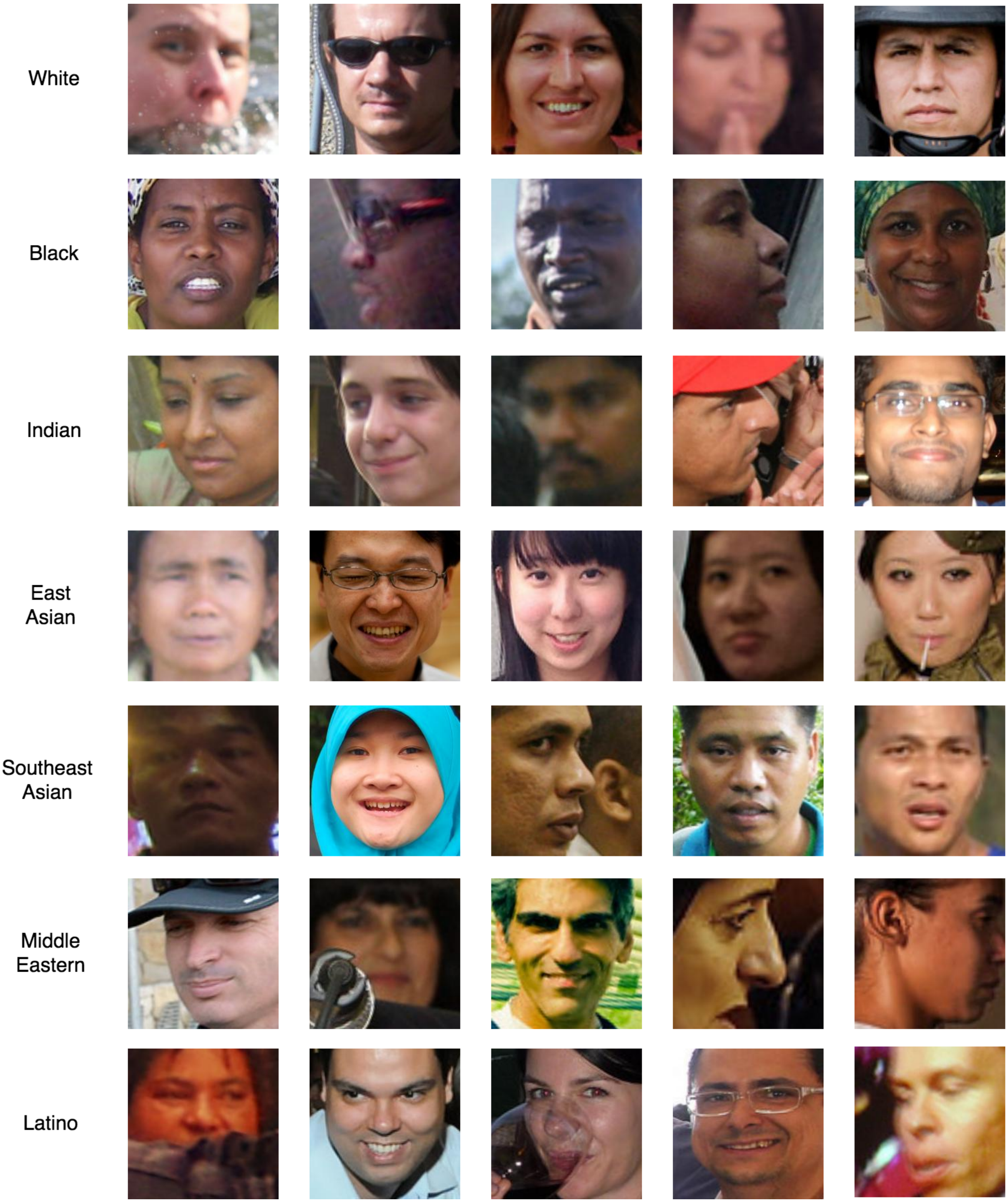


Figure 3.7: FairFace [84] is balanced across 7 racial groups: White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, and Latino.

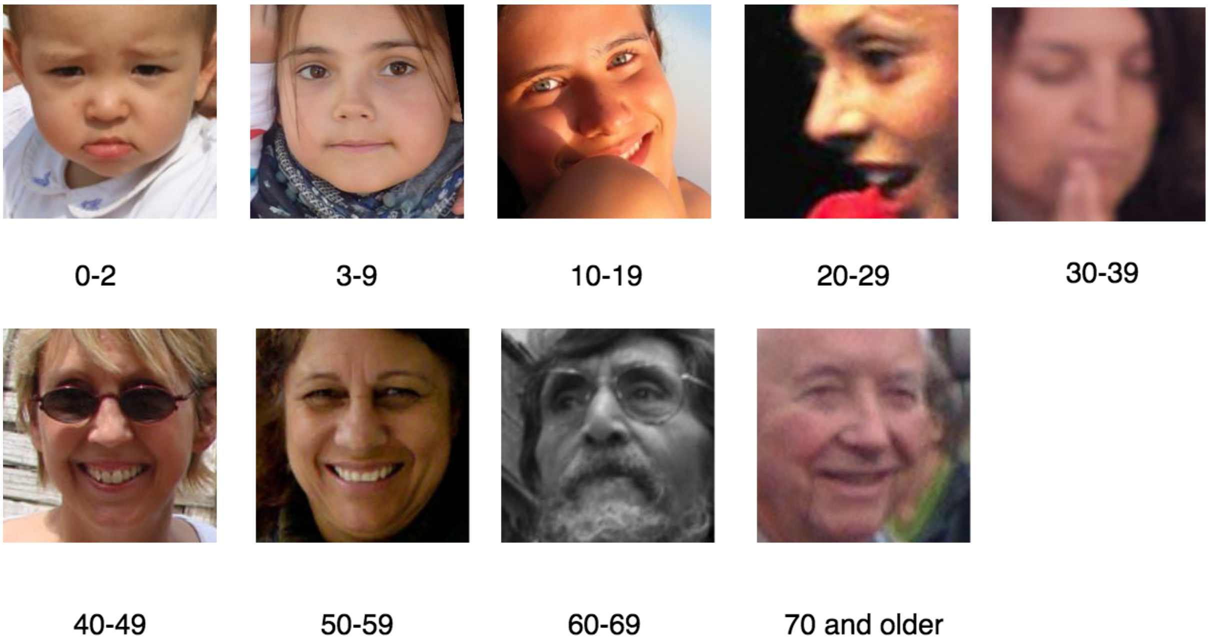
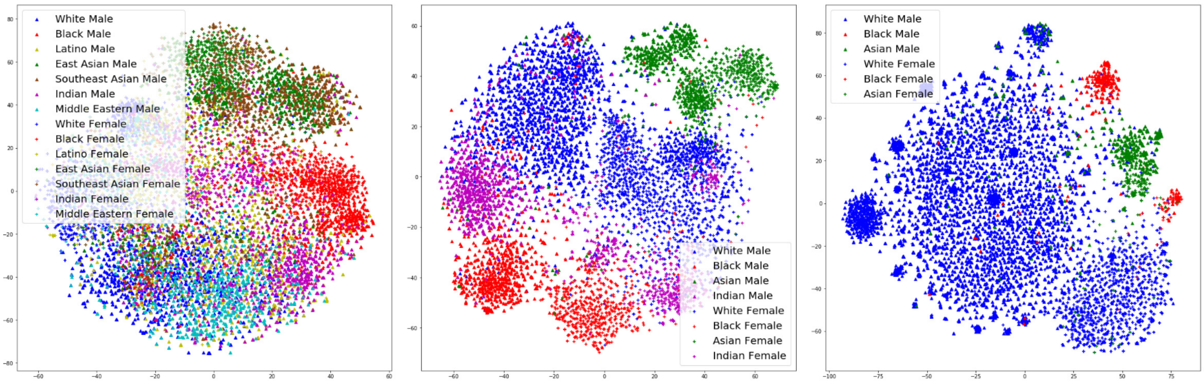


Figure 3.8: FairFace [84] is also balanced across seven age groups, from “0-2", “3-9", “10-19", “20-29", “30-39", “40-49", “50-59", “60-69", to “more than 70".



(a) FairFace (b) UTKFace (c) LFWA+

Figure 3.9: t-SNE visualizations figure from [84] illustrating the distribution of faces in two major facial recognition datasets (b) UTKFace [180] and (c) LFWA+, contrasted with the more balanced distribution and richer labels in (a) FairFace.

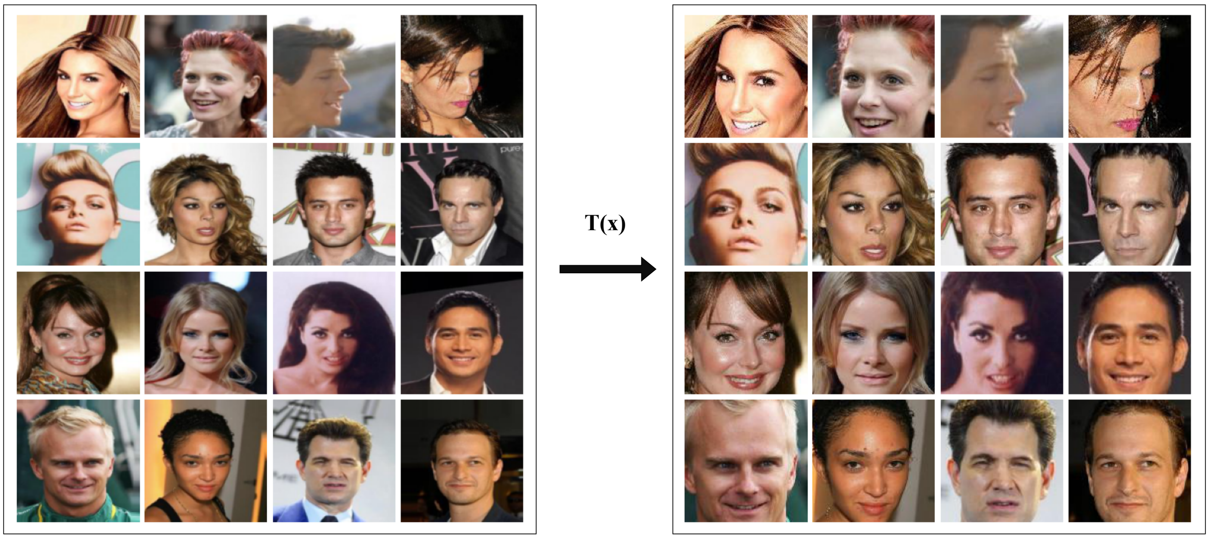


Figure 3.10: A random selection of 16 CelebA images before transformation *(left)* and the same images *(right)* after applying transformation $T$ which minimizes the average FID score between the CelebA and FairFace datasets.

Given this considerable distribution shift between the training and test data, which may also inadvertently introduce biases which we cannot be aware of, we attempt to minimize the factors not directly associated with any of our bias variables.

Firstly, to deal with the positioning issue, we perform a grid search for the optimal transformation function $T$ with respect to a similarity measurement. The Fréchet Inception Distance (FID) introduced in [71] was designed for evaluating generative adversarial networks (GANs), measuring the visual similarity between generated images and real ones. The FID compares the similarity between features extracted by an Inception v3 network trained on a large collection of real images (ImageNet) and those from the synthetic images. The mean and covariance of each of the features are compared using the Frechet or Wasserstein-2 distance, and is shown to strongly correspond to visual similarity.

No transforms other than resizing and cropping are applied to the FairFace data, ensuring that all the other diversity is still intact; the transforms bring the CelebA data in-distribution to the FairFace data without alleviating the sources of bias used to evaluate the fairness of the model. With the absolute difference between average FID scores of the CelebA and FairFace datasets guiding the search, we find the best $T$ to be a center crop to 124 x 124 plus a downsampling of all face and non-face training data to 64 x 64 such that the general resolution and positioning of the face in the images minimizes the difference between average FID scores. A sampling of the CelebA data before and after the transformation is shown in Figure 3.10.

### 3.6.2 Evaluation metrics

We evaluate using the following fairness metrics:

- **Test set metrics**: given test datasets for which the presence or absence of bias variables is known, regular target task metrics can be used to compare the difference in performance between bias-aligned and bias-conflicting samples; an unbiased model will perform equally well on both.

- **Bias amplification score** [181]: this metric was proposed first in a natural language processing setting and is generalizable to other problems. For the set of all classes C, where $Gr_c$ is the number of grayscale test set examples predicted to be of class $c$, and $Col_c$ the same for color, the mean bias amplification score is:

$$\text{bias ampl.} = \frac{1}{\mid C \mid} \sum_{c \in C} \frac{\max(Gr_c, Col_c)}{Gr_c + Col_c} - 0.5 \tag{3.15}$$

- **Equality of opportunity** [65] is satisfied when for a prediction $\hat{y}$, some class $y$, and sensitive attribute $a$, $P(\hat{y} = y \mid a = 0) = P(\hat{y} = y \mid a = 1)$. A difference in equality of opportunity score is derived as per [17], where Y is the set of possible classes, $TN^a_y$, $FN^a_y$, $TP^a_y$ and $TP^a_y$ are the number of true and false negatives and positives of class $y$ with protected attribute $a$, and $N^a$ is the total number of test set samples with attribute $a$:

$$\frac{1}{\mid Y \mid} \sum_{y \in |Y|} \left| \frac{TP^1_y}{TP^1_y + FN^1_y} - \frac{TP^0_y}{TP^0_y + FN^0_y} \right| \tag{3.16}$$

- **Equalized odds** [65] is a relaxed form of equality of opportunity where subgroups have equal true positive and false positive rates; equalized odds requires non-discrimination on only one desired outcome. A difference of equalized odds score as per [16] is as follows:

$$0.5 * \left( \left| FPR^1_y - FPR^0_y \right| + \left| TPR^1_y - TPR^0_y \right| \right) \tag{3.17}$$

### 3.6.3 Inference

To provide a fair comparison, we follow the same training and architecture choices as proposed, with the exception of the learning rate, which is set to maximise performance for the cyclical step size schedule of our Bayesian formulation. Using validation loss to choose hyperparameters, we find the optimal validation loss with 280 epochs and four cycles. With the exception of our reproduction of the DB-VAE method in [8], all experiments use a ResNet-18 convolutional neural network as the base architecture. For the DB-VAE, we follow the same architecture as used in the original paper, adapted for multi-target classification on CIFAR-10S, up-sizing the images to 64 x 64 to match the expected input dimensions, and grid search to find the optimal value for de-biasing parameter, $\alpha = 0.001$.

For the face detection dataset, we train a regular ResNet18 to convergence at 30 epochs using an 8:2 split of the CelebA/ImageNet dataset for training and validation and a standard SGD optimizer with learning rate of 0.01. The cSG-MCMC Bayesian model with uncertainty-weighted loss is similarly trained with 4 cycles of 30 epochs each for a total of 120 epochs.

## 3.7 Results

| Model | Bias (↓) | Mean acc (%,↑) | Opp. (%,↓) | Odds. (%,↓) |
|---|---|---|---|---|
| | | *Explicit method results* | | |
| S-Sampling | 0.066 | 89.1 ± 0.4 | 12.58 ± 0.2 | 6.91 ± 0.1 |
| Adversarial (1) | 0.101 | 83.8 ± 1.1 | 16.71 ± 1.4 | 9.28 ± 0.7 |
| Adversarial (2) | 0.094 | 84.1 ± 1.0 | 14.13 ± 1.4 | 7.89 ± 0.8 |
| DomainDiscrim | 0.040 | 90.3 ± 0.5 | 7.27 ± 0.3 | 4.02 ± 0.2 |
| DomainIndepend [167] | **0.004** | **92.9 ± 0.1** | **1.07 ± 0.2** | **0.59 ± 0.1** |
| FeatureLabel [157] | **0.004** | **91.5 ± 0.2** | **0.83 ± 0.1** | **0.46 ± 0.1** |
| | | *Implicit method results* | | |
| Baseline | 0.074 | 88.5 ± 0.3 | 13.07 ± 0.4 | 7.19 ± 0.2 |
| DB-VAE [8] | 0.167 | 90.2 ± 0.4 | 6.87 ± 0.5 | 0.78 ± 0.2 |
| cSG-MCMC+EpiUpWt | **0.037** | 89.1 ± 0.2 | 12.12 ± 0.2 | 6.26 ± 0.2 |

Table 3.4: Multi-class classification, bias amplification score, mean bias accuracy, equality of opportunity and equalized odds for bias benchmark dataset CIFAR-10S, a dataset with sensitive attribute bias. Note that all methods except for the baseline (a regular deterministic network with no bias mitigation), the DB-VAE, and our method, are explicit bias mitigation methods. **First**, **second**, and third best results formatted.

The results of our de-biasing method on this dataset are presented in Table 3.4. We benchmark against various other explicit and implicit bias mitigation methods, including benchmarks established by [167, 157, 8]. The explicit methods using the annotated bias variables in the training dataset to de-bias are as follows:

- **Sub-sampling / re-sampling / re-balancing**: minority training subgroups are over-sampled such that the data appears balanced for each class.

- **Adversarial training**: the model is trained to minimize classification loss while simultaneously minimizing model's ability to predict the bias variable. Two variants are used (1) a uniform confusion loss $-(\frac{1}{|D|})\sum_d \log q_d$ [159, 7] and (2) loss reversal $\sum_d \mathbb{I}\left[\hat{d}=d\right]\log q_d$ with gradient projection from [178]. [167] perform further experiments to validate the relatively poor results for both adversarial learning methods, and show that enforcing bias confusion also inadvertently increases undesired class confusion.

- **Domain discriminative training [46]**: where there are $C$ target classes and $G$ subgroups with unique bias variable combinations, a $G$*$C$-way discriminative classifier is trained, then a prior shift towards a uniform test distribution (assumed, as true test label distribution cannot be used) applied to remove the correlations between $C$ and $G$ at inference time.

- **Domain independent training [167]**: a single shared feature representation network with an ensemble of classifiers, one per subgroup; at inference, the class decision boundaries are averaged over all the classification head outputs for subgroup domains $g$ and class $\hat{y} = \arg\max_y \sum_g s(y,g,x)$.

- **Feature labeling [157]**: the bias direction for each known bias variable in the feature

| | **Baseline** | **EpiUpWt** |
|---|---|---|
| TPR Color (%),↑) | 92.1 ± 0.1 | 93.8 ± 0.1 |
| TPR Gray (%),↑) | 91.6 ± 0.1 | 93.3 ± 0.1 |
| TPR Gap (%),↓) | 1.8 | 0.5 |

Table 3.5: The uncertainty-weighted loss reduces the TPR disparity between minority and majority groups on minority bias dataset CIFAR-10M.

space is identified, and the effects of biases are removed in both feature and target class spaces.

And the implicit methods:

- **Baseline**: a deterministic CNN optimized using regular cross entropy classification loss.
- **De-biasing variational auto-encoder (DB-VAE) [8]**: an implicit over-sampling method where a model learns latent features and uses the relative sparsity of that latent space to adjust sampling probabilities of training samples while training such that under-represented regions of training data are over-sampled.

While not competitive with all bias-informed methods, our approach demonstrates an ability to de-bias blindly on both the benchmark dataset with sensitive attribute bias (CIFAR-10S), and our constructed dataset with minority group bias (CIFAR-10M).

Via a manual examination of the training set and predictive uncertainties post-training, for CIFAR-10S, we find that samples with a sensitive attribute have higher uncertainties. They constitute 20% of samples in the samples with the highest 10% uncertainties, while only 5% of samples in the dataset have a sensitive attribute. In contrast, less than 2% of samples in the lowest 10% uncertainties have a sensitive attribute.

Table 3.5 shows that the method also decreases performance gaps for CIFAR-10M by 66% and bias amplification score, equalized odds, and equalized opportunity scores of 0.002, 0.65, and 0.70 respectively.

For every subgroup, the uncertainty-weighted loss decreases the TPR gap, with the discrepancies for the 7 subgroups with lowest TPR rates shown in Figure 3.11.

Given that sample-level weighting by a factor of $N$ during training is equivalent to that sample appearing $N$ times, our approach could be categorized as a type of sub-sampling algorithm. Thus, it suffers from the same weakness as all sub-sampling algorithms, a tendency to overfit over-sampled data. This can only be partially mitigated by aggressive data augmentation. We hypothesize that this explains why increasing tunable de-biasing parameter $\kappa$ beyond the optimal value results in worse performance as shown in Figure 3.12.

Table 3.6 compares the performance of the deterministic baseline model against the Bayesian variant of the same model and the proposed weighted loss for the CIFAR-10S benchmark dataset. The ablation study shows that the Bayesian model, while with a slightly lower bias amplification

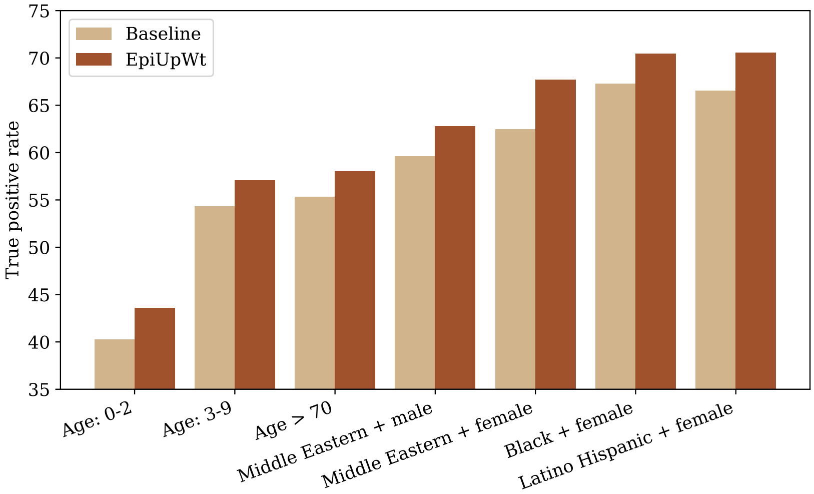


Figure 3.11: Performance discrepancies between baseline (deterministic model with no de-biasing) and Bayesian model with uncertainty-weighted loss for minority subgroups with lowest TPRs.

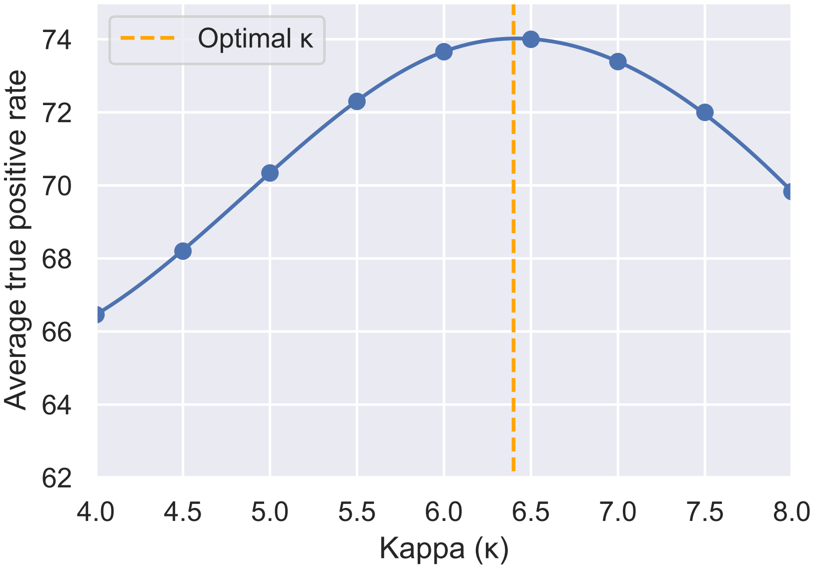


Figure 3.12: True positive rate (TPR) over the entire FairFace dataset as a function of tunable de-biasing parameter kappa $\kappa$, showing how the degree of de-biasing can be controlled by $\kappa$.

| | **Bias** (%,↓) | **Mean acc** (%,↑) |
|---|---|---|
| Baseline | 0.074 | 88.5 ± 0.3 |
| cSG-MCMC | 0.060 | 88.1 ± 0.2 |
| cSG-MCMC+EpiUpWt | 0.037 | 89.1 ± 0.2 |

Table 3.6: Ablation study on CIFAR-10S showing results of a Bayesian cSG-MCMC network with regular unweighted cross-entropy loss.

score, has a lower mean accuracy as compared to the baseline. Simply using a Bayesian model does not mitigate biases. In contrast, the weighted loss improves upon both models.

Figure 3.13 shows samples with high uncertainties, which clearly have features which make them more likely to be subject to bias. A bias-informed method could strongly mitigate bias due to known societal biases such as gender and race, or skin phenotype. But since it would be unlikely to also have access to meta-data which identifies variances in lighting, pose, image resolution, etc., all of which also result in unfairness, such methods would not target such biases. Faces with high uncertainties are more likely to be subject to discrimination due to such variances.

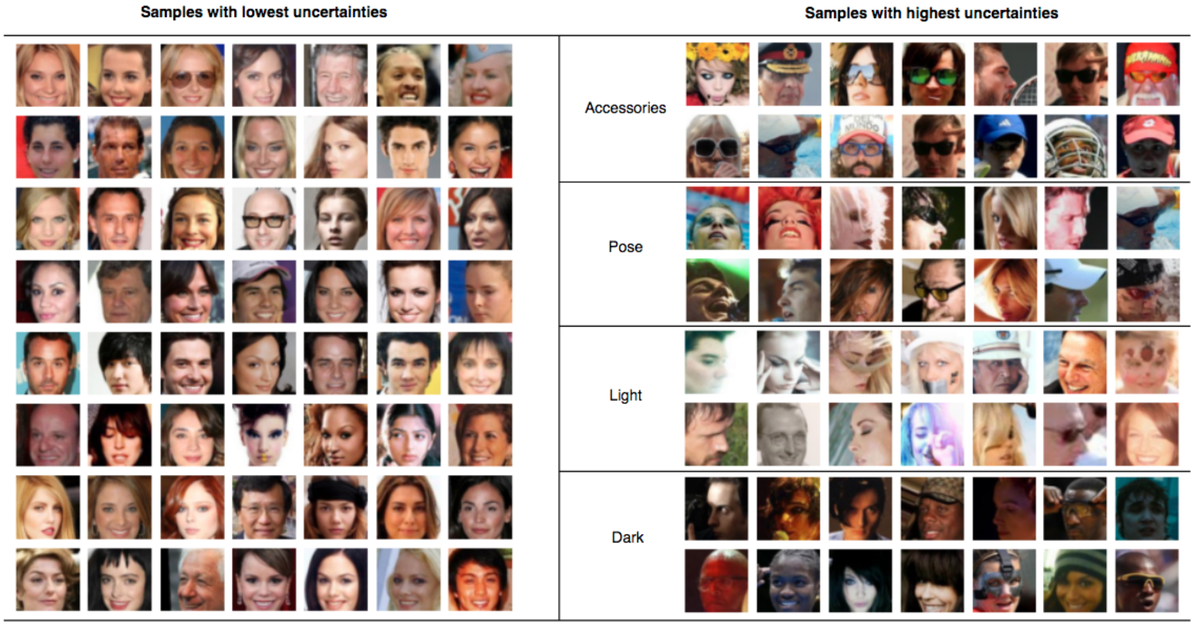

Samples with lowest uncertainties
Samples with highest uncertainties
Accessories
Pose
Light
Dark

Figure 3.13: Face training samples from CelebA with lowest predictive uncertainties (left) and faces with highest predictive uncertainties (right). The faces with low uncertainties tend to be well-lit, facing forward with hair cleanly framing the face, and primarily lighter-skinned with few obscuring accessories.

Such an approach is valuable in medical imaging applications with large population image analysis due to the inherent difficulty in collecting meta data. Sensitivity and privacy requirements result in imaging datasets with very few annotations and little, if any, associated patient meta data. This presents a challenge for bias-informed methods, and serves as motivation for further exploration of methods which can mitigate without requiring comprehensive knowledge of all biases.

## 3.8 Discussion and conclusion

We have presented the motivation for leveraging predictive uncertainties from the Bayesian paradigm for implicit bias mitigation, and shown that a simple predictive uncertainty-weighted loss function has potential for bias mitigation for datasets with unknown sources of bias. Additional results and evaluation for EpiUpWt on three additional classification datasets can be found in Chapter 4, and on a segmentation dataset in Chapter 5.

Some training datasets may contain an over-sampling from unprivileged groups, in which case the correlation with predictive uncertainties may no longer exist. Thus, this exploration focuses only on cases of minority group and sensitive attribute bias. While not competitive with all bias-informed models, this method is a step towards exploring how predictive uncertainties in Bayesian neural networks can be leveraged for identifying, understanding, and mitigating the types and sources of visual bias in data.

# Chapter 4

# Posterior estimate fine-tuning for bias mitigation

## 4.1 Overview

Having dynamically identified and leveraged the predictive uncertainty and bias correlation through cost-sensitive loss weighting during the training of a Bayesian neural network, in this chapter we explore a variation of this approach for bias mitigation which operates as a fine-tuning or post-training modification of the posterior estimates. We firstly explore where and how the uncertainty discrepancies arise in the network architecture, specifically for bias-conflicting samples. We consider the two components of a CNN, the representation component and the classification component, using network dissection introduced by Bau et al. [15] to identify convolutional kernels which focus on the bias variables. Based on this exploration, we propose a fine-tuning procedure which modifies the posterior estimate of the Bayesian neural network via a loss operating on each posterior sample individually. The method is driven by the sample epistemic uncertainties, weighting the learning step size based on the variance of the sample predictive distribution. We perform experiments which compare the effects of a regular cross-entropy loss versus a regularized version, and propose an explanation of why our composite regularized loss works better than the cross-entropy loss alone.

We select three challenging datasets with a variety of other existing benchmarks to demonstrate that Bayesian neural networks with modified posterior estimates perform comparably to, if not better than, prior existing methods and show potential worthy of further exploration.

## 4.2 Posterior estimate fine-tuning

### 4.2.1 Exploring uncertainty discrepancies

Deep neural network models trained on biased data encode biased attributes in their feature representation component, otherwise known as the encoder. De-biasing the encoder requires firstly comprehensive knowledge of the bias variables, instance-level annotations or labels for all training inputs with respect to these biases, and then a method which discourages those correlations from being learned.

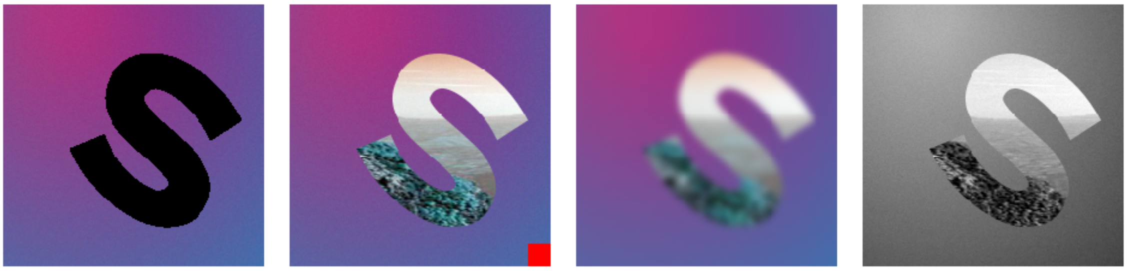


Figure 4.1: Four samples based on a single generated Synbols image from class “s" of our toy dataset, each with a different controllable bias variable *(left to right)*: the letter itself, spurious square placed in random corner, resolution of whole image, and gray-scale.

| **Parameter** | **Definition** |
|---|---|
| $A_{m,k}(\boldsymbol{x})$ | The activation map for the $m^{th}$ posterior estimate and the $k^{th}$ convolutional kernel, derived from input image $\boldsymbol{x}$ |
| $\hat{A}_k(\boldsymbol{x})$ | The pixel-wise mean activation map across all posterior estimates for the $k^{th}$ convolutional kernel |
| $f$ | An identified feature in the dataset |
| $\mathrm{IoU}_{k,f}$ | The intersection-over-union for the $k^{th}$ convolutional kernel and feature $f$ |
| $t$ | The threshold for determining whether kernel $k$ is a high-activator for feature $f$ |
| $\sigma_k(\boldsymbol{x})$ | The kernel uncertainty, the maximum pixel-wise variance across activation maps from posterior samples |

Table 4.1: Notation for a Bayesian neural network dissection.

Du et al. [45] argue that even with a learned “biased representation”, a model can be made more fair by focusing on the classification component alone. The goal of their proposed approach, Representation Neutralization for Fairness (RNF) is to neutralize the representations of input samples of different sensitive attributes in feature space, and then use those new representations to re-train or fine-tune the classification head of the DNN. RNF discourages the classification layer parameters from learning the undesired bias correlations, and instead focuses on learning core features.

We are motivated by a similar end-goal – a classification layer which has not learned bias correlations - but without explicitly de-biasing the feature representations. As bias-conflicting samples tend to have higher epistemic uncertainties than their bias-aligned counterparts [59, 3, 152], we first explore which component of the DNN - the representation or the classification component – gives rise to the discrepancies in sample epistemic uncertainties.

Synbols (Synthetic Symbols) [97] is a tool for generating feature-rich synthetic datasets of customized resolution and size, using Unicode standard symbols, a font library of over 1,000 fonts, and various background and foreground textures. Synbols-generated data can be used for classification, segmentation, anomaly detection, and other tasks.

Using Synbols, we generate a dataset which we call Biased Synbols, to create a non-trivial binary classification task with controllable biases. The foreground of each 224 x 224 image is a character displayed in a font chosen randomly from a large selection of fonts, and the texture of the character is a random natural scene, cropped to fit the character mask. The background behind the character is a 2-color gradient. We introduce four types of bias as shown in Figure 4.1. In order to control the bias and isolate its effect, we only consider one at a time for a given dataset, but study four datasets each with a slightly different bias variable to confirm consistency of results across bias types. Specifically, we choose a spatially distinct bias (the square in the corner) so that we can use an interpretability method to understand what the network has learned.

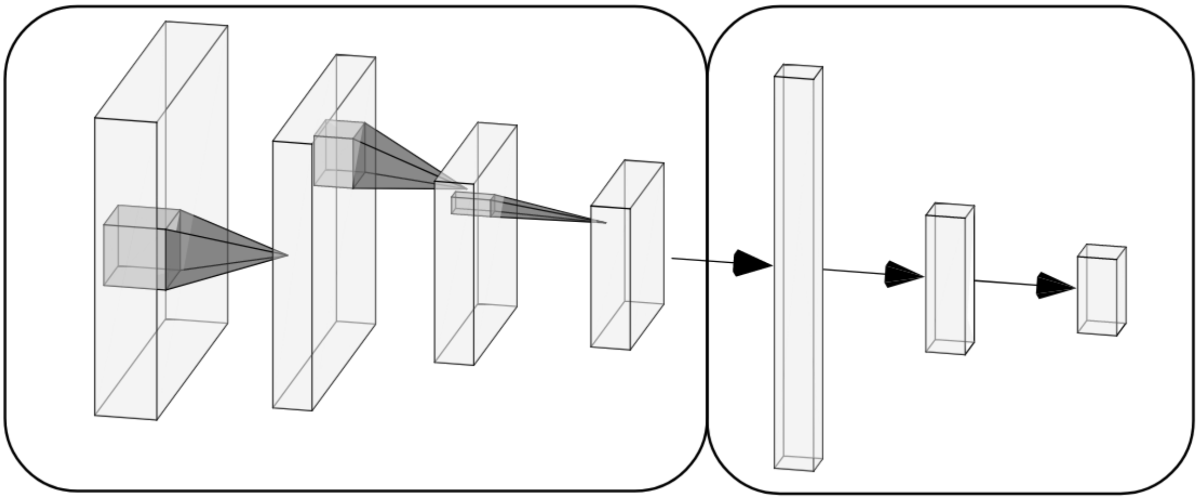


Figure 4.2: The representation component of a typical CNN architecture (*left*), resulting in flattened features which feed into the classification component (*right*) composed of several linear layers.

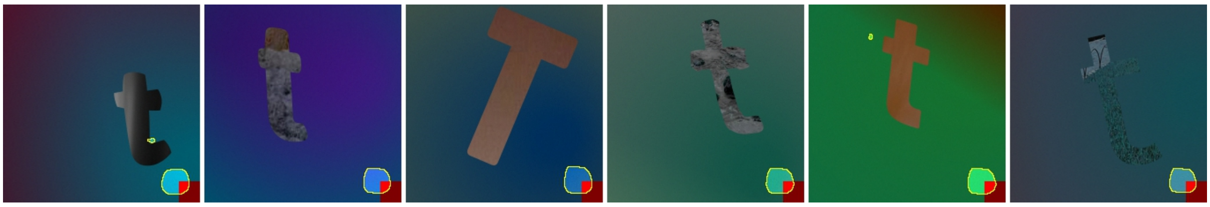


Figure 4.3: The top six training images for which kernel 10, convolution layer 2 *(bottom)* from AlexNet most strongly activates. The kernel has identified the spurious feature, the red square in the bottom right corner, and has the highest $\text{IoU}_{k,f}$ for the mask of bias feature $f$, not the character mask.

This setup allows us to dissect the network. Network dissection, proposed by Bau et al. [15], measures the alignment between convolutional kernel activations and any concept, which can be defined by segmentation masks on the dataset. The method allows for interpretable understanding of the role of individual kernels without any model re-training or fine-tuning. Bau et al.

show that individual kernels can learn specific unlabelled but semantically meaningful features of the input space, and that kernel alignment to specific features can be measured.

For every input image $\boldsymbol{x}$, the activation map $A_k(\boldsymbol{x})$ of every convolutional kernel $k$ is computed via a forward pass. This results in a set of activation maps $\mathbf{a}_k = A_k(\boldsymbol{x}) \; \forall \; \boldsymbol{x} \in D$, for each kernel. Then, the top quantile level $Q_k$ is determined such that $P(\mathbf{a}_k > Q_k) = 0.005$. For an input image and kernel pair, the activation map is upscaled via bilinear interpolation, anchored at the center of the kernel's receptive field, to match the input image size. The threshold $Q_k$ is used to create a binary segmentation mask from the activation map.

We distinguish between the core feature (the segmentation of the character only), and the bias feature (the red square in the corner). A kernel $k$ is considered a detector or high-activator for a feature $f$ if the intersection-over-union $\mathrm{IoU}_{k,f} > t$ for some threshold $t$. We extend this method to dissection of a Bayesian neural network by taking the pixel-wise mean activation across all posterior estimates (Table 4.1, $\hat{A}_k(\boldsymbol{x}) = \frac{1}{M}\sum_m A_{m,k}(\boldsymbol{x})$ instead of $A_k(\boldsymbol{x})$). This dissection identifies convolutional kernels in the representation component which are most strongly activated by specific features, such as core features (Figure 4.4) or bias features (Figure 4.3). We generate

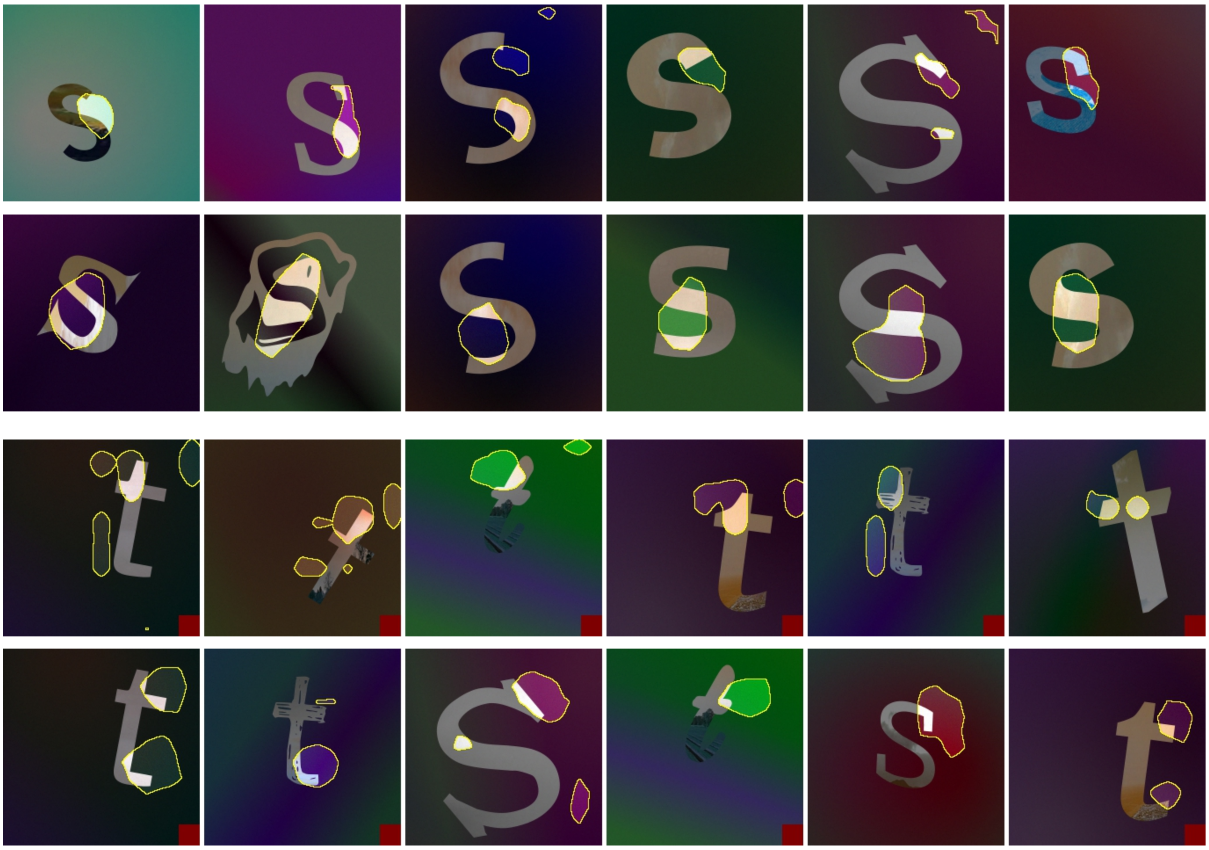
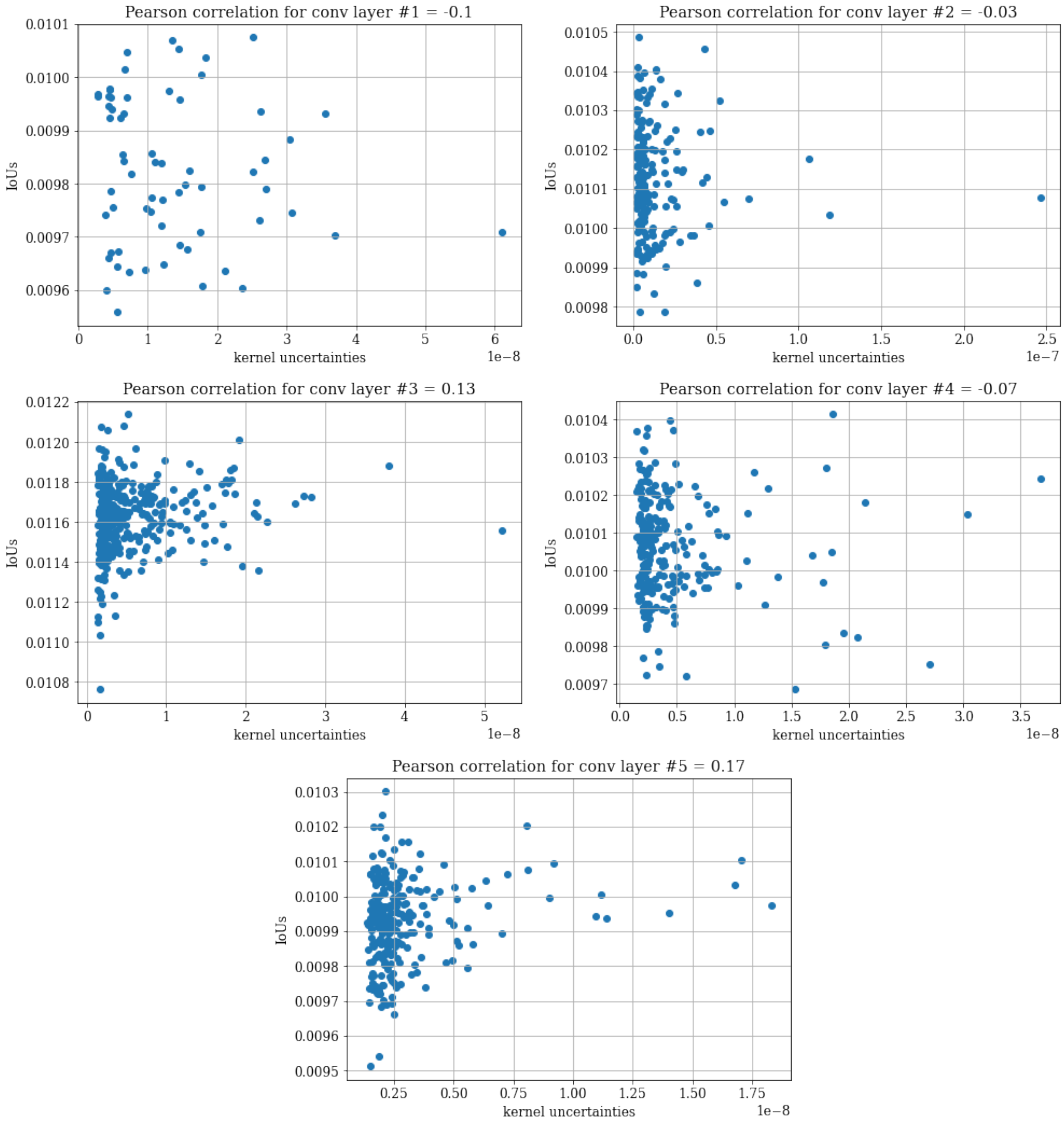


Figure 4.4: The top six training images for which four kernels in the second convolutional layer of AlexNet most strongly activate, with the corresponding dissection maps. All four kernels show strongest activation for regions overlapping with the core feature, the character mask. The bottom two rows show image samples for the latter two kernels which also are from the majority group, with the spurious bias feature in the corner, yet the kernels in question do not strongly activate to the bias. In these cases, the kernels pay more attention to the characters.



Pearson correlation for conv layer #1 = -0.1

Pearson correlation for conv layer #2 = -0.03

Pearson correlation for conv layer #3 = 0.13

Pearson correlation for conv layer #4 = -0.07

Pearson correlation for conv layer #5 = 0.17

Figure 4.5: Intersection-over-union (IoU) of each kernel $k$ in each of the five convolutional layers of the Bayesian AlexNet, plotted against the kernel uncertainty estimations. Pearson correlation scores show no clear correlation exists; the uncertainties of kernels which activate more strongly for the bias feature are not consistently higher than those which activate for non-bias features, indicating that the observed predictive uncertainty discrepancies between minority and majority input samples arise from the classification component of the network.

a two letter (“s" and “t") binary classification task with train/val/test splits of 20k/5k/5k and train a Bayesian AlexNet on four versions of Biased Synbols, each with a different bias variable and $p_{bias} = 0.95$ for the “s" class and $p_{bias} = 0.05$ for the “t". We use the validation set to choose the optimal stopping point, and set $M = 10$ for $M$ posterior samples during the sampling phases of each cycle for a posterior estimate. We then compute the sample epistemic uncertainty for each test sample. In each case, the bias-conflicting samples have higher mean group uncertainties than the bias-aligned (a mean increase by 0.15, 0.16, 0.09, and 0.20 for the four variables shown in Figure 4.1 respectively, for uncertainties normalized via softmax between 0.0-1.0).

Next, we identify where the discrepancies in uncertainties arise from in the network. We find indiscernible difference in uncertainties between samples by averaging or taking the maximum of the uncertainties across the features extracted directly after the final convolutional layer (the output of the representation component in Figure 4.2). To further confirm this hypothesis that learning a biased representation does not affect uncertainties at the feature level, we focus on Biased Synbols with the spurious square in a random corner. We choose this variant of Biased Synbols since the bias attribute is spatially distinct from the core feature, the character.

We then estimate the kernel uncertainties; for each input $\boldsymbol{x}$, we compute the pixel-wise variance over the activation map distribution and define the kernel uncertainty $\sigma_k(\boldsymbol{x})$ as the maximum variance across the map, with mean kernel activation map $\hat{A}_k(\boldsymbol{x})$:

$$\sigma_k(\boldsymbol{x}) \approx \max\left(\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(A_{m,k}(\boldsymbol{x}) - \hat{A}_k(\boldsymbol{x})\right)^2}\right) \tag{4.1}$$

For every convolutional layer in AlexNet, we compute the intersection-over-union of the binary mask of the square with the thresholded kernel activation heatmaps, and measure correlation between these IoUs and the kernel uncertainties from Equation 4.1. This plot, for all kernels in each of the 5 convolutional layers of AlexNet, is shown in Figure 4.5.

The Pearson correlations per convolutional layer are -0.10, -0.03, 0.13, -0.07, and 0.17, none of which indicates a clear correlation between kernel uncertainties and stronger activations for the bias variable. Thus, simply learning features associated with bias variables does not seem to create discrepancies in uncertainties. These observations motivate our focus on the classification head and the predictive uncertainties induced by learning a biased feature weighting in the classification layers of the network.

### 4.2.2 Method

We propose a posterior fine-tuning procedure (Algorithm 3) which further fine-tunes the classification layers of each Monte-Carlo sample from the Bayesian posterior using a guiding loss function.

The representation portion of the network is frozen during the fine-tuning because (1) we aim to learn re-weighting of learned features in the classification head assuming the uncertainty discrepancies arise from this portion of the network, and (2) this significantly reduces the computational requirements. Thus, we refer to $\boldsymbol{\theta}^c$, where each posterior sample $\boldsymbol{\theta}_m = \{\boldsymbol{\theta}^r_m, \boldsymbol{\theta}^c_m\}$ is composed of the parameters from the representation and classification portions of the network respectively. Deterministic networks are often fine-tuned on the classification head only; similarly, support for operating on $\boldsymbol{\theta}^c$ alone can found in literature on “Bayesian last layer" networks [93].

We propose, firstly, a regular cross-entropy loss function with sample-wise weighting applied as in Chapter 3, and secondly, a regularizing component (see Table 4.2 for notation). Both losses are applied to individual posterior estimates. The weighting function causes the higher uncertainty samples to have larger loss contributions. A bias-aligned sample with low uncertainty in comparison has a small loss contribution. Each sample is weighted by $\boldsymbol{w}_i$ as a function of its predictive uncertainty (Equation 3.6) as shown in Equation 4.2. The distribution is shifted by 1.0 and scaling constant $\kappa$ controls the steepness of the function such that low uncertainty samples are never completely discounted, but only minimally shift the distribution. Note that batch reduction is also applied as usual, after the weighting.

| **Parameter** | **Definition** |
|---|---|
| $\boldsymbol{\theta}^r$ | The parameters of the representation component of the network |
| $\boldsymbol{\theta}^c$ | The parameters of the classification component of the network |
| $\boldsymbol{\Theta}$ | The set of M Monte-Carlo posterior samples $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ...\boldsymbol{\theta}_M\}$ |
| $\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}$ | The network with parameters from the $m^{th}$ posterior estimate |
| $L_{RCE}$ | PCGrad($L_R, L_{CE}$) |

Table 4.2: Notation for the fine-tuning procedure.

$$w_i = (1.0 + \sigma_{i,y_i})^{\kappa} \tag{4.2}$$

**Cross-entropy loss ($\boldsymbol{L_{CE}}$).** A regular negative log likelihood loss prepended by a softmax function moves individual posterior estimates towards the correct prediction. This can also be compared to artificially sharpening or tempering the posterior (see further discussion in Section 4.6). For the $m^{th}$ posterior sample $\boldsymbol{\theta}_m$ and input sample $\boldsymbol{x}_i$, the negative log likelihood measures the divergence between the true class and the predicted likelihood of each class. Equation 4.3 shows the negative log likelihood loss summing the predicted likelihood of the ground truth class for each sample, with one-hot encoded true target $\boldsymbol{y}_i$, all possible classes C, and the single posterior estimate output $\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x}_i)$. The weighting factor scales the final sample loss:

$$L_{CE} = L(\boldsymbol{y}_i, \boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x}_i), w_i) = -w_i \sum_{C} \boldsymbol{y}_i \cdot \log(\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x}_i)) \tag{4.3}$$

**$\boldsymbol{L_{CE}}$ with regularizing loss ($\boldsymbol{L_R}$).** While the cross-entropy loss $L_{CE}$ increases the likelihood of the correct target class, it is not as clear why $L_R$ should be considered. The regularizing loss $L_R$ is also computed separately for each posterior sample $\boldsymbol{\theta}_m$. $L_R$ is a negative log-likelihood with the true target $\boldsymbol{y}_i$ replaced by one-hot encoded $\hat{\boldsymbol{y}}_i$ (Equation 3.5), the argmax over the predictive mean distribution, capturing the divergence between the prediction of *a single posterior estimate* and the mean prediction of the whole posterior (Equation 4.4). This value is greater for posterior samples with predictions $\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x}_i)$ further away from the mean prediction, regardless of whether the mean prediction is correct or not. For a given sample, the regularizing loss alone narrows or sharpens the distribution around the mean prediction. The equation for the $L_R$, summing the predicted likelihood of the ground truth class for each sample, is as follows in Equation 4.4:

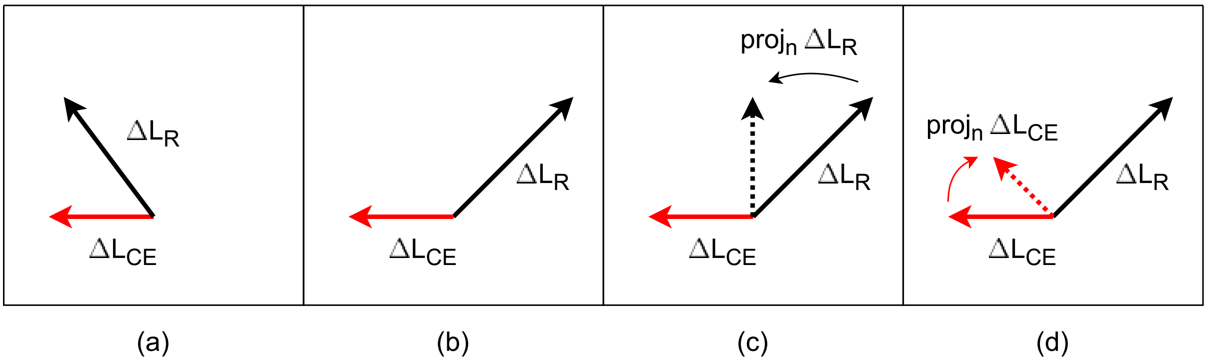


Figure 4.6: The interaction between the two loss gradients $\Delta L_{CE}$ and $\Delta L_R$ under PCGrad [177]: (a) non-conflicting gradients, simply added together unaltered; (b) conflicting gradients; (c) projection of $\Delta L_R$ onto the normal of $\Delta L_{CE}$; and (d) the projection of $\Delta L_{CE}$ onto the normal of $\Delta L_R$. Note that the additive results are not shown here.

$$L_R = L(\hat{\boldsymbol{y}}_i, \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i), w_i) = -w_i \sum_C \hat{\boldsymbol{y}}_i \cdot \log(\boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i)) \tag{4.4}$$

As the two loss gradients may conflict, the question of how to combine the two gradients arises. This problem also exists in multi-task learning where different tasks are learned in models with shared structure simultaneously. One gradient modification method or gradient surgery which is effective in multi-task learning is called *projecting conflicting gradients* (PCGrad) [177], whereby the authors posit that a significant issue in multi-task learning optimization is due to what they define as conflicting gradients. Specifically, two gradients are conflicting if they are pointing in opposite directions, i.e., have negative cosine similarity, and thus combining them has the potential to nullify the effect of both. The combining of such conflicting gradients is detrimental in particular if there is:

- an objective landscape with high curvature, and
- a large difference in their magnitudes.

In such settings, the optimizer struggles to take a productive step for both objectives. PCGrad projects the gradient of one loss onto the normal plane of the other when the gradients have negative cosine similarity. For each input sample in each mini-batch, gradients for losses $(L_{CE}, L_R)$ are computed. If there is destructive interference, one gradient is selected at random and projected onto the normal plane of the other (Figure 4.6). In the case of constructive interference, both gradients are combined as usual. The cumulative update to the weights is then averaged over the batch. From this point, we refer to the combined losses as a regularized cross-entropy loss, or $L_{RCE} = \text{PCGrad}(L_R, L_{CE})$.

In all our experiments, we find that this regularized loss $L_{RCE}$ outperforms $L_{CE}$ alone in both overall accuracy and accuracy for minority subgroups. An ablation study showing experimental support for the choice of gradient surgery over $L_{CE}$ alone or a simple weighted sum of the two losses is shown in Table 4.9. In Section 4.5, we propose two hypotheses for why $L_{RCE}$ performs better and conduct experiments testing each hypothesis.

**Algorithm 3** Fine-tuning procedure on Monte-Carlo estimate of posterior $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ...\boldsymbol{\theta}_M\}$

```
Require: Training data $X, Y$, neural network $\boldsymbol{\Phi}$, update step size $\epsilon$, posterior estimates $\boldsymbol{\Theta}$
for each fine-tuning iteration, $\forall \boldsymbol{x}_i \in X$ do
    $\boldsymbol{\mu}_i = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i)$    ▷ compute mean predictions

    $\boldsymbol{\sigma}_i = \sqrt{\frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{\Phi}_{\boldsymbol{\theta}_m}(\boldsymbol{x}_i) - \boldsymbol{\mu}_i)^2}$    ▷ compute sample uncertainties
    for each $\boldsymbol{\theta}_m = \{\boldsymbol{\theta}_m^r, \boldsymbol{\theta}_m^c\} \in \boldsymbol{\Theta}$ do    ▷ examine each posterior sample separately
        $\boldsymbol{w}_i = (1.0 + \boldsymbol{\sigma}_{i,y_i})^{\kappa}$
        $\hat{\boldsymbol{y}}_i = \text{argmax}(\boldsymbol{\mu}_i)$    ▷ compute predictions
        $L_{CE} = L(\boldsymbol{y}_i, \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i), \boldsymbol{w}_i)$    ▷ cross-entropy loss
        $L_R = L(\hat{\boldsymbol{y}}_i, \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x}_i), \boldsymbol{w}_i)$    ▷ regularizing loss
        $\boldsymbol{\theta}_m^c \leftarrow \boldsymbol{\theta}_m^c - \epsilon\text{PCGrad}(L_R, L_{CE})$    ▷ final update to individual posterior sample
    end for
end for
```

## 4.3 Experiments

### 4.3.1 Datasets

We experiment using three datasets with benchmarks established by Shreshtha et al. [143] for visual bias mitigation.

**Biased MNIST**

Biased MNIST [142, 143](see discussion of the differences between the two versions in Section 3.6.1) is a challenging benchmark bias dataset for assessing analysis of multiple bias sources. The dataset encodes contextual biases into a 5 x 5 grid of cells, one of which contains one of the target MNIST 10 digits. Each image is 160 x 160 pixels. Bias variables include a) digit size/scale (number of grid cells the digit occupies), b) digit color, c) type of background texture, chosen from simple patterns such as horizontal, vertical, or diagonal dashes of varying width and spacing, d) background texture color, e) co-occurring letters from a standard alphabet, and f) colors of the co-occurring letters. Figure 4.7 shows majority examples from the training set and uncorrelated examples from the test set; majority examples have digit classes correlated with specific co-occurring letters, while minority examples have random co-occurring letters.

With $p_{bias} = 0.95$, each digit co-occurs 95% with each bias source. The test and validation sets are unbiased with a 50K/10K/10K train/val/test split. For evaluation of implicit bias mitigation methods, using an unbiased validation set could prematurely stop the overfitting to the training set (and over-learning of bias variables). Experiments shown in Tables 4.6, 4.7, 4.8, and 4.10, use the official published validation set for fair comparison against the existing benchmarks; however, results using a validation set matching the biased distribution in the training set are also shown in Section 4.4. The validation set is not used to stop the fine-tuning for BayResNet+$L_{RCE}$, only for hyperparameter selection for the baseline Bayesian model, BayResNet, and we discuss the effect of using a biased validation set in Section 4.4.2.

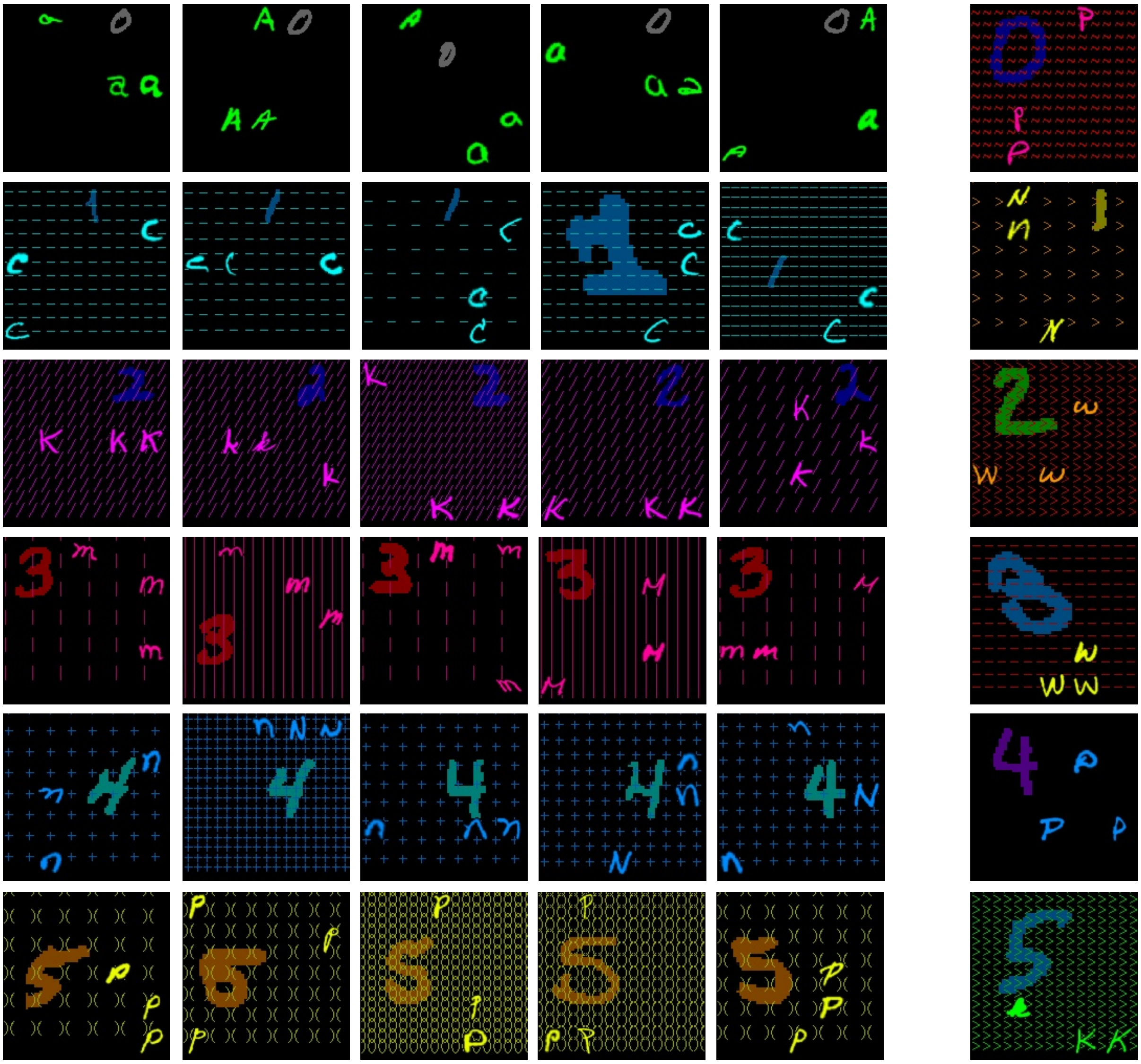


Figure 4.7: Bias-aligned samples from Biased MNIST training set with $p_{bias} = 0.95$ (*left*), and uncorrelated, bias-conflicting samples from the test set (*right*).

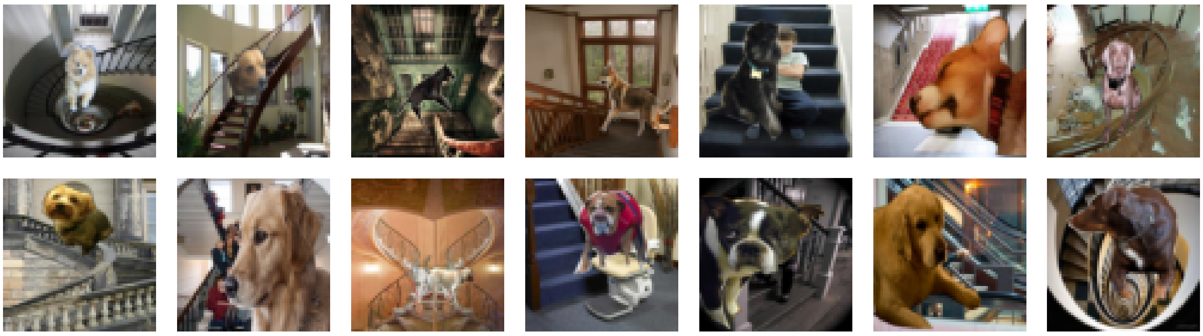


Figure 4.8: Samples from the training set of COCO-on-Places where foreground objects are all correlated with a background category; the “dog" object is paired with the “staircase" background.

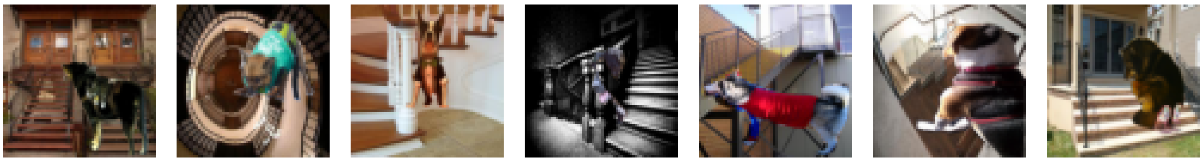


Figure 4.9: COCO-on-Places in-distribution test set samples from the “dog" class, with the same “staircase" background pairing as in 80% of occurrences in the training set.

### COCO-on-Places

Ahmed et al. [2] compose a dataset which correlates background images from the Places [183] dataset with COCO [106] object foregrounds (Figure 4.8). They also consider several scenarios to evaluate whether a model has learned *invariant features*, i.e., core features which are independent of biases. Towards these aims, they provide four test sets: in-distribution - background and foreground correlations matching those that appear in the training set (Figure 4.9), non-systematic shift - target foregrounds on previously unseen backgrounds (Figure 4.10), systematic shift - the training distribution backgrounds are used, but never with the training set foreground (Figure 4.11), and anomaly detection. For our purposes of bias mitigation, we consider the first three.

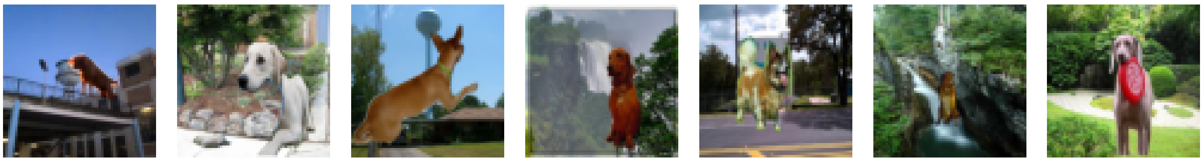


Figure 4.10: COCO-on-Places out-of-distribution test set samples from the “dog" class, where foreground objects are paired with backgrounds from categories which do not appear at all in the training set.

Each of the 9 foreground objects includes 800 images, and the test set has 100 images per category. The pairings are shown in Table 4.3. We use the official validation splits from Ahmed et al. [2].

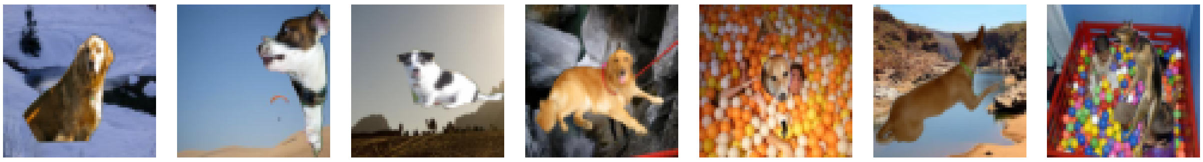


Figure 4.11: COCO-on-Places systematic shift test set samples from the “dog" class, where foreground objects are paired with previously seen but not correlated backgrounds; i.e., not paired with “staircase" backgrounds.

| | **train** | | **validation and test** | |
|---|---|---|---|---|
| | *majority* | *minority* | *validation bgs* | *unseen test bgs* |
| **boat** | beach | kasbah | oast house | water tower |
| **plane** | canyon | lighthouse | orchard | waterfall |
| **truck** | building | pagoda | viaduct | zen garden |
| **dog** | staircase | rock arch | | |
| **zebra** | desert | | | |
| **horse** | crevasse | | | |
| **bird** | bamboo | | | |
| **train** | broadleaf | | | |
| **bus** | ball pit | | | |

Table 4.3: The 9 COCO-on-Places foreground object categories *(far left)* alongside their majority and minority backgrounds; for the training set, the majority objects on far left in bold are paired with the background classes in the second column. Note no particular pairing for minority, validation, and unseen test backgrounds. Non-majority objects are randomly paired with one of the specified backgrounds.

### Biased Action Recognition (BAR)

BAR [119] includes manually selected real-world images of action-background pairs with realistic bias scenarios. For each of the six action classes, the backgrounds are correlated to the action as shown in Table 4.4. In addition, BAR is a small dataset as might be found in specific domains such as medical imaging where larger datasets are difficult to collect. As no split is consistently used in literature, we randomly define a training and validation set split of 1641/300 with the official test set of 654 images.

Figure 4.12 shows training samples from the “climbing" action class with bias towards backgrounds of buildings and other structures; Figure 4.13 similarly shows correlations between the “racing" action class with bias towards race track backgrounds; and Figure 4.14 shows correlations between the “throwing" action class with bias towards sporting field backgrounds. Each figure also shows samples from the test set, where a bias-conflicting background is present.

## 4.3.2 Evaluation metrics

We evaluate model fairness for each test dataset and model in terms of test set classification accuracy, averaged over the dataset. As each training and test set is constructed differently, we note the significance of this metric for each set. Firstly, for Biased MNIST, test set samples are “unbiased" with randomly chosen values for each of the bias variables. As the bias variables are

| | **train** | **test** |
|---|---|---|
| climbing | 326<br>constructed structures or dry rock | 105<br>snowy mountains, outdoors |
| diving | 520<br>natural bodies of water; sea or lake | 159<br>indoor gyms and pools |
| fishing | 163<br>shore of natural bodies of water; sea or lake | 42<br>no water visible; shore background only |
| racing | 336<br>paved and delineated formal racing tracks | 132<br>rough terrain or indoor tracks |
| throwing | 317<br>professional sporting fields | 85<br>urban backgrounds |
| vaulting | 279<br>blue sky | 131<br>spectators or other sport event backgrounds |
| **Total** | 1941 | 654 |

Table 4.4: The BAR dataset: spurious correlations between background and the training set, the test set without those correlations, and the image count of each action class.

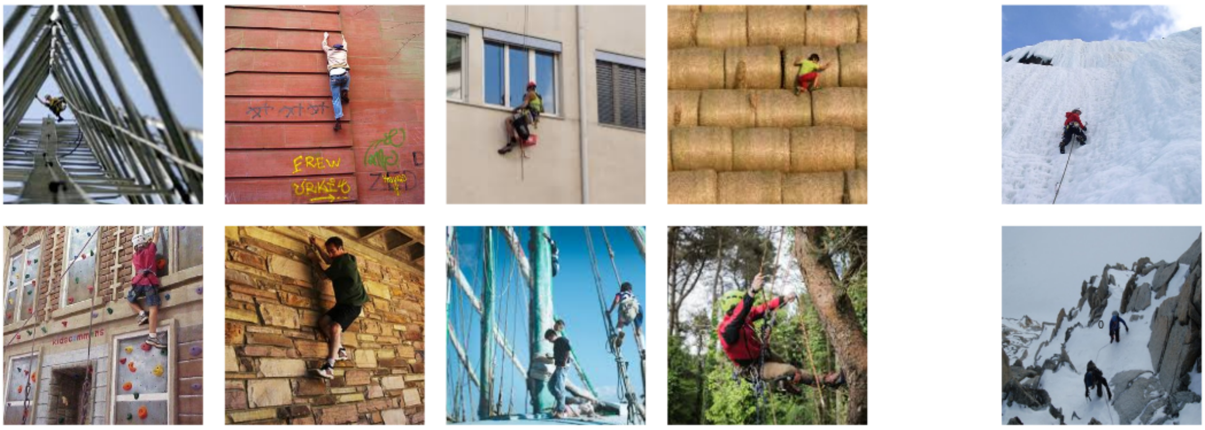


Figure 4.12: BAR training samples from the "climbing" action class (*left*) correlated with structural backgrounds and the test set (*right*) with mostly outdoor snowy mountains.

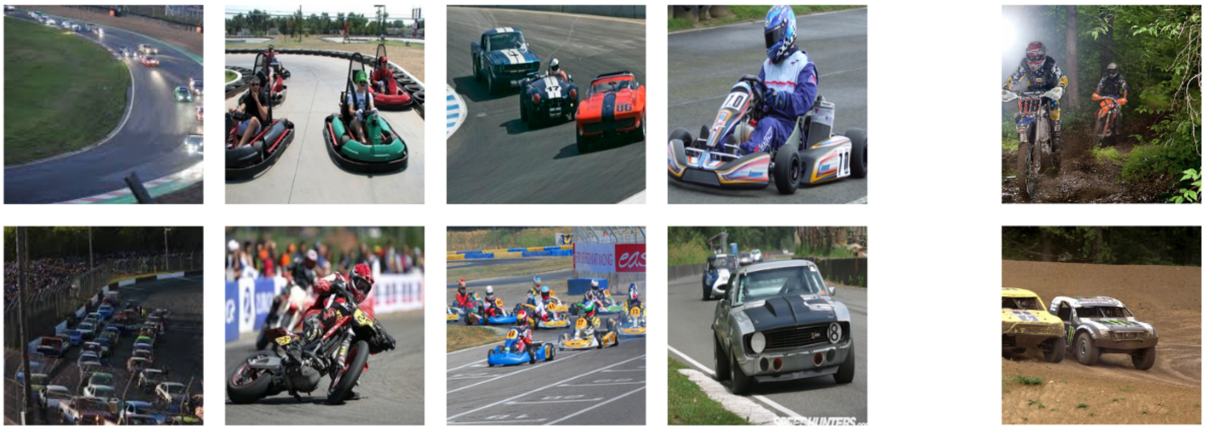


Figure 4.13: BAR training samples from the "racing" action class (*left*) correlated with race tracks and the test set (*right*) with more rugged terrain.

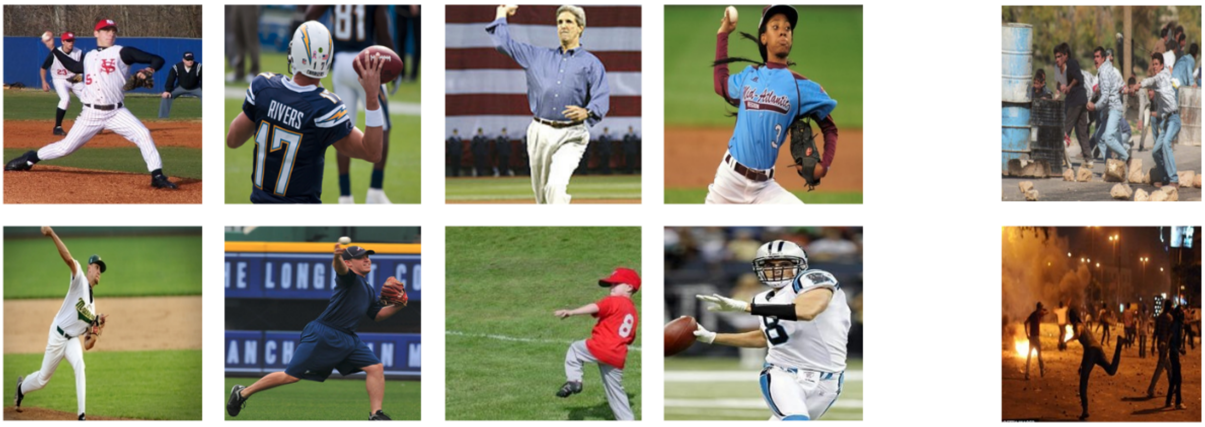


Figure 4.14: BAR training samples from the “throwing” action class (*left*) correlated with organized sporting fields and the test set (*right*) with urban scenes.

co-occurring with $p_{bias} = 0.95$ for each, a random selection in the test set means that very few samples with the same $p_{bias}$ as the training distribution exist. Thus, the test set accuracy is also a measurement of minority subgroup accuracy.

The COCO-on-Places test datasets evaluate performance under specific conditions. The first set is not helpful for a fairness evaluation perspective, as we expect an in-distribution test set to match training set performance. The second set with unseen backgrounds with non-systematic shift – where foreground objects are placed on backgrounds not present in the training set – tests out-of-distribution performance which can be a different problem than the bias problem (we discuss the distinction further in Chapter 5). Only the third test set with systematic shift - foreground objects placed on previously seen backgrounds - constitutes a measurement of minority subgroup accuracy and truly tests for model fairness.

Finally, in BAR, none of the action-context pairings in the test set appear at all in the training set. An equivalent setup as the second COCO-on-Places test set, the BAR test set also is more of an out-of-distribution test than strictly a fairness test; however, in both of these cases, performance on these datasets still evaluates how well the model correlates the core features (belonging to the foreground object) with the target class, as a strong correlation can override any extra features from unseen, out-of-distribution backgrounds.

### 4.3.3 Group predictive uncertainties

Unlike Biased-Synbols where the bias variable and its effect on model uncertainty can be observed and controlled in isolation, our benchmark datasets are significantly more complex. Bias-inducing features co-occur, but not always on the same samples (Biased MNIST). Classes can be severely imbalanced (BAR contains only 163 fishing images compared to its largest class of 520 images). Furthermore, both COCO and BAR exhibit intra-class diversity; the “racing" category includes bikes, motorcycles, runners, and a wide variety of vehicles, and COCO objects occur with different angles and coloring. This diversity also introduces bias into the dataset, albeit unintentional and unaccounted for. The mean group sample uncertainties extracted from

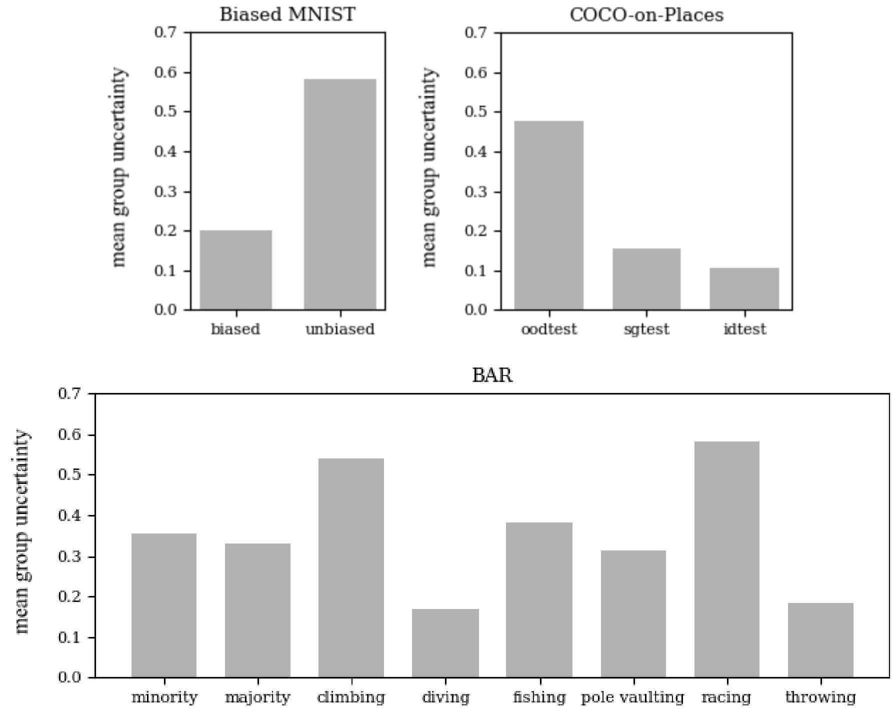


Figure 4.15: Mean uncertainties for subgroups of each test dataset, showing bias-induced discrepancies, but also increased uncertainties arising from a variety of other sources, some of which may be unintentional bias present in the data.

Bayesian ResNet18 models trained on each dataset reflect this as well, as shown in Figure 4.15.

While this entangled experimental setup makes evaluation of the impact of de-biasing methods more difficult, the scenario is certainly realistic given the state of real-world visual datasets. Instead of explicitly disentangling the bias sources, our approaches depend on the posterior estimates adequately expressing their own shortages in information. Thus, class imbalance, intra-class diversity, unintentional biases, and known biases are considered simultaneously.

### 4.3.4 Inference

Optimal parameters for cSG-MCMC are determined via grid search (Table 4.5). We fixed each schedule to 2 cycles and 3 moments sampled per sampling phase. Batch sizes for the baseline Bayesian models were fixed at 128 for training, and 64 for the fine-tuning procedure due to memory constraints (32 for BAR due to larger image size). Lines 2-9 of Algorithm 3 are computed batch-wise and averaged, with one update to parameters per batch in Line 10. Validation accuracy was used to determine stopping points and optimal hyper-parameters for the baseline Bayesian models and training loss sharpening procedure. Training took place on several IBM Power 9 dual-CPU nodes with 4 NVIDIA V100 GPUs (see Acknowledgements for references to Bede and ARC3). We refer readers to the Appendix for the optimal parameters for all experiments.

Following methodology reported in [143], we reduce the kernel size of the first convolutional layer of ResNet18 from 7 to 3 for COCO due to the small image size (64 x 64). We also initialize the

| | lower bound | upper bound | increment size |
|---|---|---|---|
| init step size | 0.01 | 0.5 | $n$*2 |
| cycle length | 150 | 650 | 100 |
| Gaussian noise control parameter $\alpha$ | 0.1 | 0.7 | 0.2 |

Table 4.5: Grid search parameters and value ranges with increment size for the hyper-parameter fine-tuning of the baseline Bayesian models.

priors for the BAR Bayesian ResNet18 models using the weights from a deterministic ResNet18 trained on an ImageNet subset of 100 classes.

The fine-tuning procedure requires keeping a handle on each Monte Carlo posterior sample from $p(\boldsymbol{\theta} \mid D)$, and back-propagating on each of these posterior samples for every iteration. Thus, the procedure has time complexity of $O(M \cdot N \cdot K)$ for $M$ posterior samples, $K$ operations as required for the network architecture, and $N$ iterations, versus $O(N \cdot K)$ for a deterministic network.

## 4.4 Results

### 4.4.1 Bias mitigation evaluation

For a fair comparison against prior reported results on these datasets, we use a ResNet18 throughout the experiments. We compare against several explicit and implicit mitigation methods, separated in each table: cost-sensitive learning (UpWt), Group Distributionally Robust Optimization (gDRO), and predictive group invariance (PGI) for explicit approaches (see Section 2.3.3), and the baseline empirical risk minimization (ERM) of a standard deterministic network, Spectral Decoupling (SD) and OccamNet (Section 2.3.4) for implicit approaches. We also include our approach from Chapter 3 (EpiUpWt). Two of the methods do not perform any de-biasing for comparison (1) ERM for a deterministic ResNet18, and (2) a Bayesian ResNet18, both trained with normal cross entropy loss. As seen in Table 4.6), the fine-tuning procedure gives competitive results on BAR and competitive performance on Biased MNIST compared to all implicit mitigation methods except for one. In contrast, the approach struggles with the COCO-on-Places dataset.

We observe poorer performance on the COCO dataset. Posterior estimate fine-tuning always increases model fairness when compared to its baseline Bayesian starting point. Yet as displayed in Table 4.7, the Bayesian ResNet struggles with both the out-of-distribution and unbiased background test sets, with differences of -3.5% and -2.6% respectively (49.7% versus 53.2%, and 33.0% versus 35.6%) compared to the deterministic ERM baseline. Building on the BayResNet, the sharpening thus starts with a *more unfair model* than methods based on a deterministic ERM model, giving it a distinct disadvantage. Assuming the discrepancy in performance is somewhat due to the stochastic nature of the network, BayResNet+EpiUpWt would also have a disadvantage but does better on the biased test set than BayResNet+$L_{CE}$ and BayResnet+$L_{RCE}$; nonetheless, as is consistent across datasets, the fine-tuning methods still improve on BayRes-

| Architecture+Method | Biased MNIST | COCO | BAR |
|---|---|---|---|
| *Explicit method results* | | | |
| ResNet+UpWt | 37.7 ± 1.6 | 35.2 ± 0.4 | 51.1 ± 1.9 |
| ResNet+gDRO [134] | 19.2 ± 0.9 | 35.3 ±0.1 | 38.7 ±2.2 |
| ResNet+PGI [2] | **48.6** ± 0.7 | **42.7** ± 0.6 | 53.6 ±0.9 |
| *Implicit method results* | | | |
| ResNet+ERM | 36.8 ±0.7 | 35.6 ± 1.0 | 51.3 ±1.9 |
| ResNet+SD [126] | 37.1 ± 1.0 | 35.4 ± 0.5 | 51.3 ±2.3 |
| OccamResNet [143] | **65.0** ±1.0 | **43.4** ± 1.0 | 52.6 ±1.9 |
| BayResNet+EpiUpWt [152] | 34.6 ±1.1 | 34.3 ± 0.8 | 52.1 ±1.5 |
| BayResNet+$L_{CE}$ | 34.5 ±1.7 | 34.8 ± 0.7 | **53.9** ±0.6 |
| BayResNet+$L_{RCE}$ | 38.7 ±0.6 | 34.8 ±0.6 | **54.2** ±0.7 |

Table 4.6: Unbiased test set accuracies comparing BayResNet against current debiasing methods. **First**, **second** and third best results are formatted.

Net+EpiUpWt on the two unbiased test sets.

Regardless, this highlights a weakness of our Bayesian-based approaches. We consider the vast topic of comparing the predictive performances of deterministic and Bayesian neural networks out of the scope of this research, but acknowledge that Bayesian neural networks can struggle to match deterministic benchmarks, which directly affects all Bayesian-based methods including the fine-tuning approach. We weigh these sacrifices in performance against the added benefit of quantified prediction uncertainty estimates, which deterministic models cannot produce.

| Architecture+Method | Biased Bgs | Unseen Bgs | Seen, but Unbiased Bgs |
|---|---|---|---|
| *Explicit method results* | | | |
| ResNet+PGI [2] | 77.5 ±0.6 | 52.8 ±0.7 | 42.7 ±0.6 |
| OccamResNet+PGI [2] | 82.8 ±0.6 | **55.3** ±1.3 | **43.6** ±0.6 |
| *Implicit method results* | | | |
| ResNet+ERM | 84.9 ± 0.5 | 53.2 ±0.7 | 35.6 ±1.0 |
| OccamResNet [143] | 84.0 ±1.0 | **55.8** ±1.2 | **43.4** ±1.0 |
| BayResNet | 84.3 ±0.4 | 49.7 ±1.3 | 33.0 ±0.2 |
| BayResNet+EpiUpWt [152] | **85.8** ±0.1 | 50.3 ±1.1 | 34.3 ±0.8 |
| BayResNet+$L_{CE}$ | 85.6 ±0.6 | 51.3 ±0.7 | 34.8 ±0.7 |
| BayResNet+$L_{RCE}$ | **85.7** ±0.4 | 51.2 ±0.2 | 34.8 ±0.6 |

Table 4.7: Accuracies on each three test splits of COCO-on-Places: biased backgrounds (Bgs), unseen backgrounds, and seen but unbiased backgrounds.

BAR is a counterexample, where the Bayesian baseline outperforms or does at least as well as the deterministic one across class groups. The most realistic and complex dataset of the three, BAR is neither texturally simple like Biased MNIST nor synthetically created and low-resolution like COCO-on-Places. The posterior estimate fine-tuning method performs well on BAR (Table 4.10), especially considering the large discrepancy in performances across classes and methods. This discrepancy is present regardless of method. But notably, while various methods do well in two or three classes, BayResNet+$L_{RCE}$ has good performance across four

| | ResNet+ERM<br>maj/min ($\Delta$) | BayResNet+$L_{CE}$<br>maj/min ($\Delta$) | BayResNet+$L_{RCE}$<br>maj/min ($\Delta$) |
|---|---|---|---|
| Digit Scale | **87.2/31.3 (55.9)** | 84.3/27.7 (56.6) | 89.9/32.2 (57.7) |
| Digit Color | **78.5/32.1 (46.4)** | 76.7/28.4 (48.3) | 82.6/32.9 (49.7) |
| Digit Position | 74.2/26.4(47.8) | **74.5/27.3 (47.2)** | 88.3/35.4 (52.9) |
| Texture | 76.1/32.4 (43.7) | 69.9/29.2 (40.7) | **61.3/35.3 (26.0)** |
| Texture Color | 41.9/36.3 (5.6) | 37.5/32.8 (4.7) | **40.0/37.8 (2.2)** |
| Letter | 46.7/35.7 (11.0) | 42.0/32.3 (9.7) | **44.2/37.3 (6.9)** |
| Letter Color | 45.7/35.9 (9.8) | 45.5/32.1 (13.4) | **45.2/37.2 (8.0)** |
| Test Accuracy | 36.8 | 34.5 | 38.7 |

Table 4.8: Accuracies on majority (maj)/minority (min) groups and bias-induced accuracy gaps for each bias variable in BiasedMNIST ($p_{bias} = 0.95$). Smallest accuracy gaps ($\Delta$) for each bias variable are in **bold**; note that this does not necessarily mean better minority group accuracy.

| Loss | Biased MNIST | COCO | BAR |
|---|---|---|---|
| No sharpening | 32.0 ± 1.2 | 33.0 ± 0.2 | 52.7 ± 2.6 |
| $L_R$ | 34.8 ± 2.1 | 33.3 ± 1.4 | 53.5 ±1.0 |
| $L_{CE}$ | 34.5 ± 1.7 | 34.8 ± 0.7 | 53.9 ± 0.6 |
| $L_{CE} + k \cdot L_R$ | 33.4 ± 0.2 | 32.3 ± 0.3 | 53.2 ± 0.4 |
| PCGrad($L_R, L_{CE}$) | **38.7** ± 0.6 | **34.8** ± 0.6 | **54.2** ± 0.7 |

Table 4.9: Ablation study on fine-tuning procedure loss functions showing performance on the most challenging test sets for each dataset. The performance of the weighted sum of the two losses demonstrates the importance of PCGrad in minimizing destructive interference. Validation accuracy was used for optimization in each case.

out of seven classes (and note that although $L_{CE}$ has a smaller majority/minority gap for the “digit position" bias variable, both the minority and majority group accuracies are higher for $L_{RCE}$). For this dataset, perhaps because of its ability to regularize despite the small dataset size and substantial intra-class diversity, the baseline BayResNet provides a strong starting point for the fine-tuning. For every class, the two proposed mitigation methods improve on the BayResNet baseline results.

Table 4.8 shows the majority/minority group accuracies and bias-induced accuracy gap for the six bias variables in Biased MNIST. Compared to the baseline ERM, the fine-tuned model decreases the accuracy gaps between majority and minority groups across the dataset for 4 out of 6 of the biases (texture, texture color, letter, and letter color). While BayResNet+$L_{RCE}$ results on Biased MNIST are not as competitive, results still demonstrate that the fine-tuning objective aims at increasing model fairness rather than simply rewarding accuracy gains on the majority groups. In addition, uncertainty distributions across training sets do not collapse under $L_{RCE}$ fine-tuning (see subsequent discussion and experiments in Section 4.5), so sample uncertainty estimates extracted at inference time may still be useful for triage or other purposes in high-risk scenarios.

| Methods | Overall | Climbing | Diving | Fishing | Pole Vaulting | Racing | Throwing |
|---|---|---|---|---|---|---|---|
| *Explicit method results* | | | | | | | |
| ResNet+UpWt | 51.1 ±1.9 | 61.7 ±13.2 | **43.9** ±5.8 | 42.3 ±8.3 | 52.3 ±7.4 | 67.9 ±6.7 | 28.2 ±12.8 |
| ResNet+gDRO [134] | 38.7 ±2.2 | 49.5 ±8.5 | 40.3 ±8.4 | 44.0 ±10.4 | 39.9 ±7.1 | 41.7 ±4.0 | 13.5 ±5.9 |
| ResNet+PGI [2] | 53.6 ±0.9 | 61.2 ±10.4 | 38.4 ±4.1 | 42.9 ±8.4 | **73.3** ±3.7 | 68.9 ±5.9 | 23.5 ±1.9 |
| OccamResNet+UpWt | 52.2 ±1.4 | 57.9 ±1.8 | 35.7 ±7.5 | 51.8 ±11.2 | 64.3 ±8.8 | 71.8 ±3.8 | 27.4 ±3.5 |
| OccamResNet+gDRO [134] | 52.9 ±0.8 | 51.2 ±9.6 | 42.8 ±8.2 | 52.3 ±5.1 | 63.5 ±7.3 | 74.2 ±5.2 | 25.3 ±4.5 |
| OccamResNet+PGI [2] | **55.9** ±0.7 | 64.2 ±5.1 | **52.3** ±6.4 | 51.4 ±8.3 | 64.4 ±4.1 | 70.9 ±8.1 | 18.6 ±6.8 |
| *Implicit method results* | | | | | | | |
| ResNet+ERM | 51.3 ±1.9 | 69.5 ±7.5 | 29.2 ±1.8 | 39.9 ±16.2 | 55.5 ±6.4 | 75.6 ±5.6 | 31.8 ±4.3 |
| ResNet+SD [126] | 51.3 ±2.3 | 62.1 ±7.5 | 35.8 ±2.0 | 51.2 ±6.4 | 62.4 ±9.2 | 71.6 ±10.0 | 18.5 ±6.7 |
| OccamResNet [134] | 52.6 ±1.9 | 59.3 ±3.8 | 42.3 ±7.5 | 44.6 ±14.9 | 60.5 ±8.6 | 74.1 ±7.2 | 22.1 ±3.9 |
| OccamResNet+SD [126] | 52.3 ±2.4 | 56.4 ±6.8 | 34.3 ±5.8 | 55.4 ±7.4 | **69.1** ±4.9 | 72.9 ±4.2 | 21.8 ±2.1 |
| BayResNet | 52.7 ±2.6 | **71.1** ±2.7 | 28.3 ±10.3 | 54.8 ±8.2 | 54.2 ±3.3 | 78.0 ±3.9 | 31.8 ±2.6 |
| BayResNet+EpiUpWt [152] | 52.1 ±1.6 | 67.9 ±2.2 | 27.9 ±5.1 | 51.6 ±5.4 | 59.8 ±3.5 | 76.2 ±0.4 | 30.0 ±2.9 |
| BayResNet+$L_{CE}$ | 53.9 ±0.6 | 69.5 ±1.1 | 34.0 ±2.1 | **57.1** ±2.4 | 52.0 ±3.0 | **79.5** ±0.5 | **32.9** ±2.7 |
| BayResNet+$L_{RCE}$ | **54.2** ±0.7 | **72.1** ±2.0 | 32.1 ±3.2 | **59.5** ±2.3 | 52.7 ±2.7 | **79.6** ±0.7 | **32.6** ±2.9 |

Table 4.10: Overall and per-class accuracies on BAR, showing large discrepancies in performance across methods and classes.

### 4.4.2 Ablation studies

#### Objective loss

We consider the impact of $L_{RCE}$ on the multi-task loss in the ablation study shown in Table 4.9. For weighted loss $L_{CE} + k \cdot L_R$, scalar weight $k$ is chosen using a grid search in range $k = 2$ to $k = 10$ with optimal $k$ being in range 3 to 5 for all datasets with negligible impact for $k < 3$ and declining accuracy (biased and unbiased sets) for values $k > 5$; this loss simply adds the two losses together regardless of conflict, with a fixed weighting term across all samples.

#### Biased MNIST validation set

To evaluate the effect of using an unbiased validation set on the hyperparameter selection for the baseline Bayesian model, we generate a second Biased MNIST validation set [1] which contains the same biases as the training set with $p_{bias} = 0.95$ for each of the bias variables, and perform the same grid search on hyperparameters as shown in Table 4.5 using the loss from the new biased validation set. As in previous experiments, we select the set of parameters for which the mean validation loss across the posterior estimates is lowest after the final cycle.

The cSG-MCMC Bayesian formulation is not as dependent on validation loss for selecting hyper-parameters because the cycle lengths are fixed for each experiment; the cycle length determines the learning rate step size and rate of decrease. We find strong evidence of overfitting to the training set irrespective of validation set. In fact, we arrive at the same optimal hyper-parameters as when using the biased validation set, using the validation loss criteria. Comparing the biased validation set experiments from Section 4.4 and experiments that we perform using the generated unbiased validation set, we find that the training accuracy tops 99.9% in both cases. The validation accuracy for the optimal models are 94.7% (biased validation set) and 35.0% (unbiased validation set). This illustrates to what extent the model is learning the training set. As a result, the use of a biased versus unbiased validation set has little impact on our method. We hypothesize that the change in dataset could have a stronger detrimental impact on the other de-biasing methods in published benchmarks.

## 4.5 Hypotheses related to the regularizing loss

Each posterior estimate sample $\boldsymbol{\theta_m}$ which results in an incorrect prediction $(\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x}_i)) \neq \boldsymbol{y}_i)$ has a larger cross-entropy loss than one with a correct prediction. Applying the corresponding gradient to a posterior estimate sample moves it such that the true class likelihood is higher for that sample. The experimental results for Bayesian models with $L_{CE}$ fine-tuning confirm this; the fine-tuned models always perform better in terms of accuracy and fairness compared to those without any fine-tuning. However, we also find that adding in the regularizing loss $L_R$ using PCGrad further improves the performance.

As the effectiveness of $L_R$ is counter-intuitive, particularly when the sample predictive mean

---

[1] Dataset generation script adjusted from: https://github.com/erobic/occam-nets-v1/tree/master/datasets

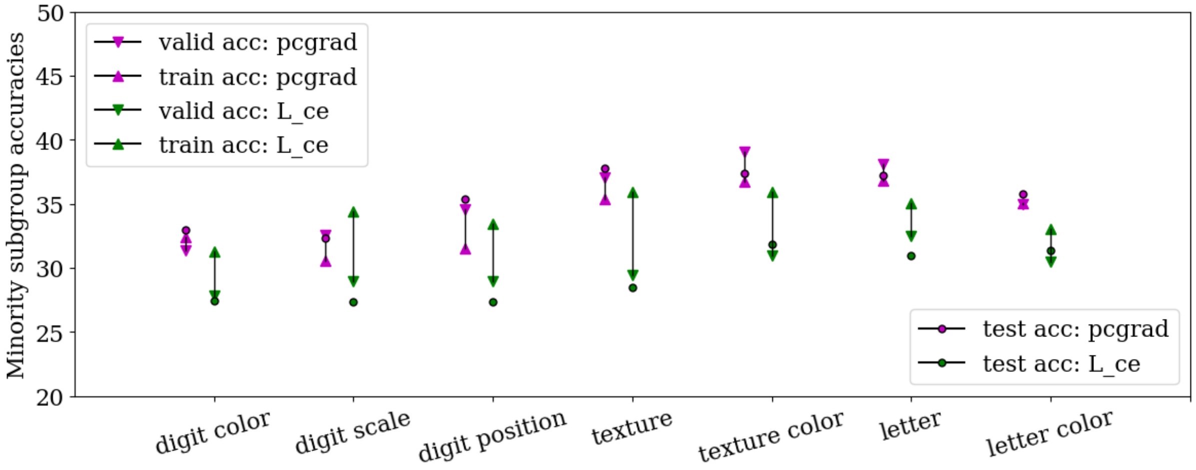


Figure 4.16: This figure shows the mean minority subgroup accuracies of each bias variable in the training, validation, and test sets of Biased MNIST for the Bayesian model fine-tuned with PCGrad($L_{CE}, L_R$) loss *(magenta markers)*, and the same model fine-tuned with $L_{CE}$ loss *(green markers)*. The length of the black lines joining the pair of markers for each bias variable represent the disparity in mean minority subgroup accuracies between training and validation sets. An overfitting model has longer black lines, and a test set accuracy *(circle markers)* further from the train set accuracy. These results suggest that $L_{CE}$ overfits more than PCGrad($L_{CE}, L_R$).

disagrees with the ground truth target, we explore two hypotheses as to why PCGrad($L_{CE}, L_R$) performs better than $L_{CE}$ alone. To test these hypotheses we use Biased MNIST, since the bias variables are relatively more controlled and isolated than in BAR and COCO-on-Places due to its synthetic nature and the restricted color scheme and feature placement. For these experiments, we explicitly access the bias variables and their values on the training, validation, and test sets to evaluate the effect of the two losses. The bias variable labels are used for evaluation only, and do not affect the implicit nature of the bias mitigation.

### 4.5.1 Hypothesis 1: The model over-fits to minority samples

Given that the loss is weighted by input uncertainty, we hypothesize that the second loss $L_R$ could be preventing the model from overfitting to the high-uncertainty samples, in particular due to the PCGrad algorithm. For each mini-batch, one gradient is selected randomly for projection onto the normal of the other. This introduces stochasticity into the process. Furthermore, PCGrad only applies for cases of destructive interference, meaning that if $L_R$ and $L_{CE}$ do not conflict, the gradients are left as is. This ensures that in these cases, $L_R$ does not interfere with $L_{CE}$. PCGrad can only direct the posterior estimate sample gradient away from the ground truth when the prediction is incorrect, the loss gradients destructively interfere, and the $L_{CE}$ gradient is randomly chosen to project onto the normal of the $L_R$ gradient.

We test this hypothesis by comparing the minority subgroup accuracies for fine-tuning the same base model using $L_{CE}$ alone versus PCGrad($L_{CE}, L_R$). A good model is expected to have similar training, validation, and test set accuracies for the minority groups, whereas an overfitting model will have high minority group training accuracy, but perform worse for the minority groups of the

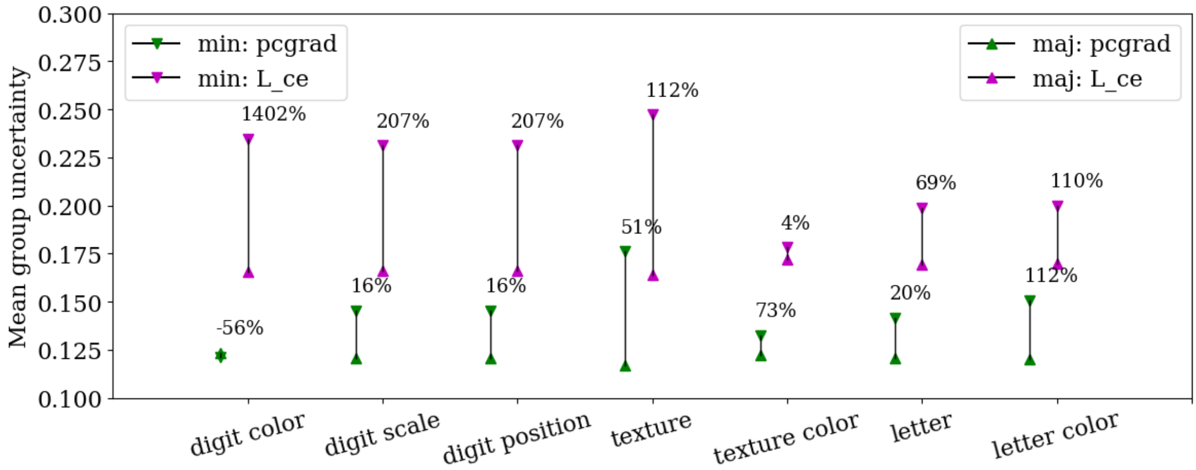


Figure 4.17: This figure shows the mean uncertainties (for minority and majority subgroups of the training set) of each bias variable in Biased MNIST after fine-tuning. The green markers are for the Bayesian model fine-tuned with PCGrad($L_{CE}, L_R$) loss, and the magenta for the same model fine-tuned with $L_{CE}$ loss. The length of the black lines joining the pair of markers for each bias variable represent the disparity in mean uncertainties; and the number above each pair is the percentage difference between the baseline Bayesian model (mean subgroup uncertainties before any fine-tuning, always higher for minority subgroups) compared to after fine-tuning.

validation and test sets. We note that for each experiment run, as in all previous experiments, validation loss over the whole validation set is used to indicate the fine-tuning stopping point. Figure 4.16 shows evidence that the fine-tuned model with $L_{CE}$ alone displays some overfitting to the training set minority samples. In contrast, the model with PCGrad($L_{CE}, L_R$) is a better fitting model.

### 4.5.2 Hypothesis 2: Bias-conflicting sample uncertainties are being driven to zero

As fine-tuning with $L_{CE}$ takes place, the input sample-wise loss is weighted by the sample epistemic uncertainty. High uncertainty input samples result in larger updates to the posterior estimate samples $\boldsymbol{\theta_m}$, sharpening the distribution around the target output. A sharpened distribution has lower variance, or lower sample epistemic uncertainty. We hypothesize that perhaps as fine-tuning progresses, the uncertainties of the minority subgroup samples are being driven to zero, so that over time the model no longer pays attention to them as their loss contributions become minimal due to the weighting function in Equation 4.2. PCGrad($L_{CE}, L_R$) may help to maintain the relative gaps between the minority and majority groups.

To test this hypothesis, we compute the mean epistemic uncertainties for minority and majority subgroups of the training set during the fine-tuning iterations using $L_{CE}$ alone. As expected, the minority group uncertainties for all bias variables begin higher than those for the majority. We then compute the gaps between the two subgroups, and compare it with the gaps when fine-tuning the same base model with PCGrad($L_{CE}, L_R$). Figure 4.17 [2] shows that in general, $L_{CE}$

---

[2] The comparison gaps table for $L_R$ alone are shown in the Appendix for reference.

actually causes the gap between minority and majority subgroup uncertainties to increase more than PCGrad($L_{CE}, L_R$). This observation does not support the hypothesis that using $L_{CE}$ alone causes bias-conflicting sample uncertainties and bias-aligned sample uncertainties to converge. In addition, the gaps for the PCGrad fine-tuned model still increase compared to the baseline Bayesian model for 6 out of 7 bias variables (see positive percentage values in Figure 4.17), indicating that neither loss drives the minority group uncertainties to zero. For both fine-tuning losses, the relative differences between minority and majority sample uncertainties are still preserved.

## 4.6 Discussion and conclusion

The method presented in this chapter allows the Bayesian neural network to freely learn all features, then implicitly adjusts the weighting of those features for posterior estimate samples which give incorrect predictions during a fine-tuning process. The degree of loss update is weighted per input sample by the sample-wise epistemic uncertainty, encouraging a stronger focus on the minority group samples and resulting in a fairer model.

Artificially adjusting the posteriors of Bayesian deep neural networks is not a foreign concept. As presented by Wenzel et al. [171], the true Bayesian posteriors of deep neural networks are rarely used in practice. Rather, *tempered* or *cold posteriors* are widely found to perform better. Tempered posteriors are equivalent to over-counting the data by a factor of $1/T$ for temperature scalar $T$ and re-scaling the prior to $p(\boldsymbol{\theta})^{1/T}$. In contrast, the true Bayes posterior corresponding to $T = 1$ usually gives sub-optimal performance.

We note, however, that our proposed fine-tuning procedure is applied only to the *estimated* posterior derived from MC sampling, not to the true Bayes posterior. The MC samples provide a finite, numerical handle on model inference. Furthermore, running the fine-tuning procedure with a sampling size $M$ of up to 20 indicates that increasing the MC sample size for a more accurate posterior estimate has negligible impact on the performance. Leibig et al. [100] similarly report that a fixed $M < 10$ provides a meaningful approximation of the posterior at low cost.

Our posterior estimate fine-tuning method adds to the body of work on implicit bias mitigation by exploiting the relationship between minority samples and epistemic uncertainties. We have demonstrated competitive performance on the BAR dataset, and similar results as other bias mitigation methods on COCO-on-Places and Biased MNIST. Furthermore, additional experiments show that for the fine-tuning procedure, a regularized loss outperforms a standard cross-entropy loss by itself. Some evidence indicates that the model with cross-entropy loss alone is more prone to over-fitting to the high uncertainty inputs, but the exact reasons and effect of the gradients of high-uncertainty samples on the posterior estimate for other samples is still not certain. Further exploration is also required to better understand when and why Bayesian neural networks struggle with respect to deterministic networks within the context of bias mitigation, and why the performances of mitigation methods vary across datasets. While our selected benchmark datasets are realistically complex, perhaps other datasets with a clearer disentangling of features and correlations within training and test datasets would support additional fairness evaluation, which we leave for future consideration.

# Chapter 5

# Extension to polyp segmentation generalisability

## 5.1 Overview

In previous chapters, we evaluated our bias mitigation methods on benchmark vision datasets for classification tasks. In this chapter, we move to a real-world imaging dataset with complex features. The biases are not well-identified, and while being potentially the most diverse labelled dataset of its kind, it is still small compared to benchmark datasets. All the existing published research indicates that the dataset is particularly challenging. We use this dataset to explore the potential of our methods to improve generalisability across various minority subgroups.

Implicit bias mitigation methods have enormous potential in medical imaging applications, where implications of discriminatory models can be life-threatening and many regions of the world do not have adequate access to trained professionals. A colonoscopy is a common procedure performed to check for polyps, growths which can become cancerous, in the colon. The problem of automated polyp segmentation is extremely relevant for aiding clinicians during endoscopy procedures. However, it is also challenging, due to the diverse nature of the images, and the difficulty of collecting large datasets for training models. We explore the applicability of the novel methods presented in Chapters 3 and 4 to this problem for improved generalisability on underrepresented populations and subgroups. Towards this end, we also extend our methods from classification to image segmentation.

While several previous studies have devised methods for segmentation of polyps, most of these methods are not rigorously assessed on multi-center datasets. Variability due to appearance of polyps from one center to another, difference in endoscopic instrument grades, and acquisition quality, result in methods with good performance on in-distribution test data but poor performance on out-of-distribution or underrepresented samples. Discriminatory models have serious implications, including misdiagnoses and perpetuating social and racial inequalities for underrepresented populations. As these populations tend to be already disadvantaged in terms of availability of medical care, these issues pose a critical challenge to the clinical adaptation of intelligent models.

We demonstrate the potential of our two Bayesian uncertainty-aware mitigation approaches to improve generalisability without sacrificing state-of-the-art performance on a challenging multi-center polyp segmentation dataset (PolypGen) with different centers and image modalities. Part of this chapter's contents are published in [151].

## 5.2 Background

### 5.2.1 Clinical motivation and context

Colorectal cancer (CRC) is the third most common cancer worldwide and the first cause of death in developed countries [49], with early screening and removal of precancerous lesions (colorectal adenomas, growths known as "polyps") resulting in longer survival rates. While not all polyps lead to cancer, all colorectal cancers begin with the growth of polyps that eventually become malignant. Once present, cancerous polyps can multiply and spread, making their early accurate detection and analysis a key factor for combating CRC.

Colonoscopy is the common procedure by which a flexible fibre-optic tube is used to examine the colon, to look for and/or remove polyps. While surgical removal of polyps (polypectomy) is a standard procedure during colonoscopy, detecting the polyps and their precise delineation from different angles is challenging, especially for sessile serrated polyps. Serrated polyps are named due to their saw-toothed appearance under a microscope, and vary in size, structure, and presentation (see Figure 5.1) with a reported miss rate by humans of 27% [182]. Over the past decade, advanced computer-aided methods and most recently machine learning methods have been explored by various researchers. However, the adaptation of these technologies into clinical settings has still not been fully achieved. One of the main reasons is issues with generalisability [6], as most existing techniques are built and adapted using carefully curated datasets - where polyps are clearly visible in a clean, non-obstructed environment. This is not representative of the majority of video footage taken during a colonoscopy.

Furthermore, polyp colonoscopy datasets are difficult to collect, and curated datasets are usually small and subject to sampling and population biases. Some databases for endoscopy images [118, 150] lack pixel-level ground truth annotations. Others [115, 5] including EndoAtlas [1] have adequate annotations, but include only a limited number of images from at most a few collection centers. These issues contribute to the difficulty in studying and evaluating the generalisability problem for intelligent models.

Recent literature demonstrates how intelligent models can be systematically unfair and biased against certain subgroups of populations. In medical imaging, the problem is prevalent across various image modalities and target tasks; for example, models trained for lung disease prediction [139], retinal diagnosis [24], cardiac MR segmentation [127], and skin lesion detection [1,

---

[1] http://www.endoatlas.org/

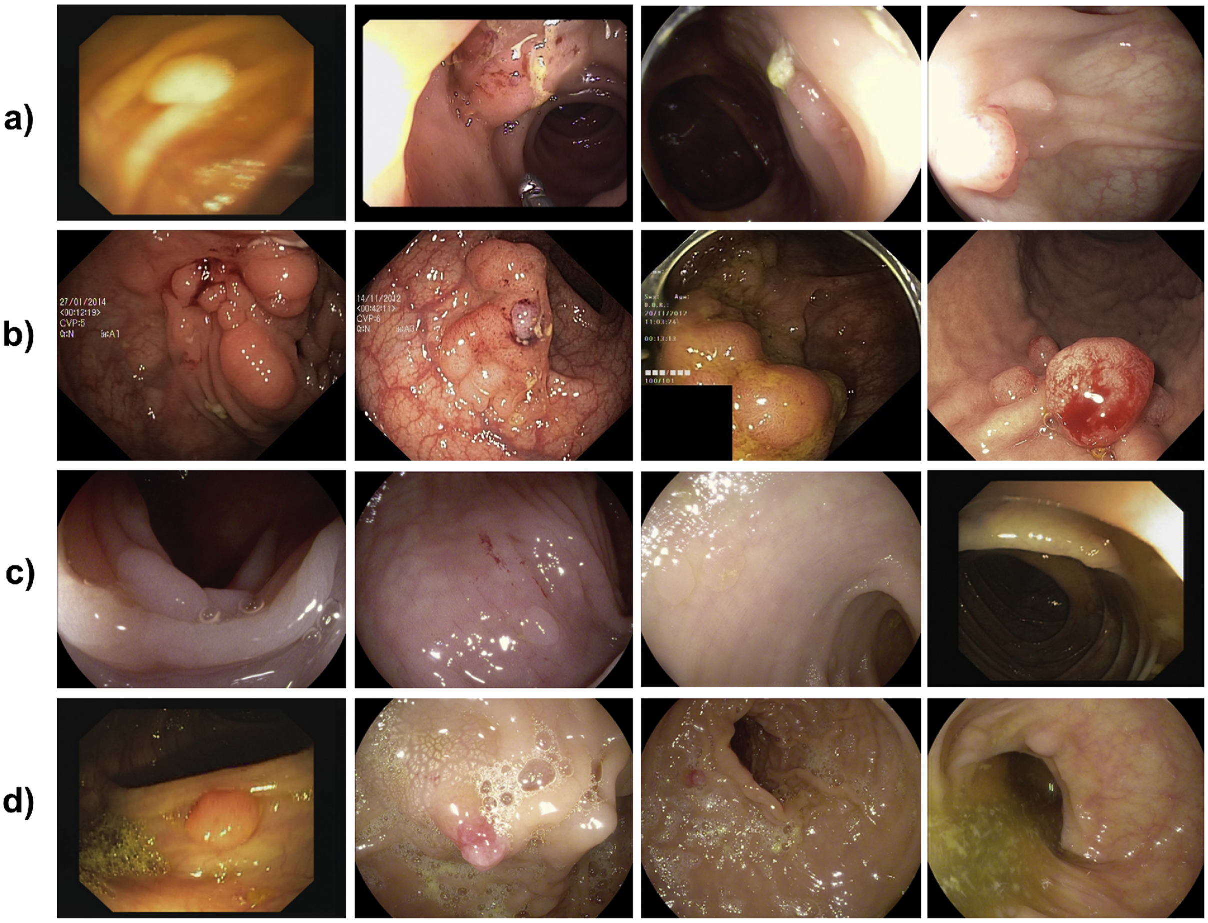


Figure 5.1: Sample polyps from [110] showing some of the challenging conditions under which polyp segmentation during a colonoscopy must occur: (a) low quality or low resolution images with varying lighting, (b) varying shapes and textures of polyps, (c) only subtle differences between the polyps and background, and (d) presence of irrelevant artifacts.

102] are all subject to biased performance against one or a combination of underrepresented gender, age, socio-economic, and ethnic subgroups. In colonoscopy specifically, variations in polyp morphology have been observed for varying ages [175], geography, race, and ethnicity [104, 28, 79]. This, and the scarcity of literature exploring bias mitigation for polyp segmentation in particular, strongly motivate the need for development and evaluation of mitigation methods which work on diverse, realistic colonoscopy datasets such as PolypGen [4] (Section 5.4.1).

Convolutional neural networks have become crucial in data-driven approaches to polyp segmentation using deep learning. In particular, many methods in recent literature are adaptations of the encoder-decoder U-Net [131] architecture, introduced for cell segmentation. Mahmud et al. [110] identify several limitations of the U-Net when applied to polyp segmentation such as semantic information lost in the typical skip connections, and propose PolypSegNet with modified skip layers and new modules designed for aggregating various scales of feature representations from all the encoder levels. Yeung et al. [176] propose Focus U-Net, a U-Net variant with gated attention to encourage selective learning of polyp features over the background and a composite loss for dealing with the spatial imbalance.

Others directly address the problem of differing polyp sizes by using off-the-shelf or modified state-of-the-art multi-scale feature pruning methods from the vision literature, such as atrous-spatial pyramid pooling in DeepLabV3 [30] or high-resolution feature fusion networks like HR-Net [146]. Similarly, MSRFNet [148] uses feature fusion networks between different resolution stages to preserve important features at each level.

### 5.2.2 Generalisability challenges

Studies [79, 6] have found that methods trained on data from specific medical centers do not generalise well on unseen center data or changes in modality. Colonoscopy data is collected as sequence data, or frames sampled from video. Datasets typically include individual frames of polyps, whereas in a clinical setting, intelligent models should be able to process sequence data. Ali et al. [6] show large performance gaps (drops in accuracy of nearly 20%) when models trained on single frames are tested on sequence data.

Generally speaking, out-of-distribution (OOD) generalisation and bias mitigation are often considered as separate problems in the literature, although they result in similar undesired outcomes. While in the bias problem formulation, models wrongly correlate one or more spurious (non-core) features with the target task, the out-of-distribution problem states that test data is drawn from a separate distribution than the training data. Some degree of overlap between the two distributions in the latter formulation exists, which in both formulations should include the core features, the features directly associated with the target task. While the bias problem is associated with fairness and ethical issues, the OOD problem is associated with model applicability and performance. Regardless of the perspective, the two problems have clear similarities, and whether labeled as “discriminatory" or “poorly performing", both result in models which struggle to generalise for certain populations.

In the OOD literature, many works focus on OOD detection, through normal or modified softmax outputs [69], sample uncertainty thresholds from Bayesian, ensemble, or other models [113, 82, 27], and distance measures in feature latent space [60]. Other approaches tackle the more difficult problem of algorithmic mitigation through disentangled representation learning, architectural and learning methods, and methods which optimise for OOD generalisability directly [140].

We consider the generalisability problem for polyp segmentation and specifically, the challenges posed by out-of-distribution samples. Samples from different geographical regions and samples of the more challenging sequence modality are both found in PolypGen, a recent multi-centre polyp dataset [4]. In this chapter, we: 1) adapt the implicit bias mitigation strategies in Chapter 3 and 4 from classification to a segmentation tasks, and 2) evaluate the suitability of these approaches on PolypGen with three separate test sets which have been shown to be challenging generalisation problems.

Our experiments demonstrate that our methods are comparable and in many cases improve the performance compared to the baseline state-of-the-art segmentation method while simultaneously decreasing performance discrepancies between different test splits.

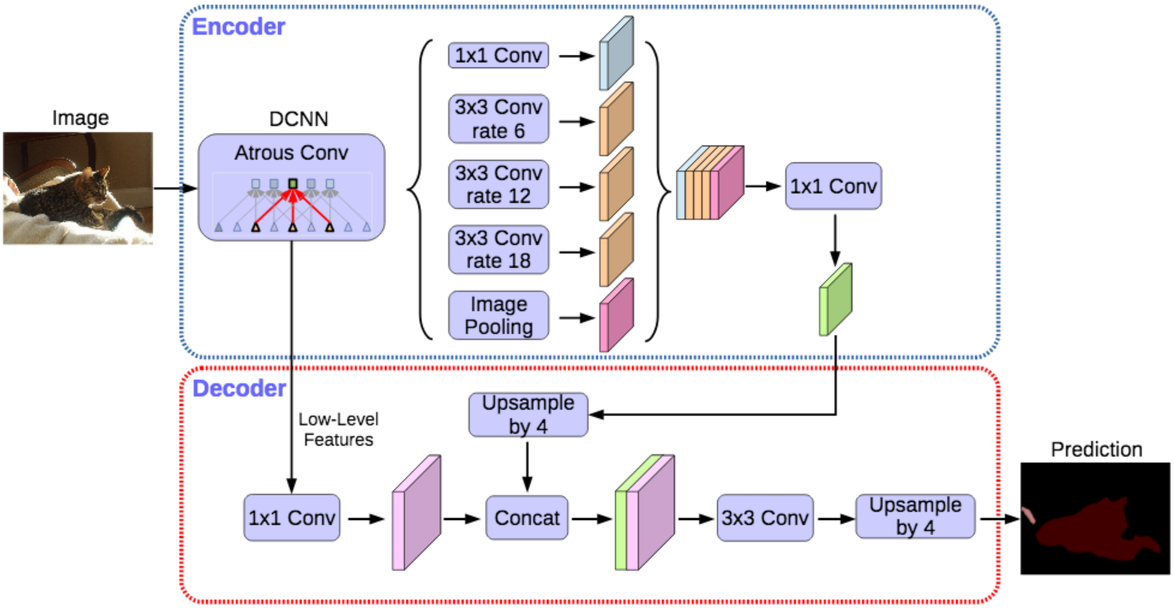


Figure 5.2: DeepLabV3+ from [32, 31], the encoder-decoder architecture with atrous convolutions at different rates (filter step-sizes) for capturing multi-scale contextual information and more effective boundary delineation.

## 5.3 Method

### 5.3.1 Semantic segmentation framework

Semantic segmentation is the computer vision problem whereby each pixel of an image is classified into a class. For polyp segmentation, this results in a dense pixel-wise binary segmentation map where each pixel is either classified as background (non-polyp) or foreground (polyp). In our approach we use DeepLabV3+ [31] as a baseline model as it is reported to give state-of-the-art performance on the PolypGen dataset [4].

DeepLabV3 [32] (Figure 5.2) employs dilated or atrous convolutions that widen the field-of-view of each convolutional filter. An atrous filter operates on the input with stride or gaps, such that the field of view is larger than the filter size. The feature maps from the atrous convolutions with various dilation rates are then combined via a form of image pooling. This design allows for better capture of multi-scale features. DeepLabV3+ improves on this architecture by combining the atrous convolutions and spatial pooling from DeepLabV3 with an encoder-decoder architecture. The encoder-decoder architecture for semantic segmentation has been widely explored and shown to be effective in medical image analysis. DeepLabV3+ takes the original DeepLabV3 architecture to create an encoding from the input, then decodes this with shortcut connections for more precise object delineation and spatial understanding at multiple scales.

Ali et al. [4] present the performance of multiple choices of encoder for DeepLabV3+ including ResNet50 and ResNet101, and MobileNet combined with DeepLabV3 and DeepLabV3+ variants. DeepLabV3+ with a ResNet50 encoder architecture results in the best-performing results for general accuracy on the published challenge test set. Thus, we build our Bayesian implementation from this network.

### 5.3.2 Bayesian DeepLabV3+

We apply a probabilistic model assuming a Gaussian prior on all trainable weights (both encoder and decoder) that are updated to the posterior using the training dataset. For the Bayesian network with parameters $\boldsymbol{\theta}$, training data with ground truth segmentation masks $D = (X, Y)$, posterior $p(\boldsymbol{\theta} \mid D)$, and input image $\boldsymbol{x_i}$, the predictive posterior distribution can be written as:

$$p(\boldsymbol{y_i} \mid D, \boldsymbol{x_i}) = \int p(\boldsymbol{y_i} \mid \boldsymbol{\theta}, \boldsymbol{x_i}) p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta} \tag{5.1}$$

As in previous chapters, we choose to approximate the posterior using stochastic gradient Monte-Carlo sampling MCMC (SG-MCMC [170]) with the cyclical learning rate schedule introduced in [179]. Stochastic gradient over mini-batches includes a noise term approximating the gradient over the whole training distribution.

The DeepLabV3+ encoder backbone computes the features for the model, and the decoder classifier component returns a binary mask. The typical learning rates for these components with Stochastic Gradient Descent (SGD) are between 0.1 and 0.001, with the backbone learning rate a factor of 10 smaller than that of the classifier. For the Bayesian formulation, we fix the starting (maximum) learning rate for the cyclical schedule with the same proportions; for example, an experiment with initial backbone learning rate $\mathrm{lr}_b = 0.001$ has initial classifier learning rate $\mathrm{lr}_c = 0.01$. Thus, for a single batch using cross entropy loss ($L_{CE}$), the computation includes the cumulative losses of both encoder and decoder and the loss noise term from the cSG-MCMC approximation.

The final estimated posterior of the Bayesian network, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ...\boldsymbol{\theta}_M\}$, consists of $M$ moments sampled from the posterior taken during the sampling phases of each learning cycle. With functional model $\boldsymbol{\Phi}$ representing the neural network, the approximate predictive mean $\boldsymbol{\mu_i}$ for one sample $\boldsymbol{x_i}$ is:

$$\boldsymbol{\mu_i} \approx \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\Phi}_{\theta_m}(\boldsymbol{x_i}) \tag{5.2}$$

We can derive a segmentation prediction mask $\boldsymbol{\hat{y}_i}$ from $\boldsymbol{\mu_i}$ by taking the maximum output between the foreground and background classes at each spatial location. The predictive uncertainty mask corresponding to this prediction (Equation 5.3) represents the model uncertainty for the predicted segmentation mask, the pixel-wise standard deviation of the predictive distribution for that sample.

$$\boldsymbol{\sigma_i} \approx \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\boldsymbol{\Phi}_{\boldsymbol{\theta_m}}(\boldsymbol{x_i}) - \boldsymbol{\mu_i}\right)^2} \tag{5.3}$$

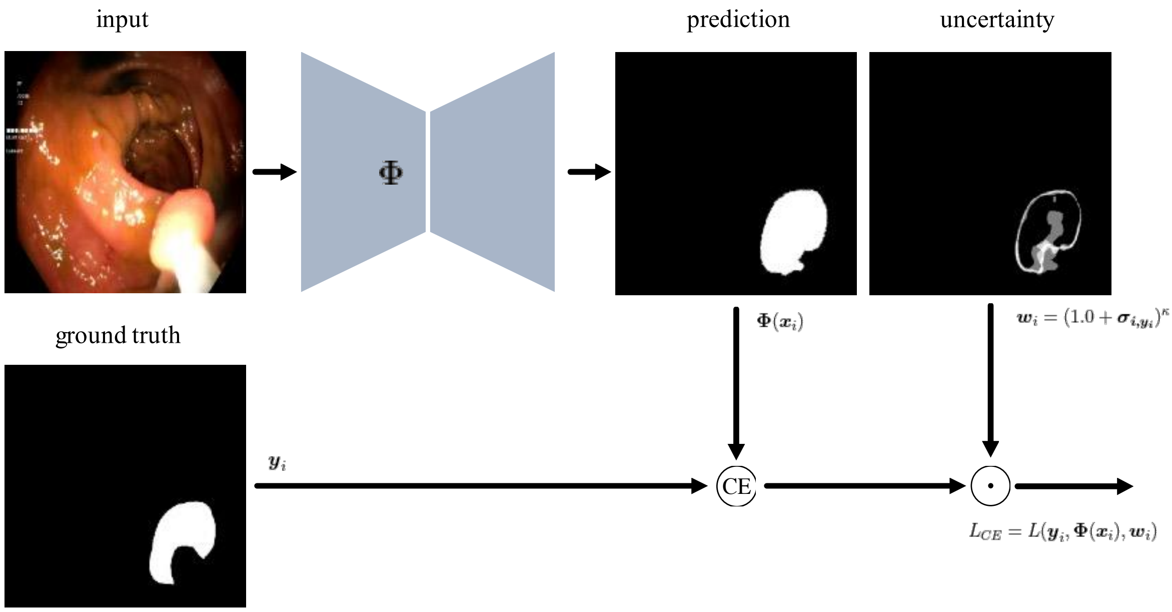

input
prediction
uncertainty
$\Phi$
ground truth
$\Phi(\boldsymbol{x}_i)$
$\boldsymbol{w}_i = (1.0 + \boldsymbol{\sigma_{i,y_i}})^{\kappa}$
$\boldsymbol{y}_i$
CE
$\odot$
$L_{CE} = L(\boldsymbol{y}_i, \boldsymbol{\Phi}(\boldsymbol{x}_i), \boldsymbol{w}_i)$

Figure 5.3: Pixel-wise weighting of cross entropy (CE) loss contribution based on predictive uncertainty maps for each training sample; the model is encouraged to focus on regions for which it is more uncertain.

### 5.3.3 EpiUpWt: uncertainty-weighted cross entropy loss for segmentation

We add the uncertainty-weighted sample loss from Chapter 3, modified for our segmentation problem (Figure 5.3). Consider input $\boldsymbol{x}_i$ with dimensions (512, 512, 3). Instead of a scalar uncertainty value per input, and a scalar weighting value (recall $\sigma_i$ and $w_i$ in Equation 3.13 and 3.14), we consider the uncertainty map $\boldsymbol{\sigma}_i$ whose spatial dimensions are equal to those of the input image with a channel each for foreground and background (512, 512, 2). $\boldsymbol{\sigma_{i,y_i}}$ refers to the uncertainties for the ground truth class for each pixel, an element-wise indexing of $\boldsymbol{\sigma}_i$ by $\boldsymbol{y}_i$. The dimensions of the weighting map are thus (512, 512).

$$\boldsymbol{w}_i = (1.0 + \boldsymbol{\sigma_{i,y_i}})^{\kappa} \tag{5.4}$$

$$L_{CE} = L(\boldsymbol{y}_i, \boldsymbol{\Phi}(\boldsymbol{x}_i), \boldsymbol{w}_i) = -\frac{1}{512 * 512} \sum_{X,Y} \left( \boldsymbol{w_i} \cdot \sum_{C} \boldsymbol{y}_i \cdot \log(\boldsymbol{\Phi}(\boldsymbol{x}_i)) \right) \tag{5.5}$$

In Equation 5.4, $\boldsymbol{\sigma_{i,y_i}}$ contains the predictive uncertainties of the ground truth class for each pixel in the input image. The sum and exponent operations are computed pixel-wise. In Equation 5.5, $\boldsymbol{y}_i$ is the one-hot encoded ground truth map with dimensions (512, 512, 2). The inner summation sums the predicted likelihood of the ground truth classes ($C$) for each pixel, foreground and background, while the outer summation operates over the whole image domain ($X, Y$ being the set of all possible pixel values) to produce one single loss value per input.

Note that Stochastic Gradient Descent is also used in the original deterministic DeepLabV3+, with reduction over the whole image. Validation metrics were used to determine the optimal $\kappa$ value.

### 5.3.4 $L_{RCE}$: Fine-tuned posteriors for segmentation

We then experiment with the fine-tuned posterior estimates from Chapter 4. The loss for each posterior estimate is computed as in Equation 5.5, with $\boldsymbol{\Phi}_{\boldsymbol{\theta}_m}$ replacing $\boldsymbol{\Phi}$ as in Equations 4.3 and 4.4. We consider $L_{RCE}$ instead of $L_{CE}$ for the fine-tuning due to its superior performance in previous experiments, and fine-tune on top of the BayDeepLabV3+EpiUpWt model. Due to the memory requirements of DeepLabV3+ with ResNet50 backbone (38.8M trainable parameters), we used several strategies to make the fine-tuning more efficient. Firstly, taking the BayDeepLabV3+EpiUpWt posterior estimates, we iterated through each estimate and saved the mean prediction maps and uncertainty maps for each of the training images. These could be loaded from disk for the fine-tuning phase. Secondly, for each forward and backward pass through the posterior estimates, the most up-to-date version of each moment's weights is saved to (and subsequently loaded from) disk.

We found that decreasing the number of moments used for the posterior estimate to $M = 3$ has a negligible effect on performance when posterior samples are deliberately chosen at the largest possible intervals of the sampling phase as shown in Figure 5.4.

## 5.4 Experiments

### 5.4.1 Datasets

PolypGen [4] is an expert-curated polyp segmentation dataset comprising of both single frames and sequence frames (frames sampled at every 10 frames from video) from over 300 unique patients across six different medical centers (Figure 5.5). Each of the 3762 annotated polyps are delineated and confirmed by six senior gastroenterologists. The acquisition, ethical approval, and patient consent for the data was handled by each medical center separately, and relevant information can be found in Table 2 of [4].

The natural data collection format is video from which single frames and sequence data are hand-selected. The single frames are clearer, better quality, and with polyps in each frame including polyps of various sizes (10k to 40k pixels), and also potentially containing additional artifacts such as light reflections, blue dye, partial view of instruments, and anatomies such as colon linings and mucosa covered with stool, and air bubbles (Fig. 5.6). In contrast, the sequence frames are more challenging and contain more negative samples (without any polyp present) and more severe artifacts, which often occur in colonoscopy. Our training set includes 1449 single frames from five centers (C1 to C5) and we evaluate on the three tests sets used

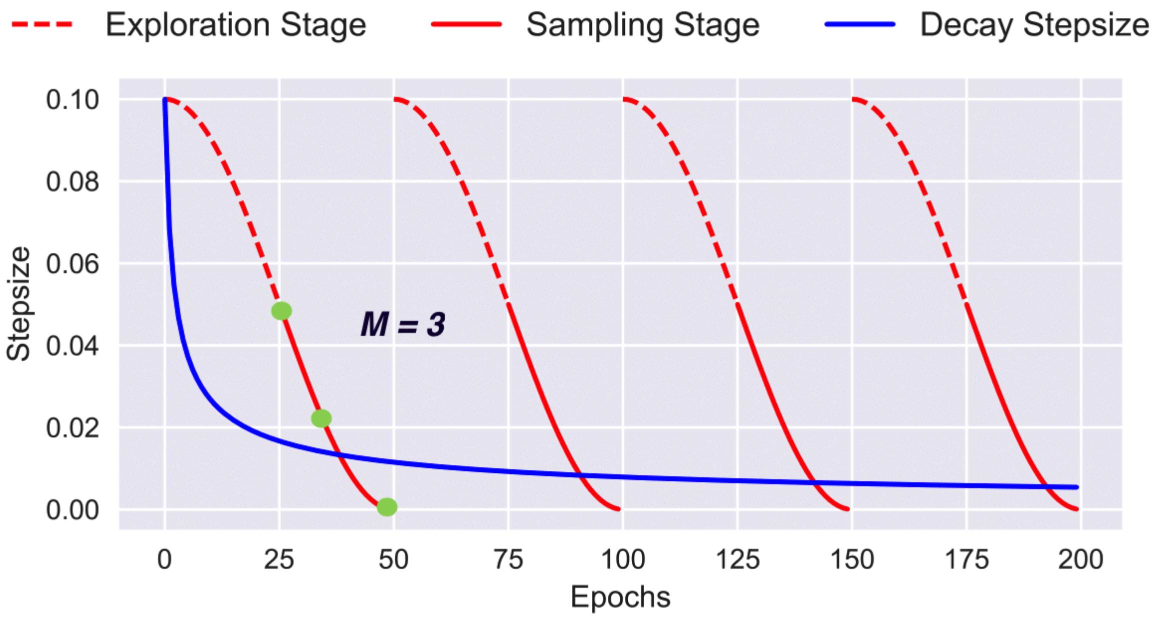


Figure 5.4: cSG-MCMC with a choice of smaller $M$ for comparable performance and uncertainty estimates and lower resource requirement: selecting moments with maximized intervals during the sampling phase. Note that using this sampling size under the regular criteria would mean sampling in the last three epochs of the sampling phase.

| | dataset | image count | modality | center(s) |
|---|---|---|---|---|
| training | C1-5-SIN | 1449 | single | C1-5 |
| testing | C6-SIN | 88 | single | C6 |
| | C1-5-SEQ | 124 | sequence | C1-5 |
| | C6-SEQ | 432 | sequence | C6 |

Table 5.1: The PolypGen training and three test set image counts, modalities, and collection centres.

for generalisability assessment in the literature [6, 4] (Figure 5.6). We create a validation set, a randomly selected 10% split of the training data, but with non-overlapping patients.

The first of the three test datasets (Table 5.1) has 88 single frames from an unseen center C6 (C6-SIN), and the second has 124 frames from seen centers C1-5; however, these are more challenging as they contain both positive and negative samples with different levels of corruption that are not as present in the curated single frame training set. Here, the first test dataset (C6-SIN) comprises of hand selected images from the colonoscopy videos while the second test dataset (C6-SEQ) contains a sequence of images obtained by sampling one from every $10^{th}$ frame of video, which represents most closely the actual colonoscopy data. The third test dataset includes 432 frames from sequence data also from unseen center C6 (C6-SEQ). As no C6 samples nor sequence data are present in the training data, these test sets present a challenging generalisability problem. [2].

---

[2] C1-5-SEQ and C6-SEQ data are referred to as DATA3 and DATA4, respectively, in [6]

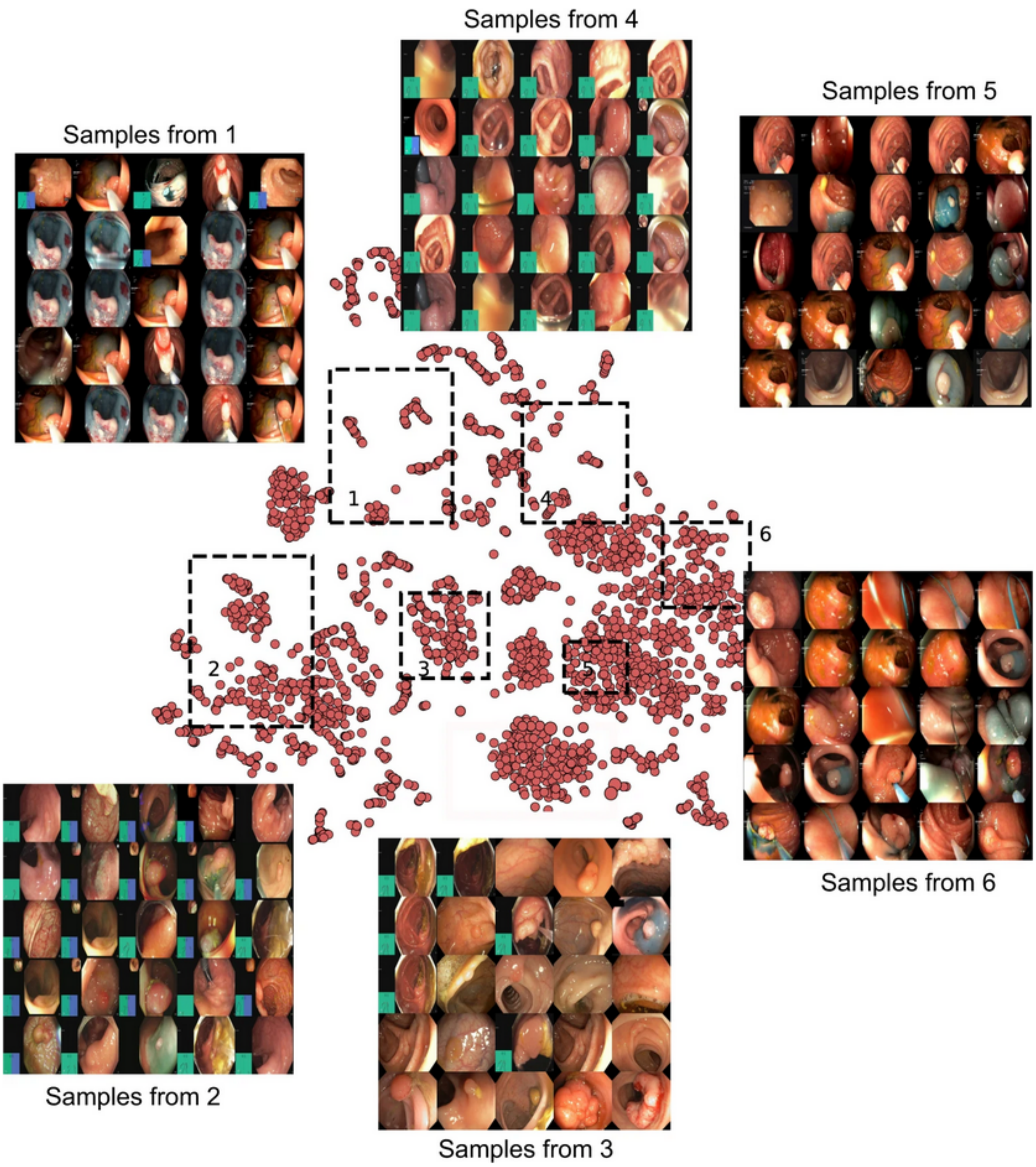


Figure 5.5: The 2D t-SNE embedding of the PolypGen training set as shown in [6] from deep autoencoder extracted features: for each of the six centers (C1 to C6) from which data was collected, 25 random images are displayed in a grid as exemplars, showing the diversity of features across the centers with positive (polyp) and negative (no polyp present) samples, presence of dyes, different endoscopy locations, and source modality (sequence vs. single frame).

### 5.4.2 Inference

Training was carried out on several IBM Power 9 dual-CPU nodes with 4 NVIDIA V100 GPUs. Validation metrics were used to determine optimal models for all experiments with hyperparameters chosen via grid search.

Perhaps due to some frames containing very large polyps with high uncertainties, we found that the gradients of Bayesian models with uncertainty-weighted loss (BayDeepLabV3+EpiUpWt)

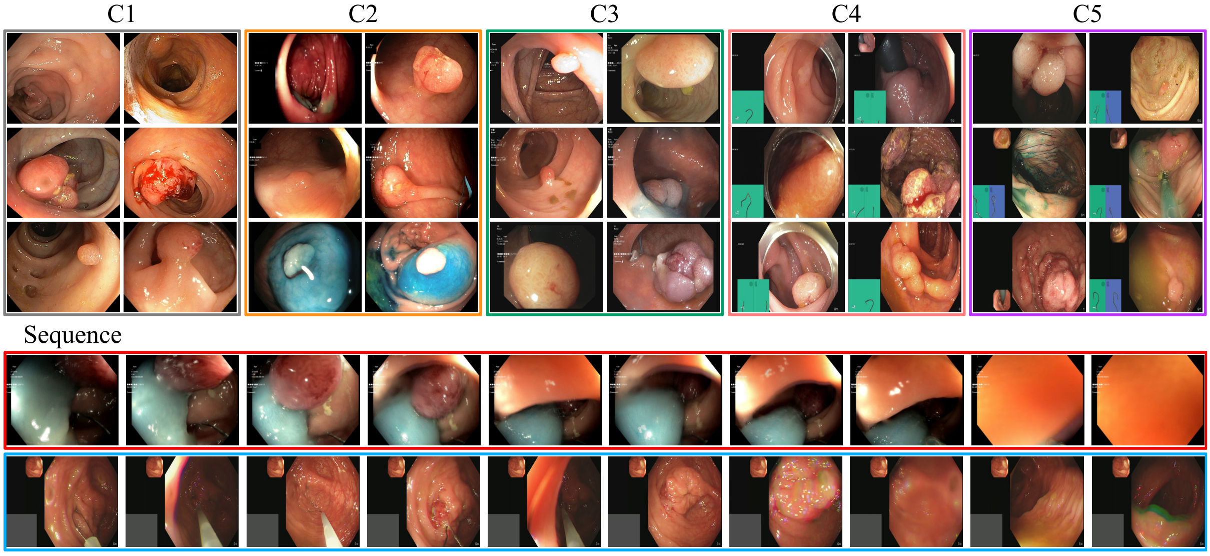


Figure 5.6: Samples from the PolypGen dataset; from (*top*) C1-5 single frames and (*bottom*) C1-5-SEQ; (*top*) highlights the data distribution of each center (C1-C5), which consists of curated frames with well-defined polyps; (*bottom*) demonstrates the variability of sequential data due to the presence of artifacts, occlusions, and polyps with different morphology.

occasionally exploded during the second learning cycle. Clipping the absolute gradients at 1.0 for all weights prevented this issue. The fine-tuning method did not have this problem, perhaps due to each posterior estimate receiving updates separately. All final Bayesian DeepLabV3+ (BayDeepLabV3+) models had 2 cycles, a cycle length of 550 epochs, noise control parameter $\alpha = 0.9$, and an initial learning rate of 0.1, parameters which were determined via grid search. For BayDeepLabV3+EpiUpWt, we found optimal results with de-biasing tuning parameter $\kappa = 3$. Posterior estimates for BayDeepLabV3+ and BayDeepLabV3+EpiUpWt included 6 and 4 samples per cycle, respectively. For BayDeepLabV3+EpiUpWt+$L_{RCE}$, optimal results were found with an initial learning rate of 0.01.

## 5.5 Results

### 5.5.1 Predictive uncertainties

The predictive uncertainty masks have the same dimensions as the prediction masks. Examples of some masks from DeepLabV3+EpiUpWt, alongside the original frame, ground truth mask, and predicted segmentations, are shown in Figure 5.7 and 5.10, with further examples in the Appendix. They provide insight into the challenging parts of the laparoscopy scene, and can be of use to users in clinical settings.

### 5.5.2 Generalisability evaluation

We evaluate the current state-of-the-art deterministic model, DeepLabV3+ with ResNet50 encoder using publically available checkpoints [3] on the three test sets, and compare against:

---

[3] https://github.com/sharib-vision/PolypGen-Benchmark, last accessed July 2024

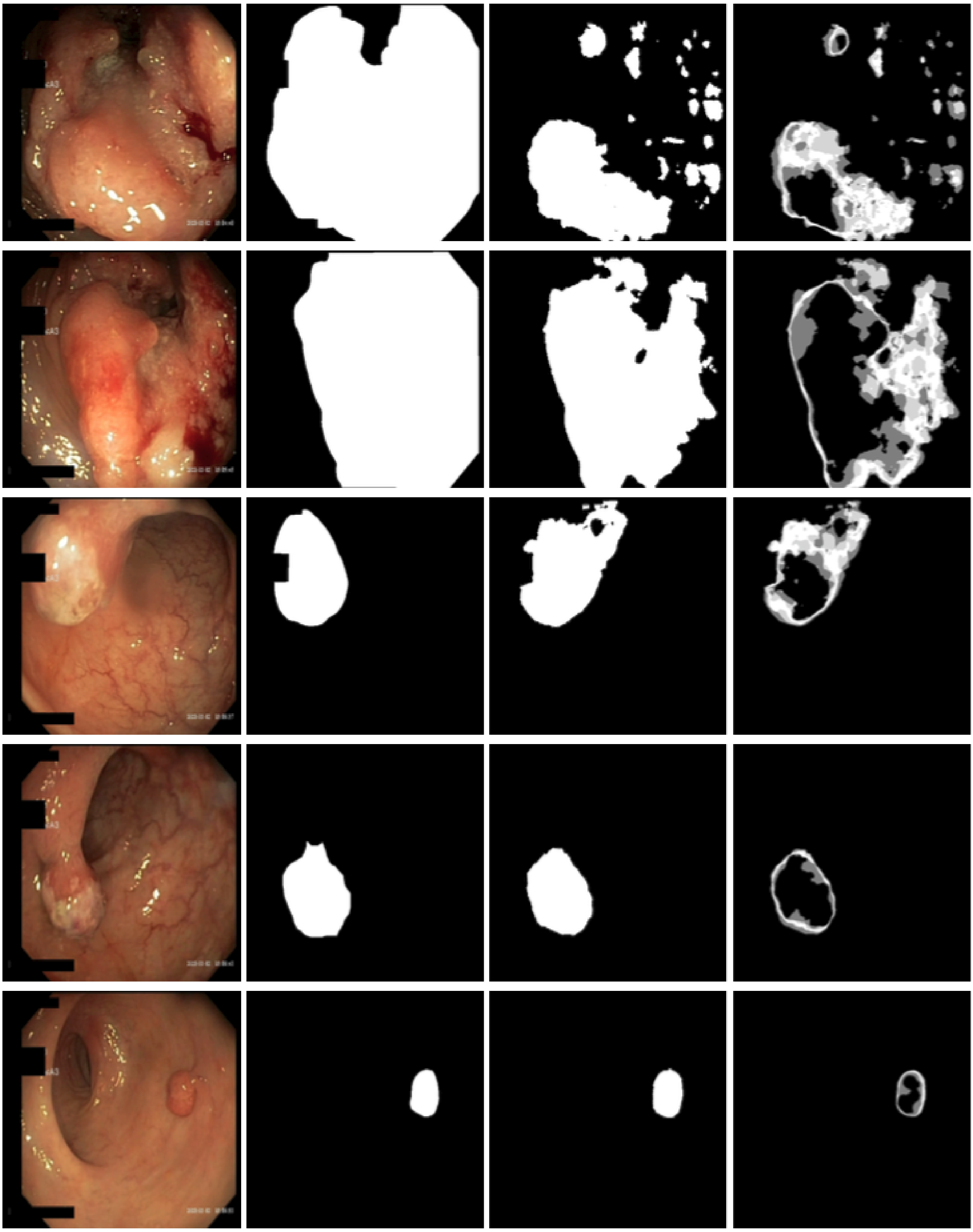


Figure 5.7: Five frames from Center 6 sequence data, showing the original frame (*far left*), the ground truth segmentation mask (*middle left*), the prediction of the Bayesian DeepLabV3+ (*middle right*), and the corresponding predictive uncertainty mask after the first cycle (*far right*). For the first two rows showing particularly large polyps, the Bayesian model shows high uncertainty for the areas with unusual appearance such as the recessed portions and poorly lit sections.

the baseline Bayesian model BayDeepLabV3+, BayDeepLabV3+EpiUpWt with uncertainty-weighted loss, and BayDeepLabV3+$L_{RCE}$ with a fine-tuned posterior estimate. We report

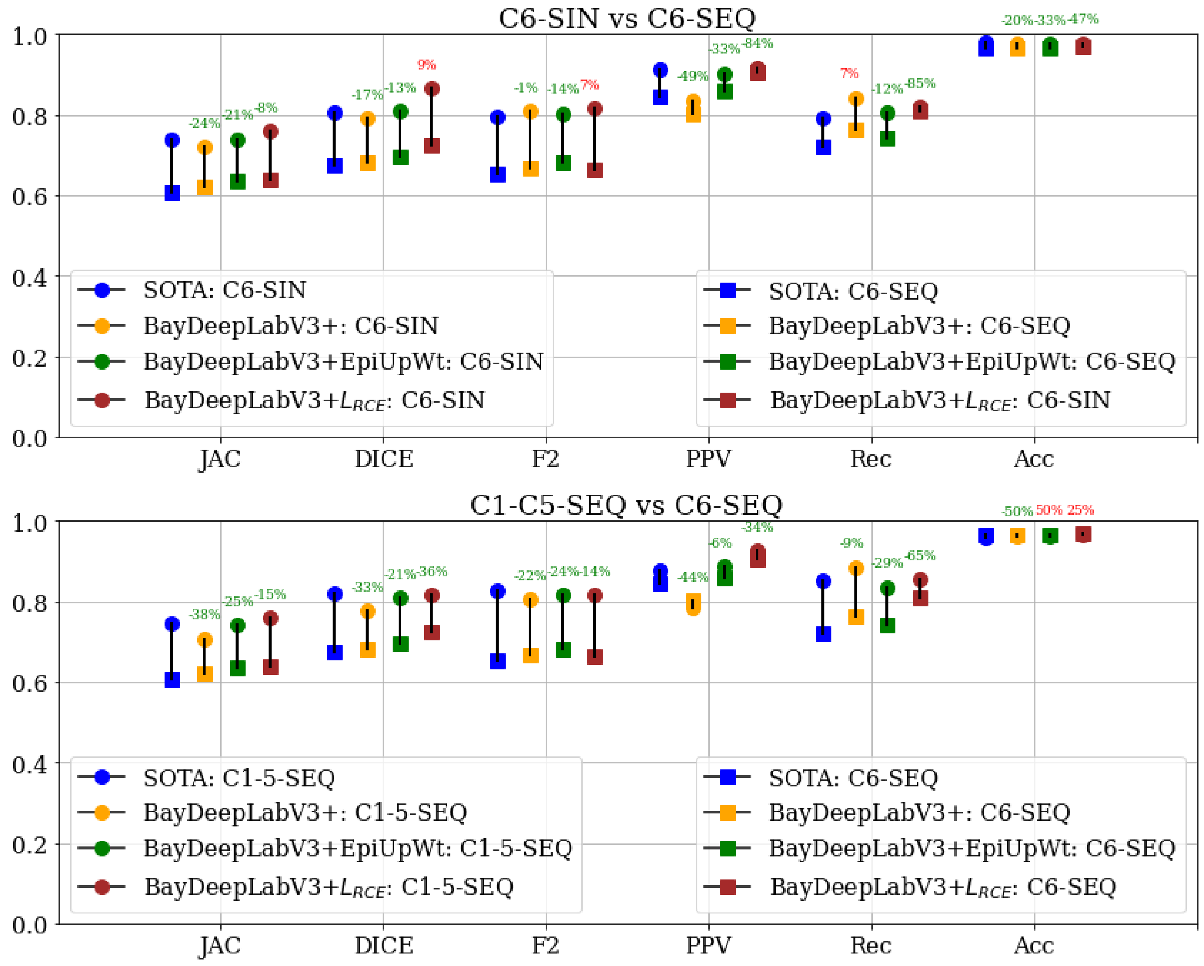


Figure 5.8: Performance gaps of the three models (state-of-the-art deterministic DeepLabV3+, BayDeepLabV3+, BayDeepLabV3+EpiUpWt, and BayDeepLabV3+EpiUpWt+$L_{RCE}$ - referred to as BayDeepLabV3+$L_{RCE}$ on the figure) between the three different test sets; *(top)* comparing performance on single vs. sequence frames from out-of-distribution test set C6 (C6-SIN vs. C6-SEQ), and *(bottom)* sequence frames from C1 - C5 vs. unseen C6 (C1-5-SEQ vs. C6-SEQ). The subtext above bars indicates the percent decrease in performance gap compared to SOTA; a larger percent decrease and shorter vertical bar length indicate better generalisability.

results for the following metrics:

- **Jaccard index (JAC or IoU)** measures the intersection over union of the two segmentation maps.
- **Dice coefficient (F1 Score)** is the harmonic mean of precision and recall, an equal weighting of precision and recall. For two segmentation maps, this is equivalent to twice the intersection divided by the union.
- **F$_\beta$-measure with $\beta = 2$ (F2)** is the weighted harmonic mean of the precision and recall, with a higher weighting on recall.
- **Positive predictive value (PPV)** is the precision, the ratio of true positives over all

| **Dataset** | **Method** | **JAC** | **Dice** | **F2** | **PPV** | **Recall** | **Accuracy** |
|---|---|---|---|---|---|---|---|
| C6-SIN | SOTA | 0.738±0.3 | 0.806±0.3 | 0.795±0.3 | 0.912±0.2 | 0.793±0.3 | **0.979±0.1** |
| | BayDeepLabV3+ | 0.721±0.3 | 0.790±0.3 | 0.809±0.3 | 0.836±0.2 | **0.843±0.3** | 0.977±0.1 |
| | BayDeepLabV3+EpiUpWt | 0.740±0.3 | 0.810±0.3 | 0.804±0.3 | 0.903±0.1 | 0.806±0.3 | 0.977±0.1 |
| | BayDeepLabV3+EpiUpWt+$L_{RCE}$ | **0.759±0.4** | **0.867±0.2** | **0.817±0.3** | **0.917±0.3** | 0.820±0.3 | 0.978±0.1 |
| C1-5-SEQ | SOTA | 0.747±0.3 | **0.819±0.3** | **0.828±0.3** | 0.877±0.2 | 0.852±0.3 | 0.960±0.0 |
| | BayDeepLabV3+ | 0.708±0.3 | 0.778±0.3 | 0.805±0.3 | 0.784±0.3 | **0.885±0.2** | 0.963±0.0 |
| | BayDeepLabV3+EpiUpWt | 0.741±0.3 | 0.810±0.3 | 0.815±0.3 | 0.888±0.2 | 0.836±0.3 | 0.961±0.0 |
| | BayDeepLabV3+EpiUpWt+$L_{RCE}$ | **0.758±0.3** | 0.817±0.2 | 0.815±0.4 | **0.927±0.2** | 0.856±0.3 | **0.965±0.1** |
| C6-SEQ | SOTA | 0.608±0.4 | 0.676±0.4 | 0.653±0.4 | 0.845±0.3 | 0.719±0.3 | 0.964±0.1 |
| | BayDeepLabV3+ | 0.622±0.4 | 0.682±0.4 | 0.669±0.4 | 0.802±0.3 | 0.764±0.3 | 0.965±0.1 |
| | BayDeepLabV3+EpiUpWt | 0.637±0.4 | 0.697±0.4 | **0.682±0.4** | 0.858±0.3 | 0.741±0.3 | 0.967±0.1 |
| | BayDeepLabV3+EpiUpWt+$L_{RCE}$ | **0.640±0.2** | **0.725±0.2** | 0.665±0.3 | **0.906±0.2** | **0.809±0.3** | **0.970±0.1** |

Table 5.2: Evaluation of the state-of-the-art deterministic DeepLabV3+, BayDeepLabV3+, and our proposed BayDeepLabV3+EpiUpWt, and BayDeepLabV3+EpiUpWt+$L_{RCE}$, showing mean and standard deviations across the respective test dataset samples. **First** and second best results for each metric per dataset are formatted.

positive predictions. PPV in particular has high clinical value as it indicates a more accurate delineation for the detected polyps.

In addition, we report recall (Rec) and mean pixel-wise accuracy (Acc). Mean accuracy is less indicative of successful polyp segmentation since the majority of each frame is the non-polyp background class. Recall is meaningful from a clinical perspective, as too many false positives can cause inconvenience to endoscopists during a colonoscopy and hence can hinder clinical adoption of methods.

Figure 5.8 illustrates that the BayDeepLabV3+EpiUpWt matches SOTA performance across most metrics and various test settings, even outperforming in some cases; simultaneously, the performance gaps between different test sets representing challenging features (1) image modalities (single vs. sequence frames) and (2) source centers (C1 - C5 vs. C6) are decreased. Simply using the Bayesian version of a deterministic model improves the model's ability to generalise, yet comes with a sacrifice in performance across metrics and datasets. Our proposed uncertainty-weighted loss achieves better generalisability without sacrificing performance (also see Table 5.2).

We note performance superiority to SOTA especially on C6-SEQ, with an approximately 3% improvement on Dice. We can also observe slight improvement on PPV for test sets with sequence (both held-out data and unseen centre data). Finally, we note that in clinical applications, the uncertainty maps for samples during inference could be useful for drawing clinicians' attention towards potentially challenging cases, increasing the likelihood of a fairer outcome.

The fine-tuning method, BayDeepLabV3+EpiUpWt+$L_{RCE}$ metrics are competitive, particularly for PPV across all test datasets, but do not decrease performance gaps between test datasets as much as the EpiUpWt model alone. We hypothesize by observing the rapid decrease in training and validation loss patterns during the fine-tuning procedure that there may be some overfitting to the training data.

### 5.5.3 Bias mitigation evaluation

The three available test sets allow for generalisability assessment, but strictly speaking, are not bias scenarios since data from Center 6 and sequence modality images are completely absent from the training set. While we can assume that the core features which describe both classes (background and polyp foreground) must be present across all centers, both C6-SIN and C6-SEQ could be better classed as out-of-distribution test sets.

As the two bias mitigation methods proposed rely on minority group samples of high uncertainties being present in the training data, we design a second set of experiments to model this bias scenario. In these experiments, we add samples from Center 6 into the training set, removing them from the test set. C6-SEQ introduces two factors which are out-of-distribution: both (1) the collection center and (2) the image modality (sequence) are previously unseen. C1-5-SEQ only introduces one unseen factor, the image modality, since the center data is included in the training set. Similarly, C6-SIN only introduces one unseen factor, as the modality (single frame) is previously seen but the collection center is not. For the bias mitigation experiments, we add a small number of samples from each of the three test datasets, separately, into the training set. For all experiments, the maximum number of frames (116) is removed from each test set, so that the performances for models across all cuts is evaluated on the same test sets. While ideally we would evaluate on the same test sets as the generalisability experiments, due to having decreased the size of the test set, these results must be considered independently.

| percentage of cut | 1.25% | 2.5% | 5.0% | 10% |
|---|---|---|---|---|
| minority / majority ratio | 1:80 | 1:40 | 1:20 | 1:10 |
| image count | 15 | 29 | 59 | 116 |

Table 5.3: The approximate image count and minority/majority ratios for the minority subgroups added into the training set for the bias mitigation experiments.

Modeling the problem in this way makes the out-of-distribution test set a minority subgroup in the training set. We create the subgroup sizes shown in Table 5.3. In some cases, the number of images is rounded up or down by one to include a complete sequence. For each case, the minority subgroup samples are selected such that no frames from the same patient are present in both the test and training sets. For sequence data, entire sequences are selected such that the test set includes no frames from any sequences seen during training.

Figure 5.9 shows firstly that adding minority group data to the training set for the SOTA deterministic DeepLabV3+ model gives inconsistent results; we found that adding data did not always improve generalisability. The BayDeepLabV3+EpiUpWt model exhibits the clearest correlation between additional training data and minority group test performance. For each test dataset, increasing the number of minority group samples in the training set improved performance. BayDeepLabV3+EpiUpWt+$L_{RCE}$, however, only shows this correlation for the C6-SEQ dataset; performance changes very little for C6-SIN and is inconsistent for C1-5-SEQ.

Since it is generally understood that adding diversity to the training set improves performance on similarly diverse test sets, we hypothesize that in these experiments with limited data, the choice of samples added to the training set is non-trivial. We expect more stable correlations across all methods, including the SOTA model without any bias mitigation, given a larger number of samples in both training and test sets. All results in tabular form, evaluated across all three datasets, are provided in the Appendix.

## 5.6 Discussion and conclusion

We have motivated the critical problem of model fairness in polyp segmentation on a multi-center dataset, and modified two Bayesian bias mitigation methods to the task. The results on three challenging test sets show potential for improving generalisability while maintaining competitive performance across all metrics. Furthermore, the proposed mitigation method is implicit, not requiring comprehensive knowledge of biases or out-of-distribution features in the training data. This is of particular importance in the medical community given the sensitivity and privacy

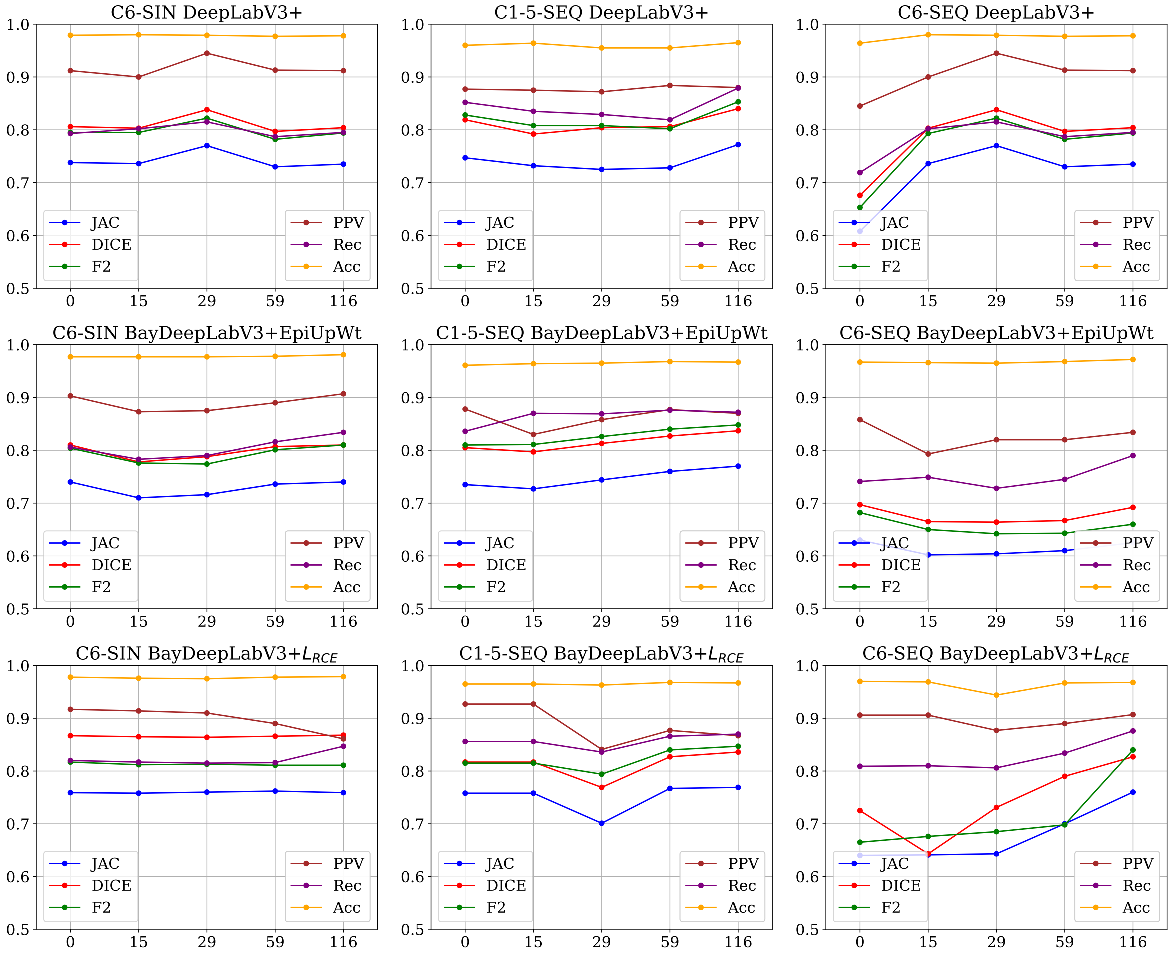


Figure 5.9: Baseline SOTA model DeepLabV3 (*top*), BayDeepLabV3+EpiUpWt (*middle*), and BayDeepLabV3+EpiUpWt+$L_{RCE}$ (*bottom*) results for the bias mitigation experiments evaluated on the three test datasets: C6-SIN (*left*), C1-5-SEQ (*center*), and C6-SEQ (*right*). For each figure, the x-axis shows the number of minority group frames added to the training set. The first row shows inconsistent results when adding minority group samples to the training set for the deterministic SOTA model. Surprisingly, adding more data does not guarantee better test performance. BayDeepLabV3+EpiUpWt shows a clearer correlation in the second row; adding more minority group samples improves performance across all metrics and test datasets. BayDeepLabv3+EpiUpWt+$L_{RCE}$ in the bottom row lacks a clear correlation for C6-SIN and C1-5-SEQ, but does perform better with more minority data for C6-SEQ. (See Appendix for tabular results, and figures for the baseline BayBayDeepLabV3+.)

issues limiting collection of annotations and metadata. Our findings are highly relevant to the understudied problem of generalisation across high variability colonoscopy images.

When framing the problem as a bias rather than an out-of-distribution problem by placing minority group samples in the training set, we find that uncertainty up-weighting during training (EpiUpWt) provides more reliable performance. Metrics improve with the quantity of minority group samples. Results are less clear for the baseline model and the posterior estimate fine-tuned ($+L_{RCE}$) model.

Future work including comparisons with other generalisability, bias mitigation, and domain shift methods, and further experimentation with usage of the background class uncertainty for our mitigation methods. We anticipate that access to additional training and test data will greatly facilitate further experiments and in-depth analysis of the results.

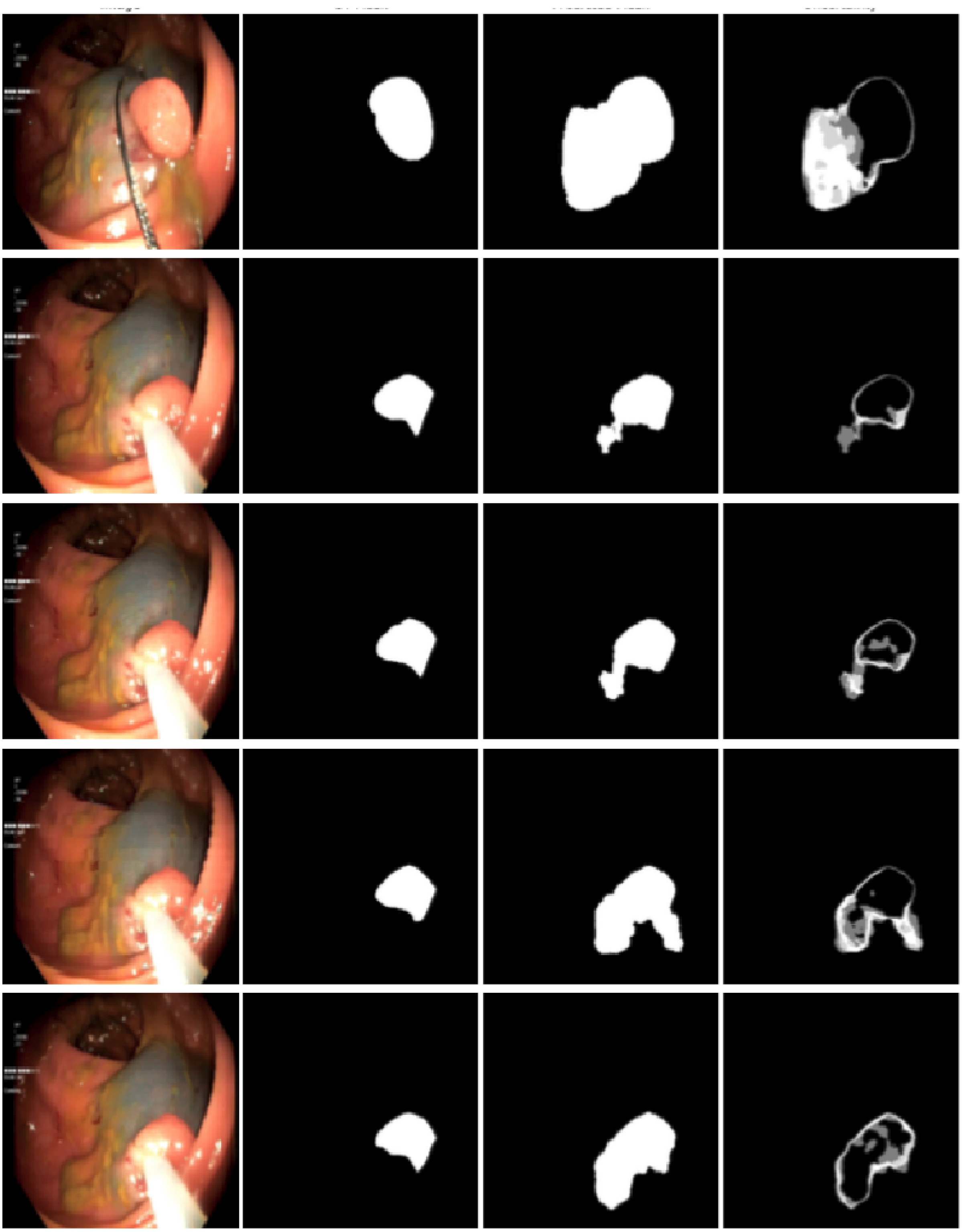


Figure 5.10: Five frames from Center 1-5 sequence data, showing the original frame (*far left*), the ground truth segmentation mask (*middle left*), the prediction of the Bayesian DeepLabV3+ (*middle right*), and the corresponding epistemic uncertainty mask after the first cycle (*far right*). Note that in these samples, the region occluded by the resection tool in the foreground is seen as uncertain by the Bayesian model, as well as the border regions of the segmentation. EpiUpWt draws attention to these portions during training.

# Chapter 6

# Conclusion

With billions of internet users, zettabytes of data being generated every year, and powerful data-driven intelligent models released open-source to the public, artificial intelligence is at the forefront of society today. Thanks to various high-profile cases, the public along with the research community has become aware of the biases and flaws in such systems. These biases reflect real biases hidden in the data, and in turn, hidden in society.

The research community has focused on understanding why and when biases are learned by models, and how to mitigate them when the sources of bias are known and unknown. Yet the problem still remains open, with most mitigation approaches relying on a good understanding of bias sources for both mitigation and fairness evaluation.

In this research, we propose two implicit bias mitigation methods for vision applications. While deep neural networks in literature benchmarks are mostly deterministic, we present the Bayesian neural network as a viable uncertainty-aware alternative. The predictive uncertainties are shown to be correlated to bias-conflicting samples, as seen in various literature, and confirmed by our experiments. As this correlation exists irrespective of bias source, we leverage these uncertainties to encourage the model to pay more attention to underrepresented instances in the input space dynamically during training (Chapter 3) and post-training as a fine-tuning procedure (Chapter 4).

In summary, our contributions include:

- an implicit bias mitigation method which uses the dynamic predictive uncertainty estimates of training samples from a Bayesian neural network to perform loss weighting during training (Chapter 3);
- a post-training fine-tuning procedure which adjusts the posterior estimates of a trained Bayesian neural network, also weighting update step-size by predictive uncertainty estimates (Chapter 3); and
- an application of both these methods to a diverse polyp segmentation dataset with known generalisability issues.

While these approaches show potential on established bias benchmark datasets, and on the challenging polyp segmentation problem, the scope of this research is limited to exploring Bayesian uncertainty-based implicit mitigation methods. Within this scope, we do not explore – theoretically or in practice – why Bayesian neural networks can, in practice, struggle to match the performance of deterministic neural networks. Since our methods rely on Bayesian neural networks for baseline performance, this can put them at a disadvantage compared to other mitigation methods.

Predictive uncertainties as we have approximated them can be further disentangled into aleatoric and epistemic uncertainties. Assuming these uncertainties are sufficiently disentangled in application, they could be leveraged separately for bias mitigation. It is possible that isolating epistemic uncertainties, and leveraging them alone, could produce fairer outcomes for both our methods. This is a line of future work which could extend upon the work in this thesis.

Furthermore, more fundamental questions remain. It is not well understood why some bias mitigation methods work better for some datasets than for others, and under what conditions specific methods are expected to do well. This is also true of our proposed approaches. For Bayesian predictive uncertainties in particular, there are also open questions related to relative uncertainties among minority samples. While we know that the minority samples are more likely to have higher uncertainties than the majority samples, it is not known how leveraging these uncertainties benefits one minority subgroup compared to another. The methods generally make the model more fair for all minority subgroups, but this work does not consider how relative fairness among the minority subgroups is affected. Another complex question, how multiple bias variables affect the uncertainties compared to a single bias variable, is also under-explored. Studying this relationship would require a careful choice of dataset, balancing both complexity and interpretability.

A deeper understanding of how to analyse datasets and their implicit biases would provide a stronger foundation upon which to predict which bias mitigation methods might work best in different scenarios, and also to guide development of new bias mitigation methods.

Despite these vast future avenues of work, our research has shown that the uncertainty estimates of Bayesian neural network can be useful for implicit bias mitigation. Our two methods consistently improve results compared to Bayesian baselines without any mitigation, and often perform competitively to other mitigation methods, both implicit and explicit. Furthermore, they show potential even in a real-world segmentation task where data is diverse and low in quantity. In conclusion, this research offers a contribution to the domain of bias mitigation, but is far from a final solution to a challenging problem.

# References

[1] Samaneh Abbasi-Sureshjani et al. “Risk of training diagnostic algorithms on data with demographic bias”. In: *Interpretable and Annotation-Efficient Learning for Medical Image Computing: Third International Workshop, iMIMIC 2020, Second International Workshop, MIL3ID 2020, and 5th International Workshop, LABELS 2020*. Springer. 2020, pp. 183–192.

[2] Faruk Ahmed et al. “Systematic generalisation with group invariant predictions”. In: *International Conference on Learning Representations*. 2021.

[3] Junaid Ali, Preethi Lahoti, and Krishna P Gummadi. “Accounting for Model Uncertainty in Algorithmic Discrimination”. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 336–345.

[4] Sharib Ali et al. “A multi-centre polyp detection and segmentation dataset for generalisability assessment”. In: *Scientific Data* 10.1 (2023), p. 75.

[5] Sharib Ali et al. “Deep learning for detection and segmentation of artefact and disease instances in gastrointestinal endoscopy”. In: *Medical Image Analysis* 70 (2021), p. 102002.

[6] Sharib Ali et al. “Assessing generalisability of deep learning-based polyp detection and segmentation methods through a computer vision challenge”. In: *arXiv preprint arXiv: 2202.12031* (2022).

[7] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. “Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings”. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[8] Alexander Amini et al. “Uncovering and mitigating algorithmic bias through learned latent structure”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 289–295.

[9] Julia Angwin et al. “Machine bias”. In: *Ethics of data and analytics*. Auerbach Publications, 2022, pp. 254–264.

[10] Martin Arjovsky et al. "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893* (2019).

[11] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. "Real-time convolutional neural networks for emotion and gender classification". In: *arXiv preprint arXiv:1710.07557* (2017).

[12] Saeid Asgari et al. "Masktune: Mitigating spurious correlations by forcing to explore". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23284–23296.

[13] Yuri Sousa Aurelio et al. "Learning from imbalanced data sets with weighted cross-entropy function". In: *Neural Processing Letters* 50 (2019), pp. 1937–1949.

[14] Hyojin Bahng et al. "Learning de-biased representations with biased representations". In: *International Conference on Machine Learning.* PMLR. 2020, pp. 528–539.

[15] David Bau et al. "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2017, pp. 6541–6549.

[16] Rachel KE Bellamy et al. "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias". In: *arXiv preprint arXiv:1810.01943* (2018).

[17] Alex Beutel et al. "Data decisions and theoretical implications when adversarially learning fair representations". In: *arXiv preprint arXiv:1707.00075* (2017).

[18] Frédéric Branchaud-Charron et al. "Can Active Learning Preemptively Mitigate Fairness Issues?" In: *arXiv preprint arXiv:2104.06879* (2021).

[19] Steve Brooks et al. *Handbook of Markov Chain Monte Carlo.* CRC Press, 2011.

[20] Nicolas Brosse et al. "On last-layer algorithms for classification: Decoupling representation from uncertainty estimation". In: *arXiv preprint arXiv:2001.08049* (2020).

[21] Tom Brown et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[22] Joy Buolamwini and Timnit Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency.* PMLR. 2018, pp. 77–91.

[23] Roberto Burguet, Ramon Caminal, and Matthew Ellman. "In Google we trust?" In: *International Journal of Industrial Organization* 39 (2015), pp. 44–55.

[24] Philippe Burlina et al. “Addressing artificial intelligence bias in retinal diagnostics”. In: *Translational Vision Science & Technology* 10.2 (2021), pp. 13–13.

[25] Remi Cadene et al. “Rubi: Reducing unimodal biases for visual question answering”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[26] Flavio Calmon et al. “Optimized pre-processing for discrimination prevention”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[27] Senqi Cao and Zhongfei Zhang. “Deep hybrid models for out-of-distribution detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4733–4743.

[28] Genta Cekodhima et al. “Demographic and histopathological characteristics of colorectal polyps: a descriptive study based on samples obtained from symptomatic patients”. In: *Slovenian Journal of Public Health* 55.2 (2016), pp. 118–123.

[29] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[30] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848.

[31] Liang-Chieh Chen et al. “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 801–818.

[32] Liang-Chieh Chen et al. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[33] Tianqi Chen, Emily Fox, and Carlos Guestrin. “Stochastic gradient hamiltonian monte carlo”. In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1683–1691.

[34] Xinlei Chen et al. “Microsoft coco captions: Data collection and evaluation server. arXiv 2015”. In: *arXiv preprint arXiv:1504.00325* (2015).

[35] Siddhartha Chib and Edward Greenberg. “Understanding the metropolis-hastings algorithm”. In: *The American Statistician* 49.4 (1995), pp. 327–335.

[36] Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. “An empirical study of invariant risk minimization”. In: *arXiv preprint arXiv:2004.05007* (2020).

[37] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. “Don’t Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th In-*

*ternational Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 4069–4082.

[38] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine Learning* 20 (1995), pp. 273–297.

[39] Kate Crawford and Trevor Paglen. “Excavating AI: The politics of images in machine learning training sets”. In: *AI & Society* 36.4 (2021), pp. 1105–1116.

[40] Yin Cui et al. “Class-balanced loss based on effective number of samples”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.

[41] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 248–255.

[42] Stefan Depeweg et al. “Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1184–1193.

[43] E DeRouin et al. “Neural network training on unequally represented classes”. In: *Intelligent Engineering Systems Through Artificial Neural Networks* (1991), pp. 135–145.

[44] Chris Drummond, Robert C Holte, et al. “C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling”. In: *Workshop on Learning from Imbalanced Datasets II*. Vol. 11. 2003, pp. 1–8.

[45] Mengnan Du et al. “Fairness via representation neutralization”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12091–12103.

[46] Cynthia Dwork et al. “Fairness through awareness”. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 2012, pp. 214–226.

[47] Eran Eidinger, Roee Enbar, and Tal Hassner. “Age and gender estimation of unfiltered faces”. In: *IEEE Transactions on Information Forensics and Security* 9.12 (2014), pp. 2170–2179.

[48] Charles Elkan. “The foundations of cost-sensitive learning”. In: *International Joint Conference on Artificial Intelligence*. Vol. 17. 1. Lawrence Erlbaum Associates Ltd. 2001, pp. 973–978.

[49] J Ferlay et al. “Cancer incidence and mortality worldwide: IARC CancerBase No. 11 Lyon”. In: *France: International Agency for Research on Cancer* (2013).

[50] Alberto Fernández et al. “SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary”. In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 863–905.

[51] K Ruwani M Fernando and Chris P Tsokos. “Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7 (2021), pp. 2940–2951.

[52] James R Foulds et al. “An intersectional definition of fairness”. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE. 2020, pp. 1918–1921.

[53] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian approximation: representing model uncertainty in deep learning”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1050–1059.

[54] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. “Deep bayesian active learning with image data”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1183–1192.

[55] Zhe Gan et al. “Scalable bayesian learning of recurrent neural networks for language modeling”. In: *arXiv preprint arXiv:1611.08034* (2016).

[56] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. “Image style transfer using convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 2414–2423.

[57] Robert Geirhos et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *arXiv preprint arXiv:1811.12231* (2018).

[58] Andrew Gelman et al. “A Weakly Informative Default Prior Distribution for Logistic and Other Regression Models”. In: *The Annals of Applied Statistics* (2008), pp. 1360–1383.

[59] Asma Ghandeharioun et al. “Characterizing sources of uncertainty to proxy calibration and disambiguate annotator and data bias”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 4202–4206.

[60] Camila Gonzalez et al. “Detecting when pre-trained nnu-net models fail silently for covid-19 lung lesion segmentation”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021*. Springer. 2021, pp. 304–314.

[61] Susan T Gooden. *Race and social equity: A nervous area of government*. Routledge, 2015.

[62] *Google apologises for Photos app’s racist blunder*. Accessed: April 1, 2023. July 2015. URL: `https://www.bbc.co.uk/news/technology-33347866`.

[63] *Google apologizes after its vision AI produced racist results.* Accessed: April 24, 2023. URL: `https://algorithmwatch.org/en/google-vision-racism/`.

[64] Melissa Hall et al. “A systematic study of bias amplification”. In: *arXiv preprint arXiv: 2201.11706* (2022).

[65] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[66] Haibo He et al. “ADASYN: Adaptive synthetic sampling approach for imbalanced learning”. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence).* IEEE. 2008, pp. 1322–1328.

[67] He He, Sheng Zha, and Haohan Wang. “Unlearn dataset bias in natural language inference by fitting the residual”. In: *arXiv preprint arXiv:1908.10763* (2019).

[68] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. “Why ReLu networks yield high-confidence predictions far away from the training data and how to mitigate the problem”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 41–50.

[69] Dan Hendrycks et al. “Using self-supervised learning can improve model robustness and uncertainty”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[70] Christian Henning, Francesco D’Angelo, and Benjamin F Grewe. “Are Bayesian neural networks intrinsically good at out-of-distribution detection?” In: *arXiv preprint arXiv: 2107.12248* (2021).

[71] Martin Heusel et al. “GANs trained by a two time-scale update rule converge to a local Nash equilibrium”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[72] Matthew D Hoffman et al. “Stochastic variational inference”. In: *Journal of Machine Learning Research* (2013).

[73] Alex Holub, Pietro Perona, and Michael C Burl. “Entropy-based active learning for object recognition”. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops.* IEEE. 2008, pp. 1–8.

[74] Neil Houlsby et al. “Bayesian active learning for classification and preference learning”. In: *arXiv preprint arXiv:1112.5745* (2011).

[75] Chen Huang et al. “Learning deep representation for imbalanced classification”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2016, pp. 5375–5384.

[76] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110 (2021), pp. 457–506.

[77] Pavel Izmailov et al. “What are Bayesian neural network posteriors really like?” In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4629–4640.

[78] Nathalie Japkowicz et al. “Learning from imbalanced data sets: a comparison of various strategies”. In: *AAAI Workshop on Learning from Imbalanced Data Sets*. Vol. 68. 2000, pp. 10–15.

[79] Debesh Jha et al. “TransNetR: transformer-based residual network for polyp segmentation with multi-center out-of-distribution testing”. In: *Medical Imaging with Deep Learning*. PMLR. 2024, pp. 1372–1384.

[80] Yongcheng Jing et al. “Neural style transfer: A review”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.11 (2019), pp. 3365–3385.

[81] Laurent Valentin Jospin et al. “Hands-on Bayesian neural networks—A tutorial for deep learning users”. In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48.

[82] Alain Jungo, Fabian Balsiger, and Mauricio Reyes. “Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation”. In: *Frontiers in Neuroscience* 14 (2020), p. 282.

[83] Faisal Kamiran and Toon Calders. “Data preprocessing techniques for classification without discrimination”. In: *Knowledge and Information Systems* 33.1 (2012), pp. 1–33.

[84] Kimmo Kärkkäinen and Jungseock Joo. “Fairface: Face attribute dataset for balanced race, gender, and age”. In: *arXiv preprint arXiv:1908.04913* (2019).

[85] Matthew Kay, Cynthia Matuszek, and Sean A Munson. “Unequal representation and gender stereotypes in image search results for occupations”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 3819–3828.

[86] Alex Kendall and Yarin Gal. “What uncertainties do we need in Bayesian deep learning for computer vision?” In: *Advances in Neural Information Processing Systems* 30 (2017).

[87] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[88] Saad M Khan et al. “A global review of publicly available datasets for ophthalmological imaging: barriers to access, usability, and generalisability”. In: *The Lancet Digital Health* 3.1 (2021), e51–e66.

[89] Byungju Kim et al. “Learning not to learn: Training deep neural networks with biased data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9012–9020.

[90] Nayeong Kim et al. “Learning debiased classifier with biased committee”. In: *arXiv preprint arXiv:2206.10843* (2022).

[91] Diederik P Kingma, Max Welling, et al. “An introduction to variational autoencoders”. In: *Foundations and Trends in Machine Learning* 12.4 (2019), pp. 307–392.

[92] Masanori Koyama and Shoichiro Yamaguchi. “Out-of-distribution generalization with maximal invariant predictor”. In: *OpenReview.net* (2020). Accessed: June 1, 2022.

[93] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. “Being bayesian, even just a bit, fixes overconfidence in relu networks”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5436–5446.

[94] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Accessed: February 1, 2021. 2009. URL: `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

[95] Miroslav Kubat, Stan Matwin, et al. “Addressing the curse of imbalanced training sets: one-sided selection”. In: *ICML*. Vol. 97. 1. 1997, p. 179.

[96] Matjaz Kukar, Igor Kononenko, et al. “Cost-sensitive learning with neural networks.” In: *ECAI*. Vol. 15. 27. 1998, pp. 88–94.

[97] Alexandre Lacoste et al. “Synbols: Probing learning algorithms with synthetic datasets”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 134–146.

[98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[99] Jungsoo Lee et al. “Learning debiased representation via disentangled feature augmentation”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25123–25133.

[100] Christian Leibig et al. “Leveraging uncertainty information from deep neural networks for disease detection”. In: *Scientific reports* 7.1 (2017), pp. 1–14.

[101] David D Lewis. “A sequential algorithm for training text classifiers: Corrigendum and additional data”. In: *ACM SIGIR Forum*. Vol. 29. 2. ACM New York, NY, USA. 1995, pp. 13–19.

[102] Xiaoxiao Li et al. “Estimating and improving fairness with adversarial learning”. In: *arXiv preprint arXiv:2103.04243* (2021).

[103] Yi Li and Nuno Vasconcelos. “Repair: Removing representation bias by dataset resampling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9572–9581.

[104] David A Lieberman et al. “Race, ethnicity, and sex affect risk for polyps> 9 mm in average-risk individuals”. In: *Gastroenterology* 147.2 (2014), pp. 351–358.

[105] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.

[106] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[107] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[108] Joan Lowy and Tom Krishner. *Driver killed in self-driving car accident for first time*. Accessed: September 10, 2023. June 2016. URL: `https://www.pbs.org/newshour/nation/driver-killed-in-self-driving-car-accident-for-first-time`.

[109] David JC MacKay. “The evidence framework applied to classification networks”. In: *Neural Computation* 4.5 (1992), pp. 720–736.

[110] Tanvir Mahmud, Bishmoy Paul, and Shaikh Anowarul Fattah. “PolypSegNet: A modified encoder-decoder architecture for automated polyp segmentation from colonoscopy images”. In: *Computers in Biology and Medicine* 128 (2021), p. 104119.

[111] Mykola Makhortykh, Aleksandra Urman, and Roberto Ulloa. “Detecting race and gender bias in visual representation of AI on web search engines”. In: *International Workshop on Algorithmic Bias in Search and Recommendation*. Springer. 2021, pp. 36–50.

[112] Ninareh Mehrabi et al. “A survey on bias and fairness in machine learning”. In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–35.

[113] Alireza Mehrtash et al. “Confidence calibration and predictive uncertainty estimation for deep medical image segmentation”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 3868–3878.

[114] Michele Merler et al. “Diversity in faces”. In: *arXiv preprint arXiv:1901.10436* (2019).

[115] Pablo Mesejo et al. “Computer-aided classification of gastrointestinal lesions in regular colonoscopy”. In: *IEEE Transactions on Medical Imaging* 35.9 (2016), pp. 2051–2063.

[116] Agnieszka Mikołajczyk, Sylwia Majchrowska, and Sandra Carrasco Limeros. “The (de) biasing effect of gan-based augmentation methods on skin lesion images”. In: *International*

*Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2022, pp. 437–447.

[117] George A Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[118] Julio Alejandro Murra-Saca. “El Salvador atlas gastrointestinal video endoscopy”. In: (2005).

[119] Junhyun Nam et al. “Learning from failure: De-biasing classifier from biased classifier”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20673–20684.

[120] Radford M Neal et al. “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo* 2.11 (2011), p. 2.

[121] Arkadi Nemirovski et al. “Robust stochastic approximation approach to stochastic programming”. In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1574–1609.

[122] Safiya Umoja Noble. “Algorithms of Oppression”. In: *Algorithms of Oppression*. New York University Press, 2018.

[123] Sebastian W Ober and Carl Edward Rasmussen. “Benchmarking the neural linear model for regression”. In: *arXiv preprint arXiv:1912.08416* (2019).

[124] *Pause giant AI experiments: An open letter*. Apr. 2023. URL: `https://futureoflife.org/open-letter/pause-giant-ai-experiments/`.

[125] *Percepto*. `https://percepto.co/`. Accessed: April 24, 2023.

[126] Mohammad Pezeshki et al. “Gradient starvation: A learning proclivity in neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1256–1272.

[127] Esther Puyol-Antón et al. “Fairness in cardiac MR image analysis: an investigation of bias due to data imbalance in deep learning based segmentation”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021*. Springer. 2021, pp. 413–423.

[128] Alec Radford et al. *Improving language understanding by generative pre-training*. Accessed: July 1, 2023. OpenAI, hayate-lab.com. 2018.

[129] Aditya Ramesh et al. “Hierarchical text-conditional image generation with clip latents”. In: *arXiv preprint arXiv:2204.06125* (2022).

[130] Mengye Ren et al. “Learning to reweight examples for robust deep learning”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4334–4343.

[131] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Vol. 9351. Springer International Publishing, 2015, pp. 234–241.

[132] Amelie Royer and Christoph H Lampert. “Classifier adaptation at prediction time”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1401–1409.

[133] Marco Saerens, Patrice Latinne, and Christine Decaestecker. “Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure”. In: *Neural Computation* 14.1 (2002), pp. 21–41.

[134] Shiori Sagawa et al. “Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization”. In: *arXiv preprint arXiv:1911.08731* (2019).

[135] Chitwan Saharia et al. “Photorealistic text-to-image diffusion models with deep language understanding”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494.

[136] Victor Sanh et al. “Learning from others’ mistakes: Avoiding dataset biases without modeling them”. In: *arXiv preprint arXiv:2012.01300* (2020).

[137] Prasanna Sattigeri et al. “Fairness GAN”. In: *arXiv preprint arXiv:1805.09910* (2018).

[138] Sebastian Schultheiß, Sebastian Sünkler, and Dirk Lewandowski. “We still trust in Google, but less than 10 years ago: an eye-tracking study.” In: *Information Research: An International Electronic Journal* 23.3 (2018), n3.

[139] Laleh Seyyed-Kalantari et al. “CheXclusion: Fairness gaps in deep chest X-ray classifiers”. In: *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*. World Scientific. 2020, pp. 232–243.

[140] Zheyan Shen et al. “Towards out-of-distribution generalization: A survey”. In: *arXiv preprint arXiv:2108.13624* (2021).

[141] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of Big Data* 6.1 (2019), pp. 1–48.

[142] Robik Shrestha, Kushal Kafle, and Christopher Kanan. “An investigation of critical issues in bias mitigation techniques”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 1943–1954.

[143] Robik Shrestha, Kushal Kafle, and Christopher Kanan. “Occamnets: Mitigating dataset bias by favoring simpler hypotheses”. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*. Springer. 2022, pp. 702–721.

[144] Yash Raj Shrestha and Yongjie Yang. “Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems”. In: *Algorithms* 12.9 (2019), p. 199.

[145] Daniele Silvestro and Tobias Andermann. “Prior choice affects ability of Bayesian neural networks to identify unknowns”. In: *arXiv preprint arXiv:2005.04987* (2020).

[146] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015).

[147] Brandon Smith et al. “Balancing the Picture: Debiasing Vision-Language Datasets with Synthetic Contrast Sets”. In: *arXiv preprint arXiv:2305.15407* (2023).

[148] Abhishek Srivastava et al. “MSRF-Net: A Multi-Scale Residual Fusion Network for Biomedical Image Segmentation”. In: *IEEE Journal of Biomedical and Health Informatics* 26.5 (2022), pp. 2252–2263.

[149] *Starship Technologies*. `https://www.starship.xyz/`. Accessed: April 24, 2023.

[150] Gregory V Stiegmann. “Atlas of Gastrointestinal Endoscopy”. In: *Archives of Surgery* 123.8 (1988), pp. 1026–1026.

[151] Rebecca S Stone et al. “Bayesian Uncertainty-Weighted Loss for Improved Generalisability on Polyp Segmentation Task”. In: *Workshop on Clinical Image-Based Procedures*. 2023, pp. 153–162.

[152] Rebecca S Stone et al. “Epistemic Uncertainty-Weighted Loss for Visual Bias Mitigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2898–2905.

[153] Harini Suresh and John V Guttag. “A framework for understanding unintended consequences of machine learning”. In: *arXiv preprint arXiv:1901.10002* 2.8 (2019).

[154] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. “End: Entangling and disentangling deep representations for bias correction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13508–13517.

[155] B Thomas and MD Fitzpatrick. “The validity and practicality of sun-reactive skin types i through vi”. In: *Arch Dermatol* 124.6 (1988), pp. 869–871.

[156] Bart Thomee et al. “YFCC100M: The new data in multimedia research”. In: *Communications of the ACM* 59.2 (2016), pp. 64–73.

[157] William Thong and Cees GM Snoek. “Feature and Label Embedding Spaces Matter in Addressing Image Classifier Bias”. In: *arXiv preprint arXiv:2110.14336* (2021).

[158] James Titcomb. *Robot passport checker rejects Asian man's photo for having his eyes closed.* Accessed: November 2, 2022. Dec. 2016. URL: `https://www.telegraph.co.uk/technology/2016/12/07/robot-passport-checker-rejects-asian-mans-photo-having-eyes/`.

[159] Eric Tzeng et al. “Simultaneous deep transfer across domains and tasks”. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2015, pp. 4068–4076.

[160] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. “Towards debiasing NLU models from unknown biases”. In: *arXiv preprint arXiv:2009.12303* (2020).

[161] Matias Valdenegro-Toro and Daniel Saromo Mori. “A deeper look into aleatoric and epistemic uncertainty disentanglement”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* IEEE. 2022, pp. 1508–1516.

[162] Kush R Varshney and Homa Alemzadeh. “On the safety of machine learning: Cyber-physical systems, decision sciences, and data products”. In: *Big Data* 5.3 (2017), pp. 246–255.

[163] Christina Wadsworth, Francesca Vera, and Chris Piech. “Achieving fairness through adversarial learning: an application to recidivism prediction”. In: *arXiv preprint arXiv:1807.00199* (2018).

[164] Angelina Wang and Olga Russakovsky. “Directional bias amplification”. In: *International Conference on Machine Learning.* PMLR. 2021, pp. 10882–10893.

[165] Mei Wang and Weihong Deng. “Mitigating bias in face recognition using skewness-aware reinforcement learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020, pp. 9322–9331.

[166] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. “Learning to model the tail”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[167] Zeyu Wang et al. “Towards fairness in visual recognition: Effective strategies for bias mitigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020, pp. 8919–8928.

[168] *Wayve Technologies Ltd.* `https://wayve.ai/`. Accessed: April 1, 2023.

[169] Gary M Weiss, Kate McCarthy, and Bibi Zabar. “Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?” In: *Dmin* 7.35-41 (2007), p. 24.

[170] 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)*. 2011, pp. 681–688.

[171] Florian Wenzel et al. “How good is the Bayes posterior in deep neural networks really?” In: *arXiv preprint arXiv:2002.02405* (2020).

[172] Depeng Xu et al. “Fairgan: Fairness-aware generative adversarial networks”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 570–575.

[173] Xingkun Xu et al. “Consistent instance false positive improves fairness in face recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 578–586.

[174] Kaiyu Yang et al. “Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 547–558.

[175] Xiaoyong Yang et al. “Colon polyp detection and segmentation based on improved MR-CNN”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2020), pp. 1–10.

[176] Michael Yeung et al. “Focus U-Net: A novel dual attention-gated CNN for polyp segmentation during colonoscopy”. In: *Computers in Biology and Medicine* 137 (2021), p. 104815.

[177] Tianhe Yu et al. “Gradient surgery for multi-task learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5824–5836.

[178] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. “Mitigating unwanted biases with adversarial learning”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 335–340.

[179] Ruqi Zhang et al. “Cyclical stochastic gradient MCMC for Bayesian deep learning”. In: *arXiv preprint arXiv:1902.03932* (2019).

[180] Zhifei Zhang, Yang Song, and Hairong Qi. “Age progression/regression by conditional adversarial autoencoder”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5810–5818.

[181] Jieyu Zhao et al. “Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints”. In: *EMNLP*. 2017.

[182] Shengbing Zhao et al. “Magnitude, risk factors, and factors associated with adenoma miss rate of tandem colonoscopy: a systematic review and meta-analysis”. In: *Gastroenterology* 156.6 (2019), pp. 1661–1674.

[183] Bolei Zhou et al. “Places: A 10 million image database for scene recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2017), pp. 1452–1464.

[184] Zhi-Hua Zhou and Xu-Ying Liu. “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *IEEE Transactions on Knowledge and Data Engineering* 18.1 (2005), pp. 63–77.

# Appendix

## Supplementary materials for Chapter 3

Table 6.1 shows the numerical results for the face detection experiment, the top 7 rows (the most challenging subgroups) of which are shown visually in Figure 3.11.

## Supplementary materials for Chapter 4

### Hyper-parameter selection

We fix de-biasing scalar term $\kappa$ in EpiUpWt to 5.0 for all experiments as we find negligible differences in performance for values $3 < \kappa < 7$.

Table 6.2 shows the optimal parameters chosen for all Bayesian architectures. A ResNet18 architecture and SGD optimizer are fixed across all experiments. We refer to the Appendix of [143] for hyperparameters and settings for the remaining methods.

For cSG-MCMC [179], we find that cyclical Stochastic Gradient Hamiltonian Monte-Carlo (SGHMC) performs better than Stochastic Gradient Langevin Dynamics (SGLD). SGHMC introduces an auxiliary momentum variable $\alpha$ which acts as a tempering to the noise variable. We direct readers to [179] for more details related to SGHMC.

Finally, for the COCO-on-Places and BAR datasets, we find that selecting a smaller sampling number $M$ for $\boldsymbol{\Theta}$ gives comparable results compared to sharpening all of the posterior samples. Our subset includes the earlier samples from each sampling phase of the learning schedule, so as to capture the most diversity. Having a smaller subset speeds up the sharpening phase.

## Supplementary materials for Chapter 5

### $L_R$ mean uncertainties

Figure 6.1 shows the mean uncertainties (for minority and majority subgroups of the training set) of each bias variable in Biased MNIST after fine-tuning for the Bayesian model fine-tuned with $L_R$ loss alone. While $L_R$ is not used alone, these results further confirm the findings in Chapter 4 that fine-tuning the posterior estimates with either PCGrad$(L_{CE}, L_R)$ or $L_R$ loss does not drive the subgroup uncertainties to zero.

| | Baseline TPR (%) | EpiUpWt TPR (%) | TPR gap (%) |
|---|---|---|---|
| Age: 0-2 | 40.28 | 43.59 | 3.31 |
| Age: 3-9 | 54.32 | 57.08 | 2.76 |
| Age >70 | 55.34 | 58.05 | 2.71 |
| Middle Eastern + male | 59.59 | 62.76 | 3.17 |
| Middle Eastern + female | 62.47 | 67.68 | 5.21 |
| Black + female | 67.30 | 70.44 | 3.14 |
| Latino Hispanic + female | 66.52 | 70.54 | 4.02 |
| Age: 50-59 | 62.99 | 66.33 | 3.34 |
| Age: 30-39 | 69.02 | 71.01 | 1.99 |
| Age: 20-29 | 70.21 | 72.58 | 2.37 |
| Age: 40-49 | 68.62 | 72.23 | 3.61 |
| Age: 10-19 | 64.88 | 68.61 | 3.73 |
| Age: 60-69 | 59.21 | 61.45 | 2.24 |
| East Asian | 67.68 | 70.94 | 3.26 |
| Latino Hispanic | 66.36 | 69.42 | 3.06 |
| Southeast Asian | 67.18 | 69.52 | 2.34 |
| Black | 66.49 | 69.02 | 2.53 |
| Indian | 68.48 | 70.51 | 2.03 |
| Middle Eastern | 60.53 | 64.37 | 3.84 |

Table 6.1: Numeric values for the true positives rates (TPRs) for the most challenging subgroups with lowest TPRs in the FairFaces test set, showing that the uncertainty-weighted loss decreases the TPR gap for each subgroup.

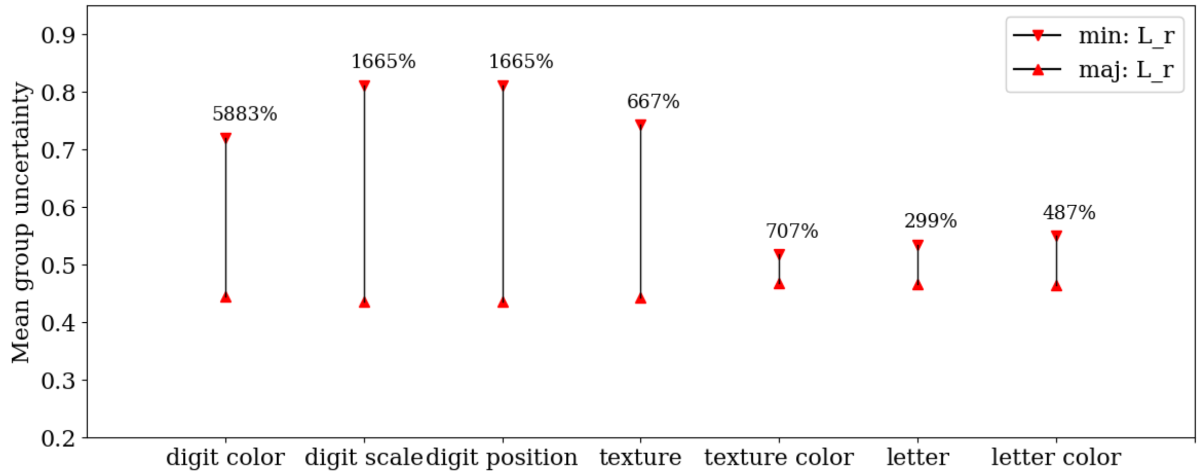


Figure 6.1: Mean uncertainties (for minority and majority subgroups of the training set) of each bias variable in Biased MNIST after fine-tuning for the Bayesian model fine-tuned with $L_R$ loss alone. The length of the black lines joining the pair of markers for each bias variable represent the disparity in mean uncertainties; and the number above each pair is the percentage difference between the baseline Bayesian model (mean subgroup uncertainties before any fine-tuning, always higher for minority subgroups) compared to after fine-tuning. Compared to $L_{CE}$ and PCGrad($L_{CE}, L_R$), $L_R$ causes the gap between minority and majority subgroup uncertainties to increase much more - and also for both means to rise.

| Architecture+Method | LR | Epochs | Cycle Length | Cycle Count | Posterior Samples (M) | Noise Friction Param ($\alpha$) | Temperature (T) |
|---|---|---|---|---|---|---|---|
| | | | *Biased MNIST* | | | | |
| BayResNet | 0.1 | - | 600 | 2 | 10 | 0.3 | 2e-5 |
| BayResNet+EpiUpWt | 0.1 | - | 600 | 2 | 20 | 0.3 | 2e-5 |
| BayResNet+$L_{CE/RCE}$ | 0.015 | 10 | - | - | 10 | - | - |
| | | | *COCO-on-Places* | | | | |
| BayResNet | 0.5 | - | 1000 | 2 | 12 | 0.2 | 1.4e-4 |
| BayResNet+EpiUpWt | 0.5 | - | 1000 | 2 | 20 | 0.3 | 1.4e-4 |
| BayResNet+$L_{CE/RCE}$ | 0.01 | 14 | - | - | 6 | - | - |
| | | | *Biased Action Recognition (BAR)* | | | | |
| BayResNet | 0.01 | - | 200 | 3 | 15 | 0.2 | 6e-4 |
| BayResNet+EpiUpWt | 0.01 | - | 200 | 3 | 18 | 0.2 | 6e-4 |
| BayResNet+$L_{CE/RCE}$ | 0.015 | 19 | - | - | 9 | - | - |

Table 6.2: Hyperparameters chosen for each reported experiment in Chapter 4 for the baseline Bayesian ResNet18, EpiUpWt, and the posterior estimate fine-tuning method BayResNet+$L_{RCE}$ and BayResNet+$L_{CE}$: learning step size (LR), epochs, cycle length, number of cycles, total posterior samples (M), noise loss parameter $\alpha$, cooling temperature for posterior (T).

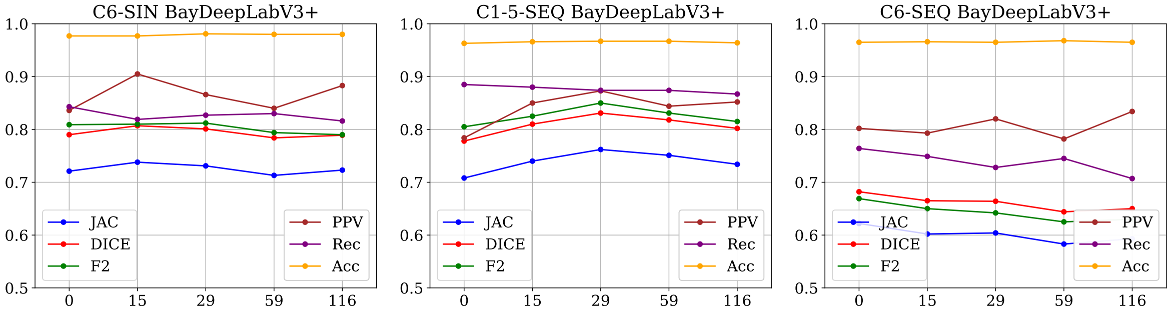

C6-SIN BayDeepLabV3+
C1-5-SEQ BayDeepLabV3+
C6-SEQ BayDeepLabV3+

Figure 6.2: Baseline Bayesian BayDeepLabV3+ results for the bias mitigation experiments evaluated on the three test datasets, demonstrating that adding more data from the test distribution does *not* always help the model generalize, and neither does adding larger quantities of it always improve test performance.

### Tabular results for bias mitigation experiments on PolypGen dataset

Figure 6.2 shows a graphical representation of the bias mitigation experiment results on the BayDeepLabV3+ model.

Table 6.3, Table 6.4, Table 6.5 and Table 6.6 show the deterministic DeepLabV3+, BayDeepLabV3+, BayDeepLabV3+EpiUpWt and BayDeepLabV3+EpiUpWt+$L_{RCE}$ results respectively for the bias mitigation experiments.

| Dataset | Number IDS | JAC | Dice | F2 | PPV | Recall | Accuracy |
|---|---|---|---|---|---|---|---|
| C6-SIN | 15 | 0.736 | 0.803 | 0.793 | 0.900 | 0.802 | 0.980 |
| | 29 | 0.770 | 0.838 | 0.822 | 0.945 | 0.815 | 0.979 |
| | 59 | 0.730 | 0.797 | 0.782 | 0.913 | 0.787 | 0.977 |
| | 116 | 0.735 | 0.804 | 0.794 | 0.912 | 0.795 | 0.978 |
| C1-5-SEQ | 15 | 0.732 | 0.792 | 0.808 | 0.875 | 0.835 | 0.964 |
| | 29 | 0.725 | 0.804 | 0.808 | 0.872 | 0.829 | 0.955 |
| | 59 | 0.728 | 0.806 | 0.802 | 0.884 | 0.819 | 0.955 |
| | 116 | 0.772 | 0.840 | 0.853 | 0.880 | 0.879 | 0.965 |
| C6-SEQ | 15 | 0.736 | 0.803 | 0.793 | 0.900 | 0.802 | 0.980 |
| | 29 | 0.770 | 0.838 | 0.822 | 0.945 | 0.815 | 0.979 |
| | 59 | 0.730 | 0.797 | 0.782 | 0.913 | 0.787 | 0.977 |
| | 116 | 0.735 | 0.804 | 0.794 | 0.912 | 0.795 | 0.978 |

Table 6.3: Bias mitigation experiment results for the state-of-the-art deterministic DeepLabV3+ for different number of in-distribution samples (IDS) which for the minority group added into the training data.

| Dataset | Number IDS | JAC | Dice | F2 | PPV | Recall | Accuracy |
|---|---|---|---|---|---|---|---|
| C6-SIN | 15 | 0.738 | 0.807 | 0.810 | 0.905 | 0.819 | 0.977 |
| | 29 | 0.731 | 0.801 | 0.812 | 0.866 | 0.827 | 0.981 |
| | 59 | 0.713 | 0.784 | 0.794 | 0.840 | 0.830 | 0.980 |
| | 116 | 0.723 | 0.789 | 0.790 | 0.883 | 0.816 | 0.980 |
| C1-5-SEQ | 15 | 0.740 | 0.810 | 0.825 | 0.850 | 0.880 | 0.966 |
| | 29 | 0.762 | 0.831 | 0.850 | 0.873 | 0.874 | 0.967 |
| | 59 | 0.751 | 0.818 | 0.831 | 0.844 | 0.874 | 0.967 |
| | 116 | 0.734 | 0.802 | 0.815 | 0.852 | 0.867 | 0.964 |
| C6-SEQ | 15 | 0.602 | 0.665 | 0.650 | 0.793 | 0.749 | 0.966 |
| | 29 | 0.604 | 0.664 | 0.642 | 0.820 | 0.728 | 0.965 |
| | 59 | 0.583 | 0.644 | 0.625 | 0.782 | 0.745 | 0.968 |
| | 116 | 0.593 | 0.650 | 0.630 | 0.834 | 0.707 | 0.965 |

Table 6.4: Bias mitigation experiment results for the baseline Bayesian model, BayDeepLabV3+ for different number of in-distribution samples (IDS) which for the minority group added into the training data.

| Dataset | Number IDS | JAC | Dice | F2 | PPV | Recall | Accuracy |
|---|---|---|---|---|---|---|---|
| C6-SIN | 15 | 0.710 | 0.778 | 0.776 | 0.873 | 0.783 | 0.977 |
| | 29 | 0.716 | 0.788 | 0.774 | 0.875 | 0.790 | 0.977 |
| | 59 | 0.736 | 0.807 | 0.801 | 0.890 | 0.816 | 0.978 |
| | 116 | 0.740 | 0.810 | 0.810 | 0.907 | 0.834 | 0.981 |
| C1-5-SEQ | 15 | 0.727 | 0.797 | 0.811 | 0.830 | 0.870 | 0.964 |
| | 29 | 0.744 | 0.813 | 0.826 | 0.858 | 0.869 | 0.965 |
| | 59 | 0.760 | 0.827 | 0.840 | 0.877 | 0.876 | 0.968 |
| | 116 | 0.770 | 0.837 | 0.848 | 0.870 | 0.872 | 0.967 |
| C6-SEQ | 15 | 0.602 | 0.665 | 0.650 | 0.793 | 0.749 | 0.966 |
| | 29 | 0.604 | 0.664 | 0.642 | 0.820 | 0.728 | 0.965 |
| | 59 | 0.610 | 0.667 | 0.643 | 0.820 | 0.745 | 0.968 |
| | 116 | 0.625 | 0.692 | 0.660 | 0.834 | 0.790 | 0.972 |

Table 6.5: Bias mitigation experiment results for BayDeepLabV3+EpiUpWt for different number of in-distribution samples (IDS) which for the minority group added into the training data.

| Dataset | Number IDS | JAC | Dice | F2 | PPV | Recall | Accuracy |
|---|---|---|---|---|---|---|---|
| C6-SIN | 15 | 0.758 | 0.865 | 0.812 | 0.914 | 0.817 | 0.976 |
| | 29 | 0.760 | 0.864 | 0.813 | 0.910 | 0.815 | 0.975 |
| | 59 | 0.762 | 0.866 | 0.811 | 0.890 | 0.816 | 0.978 |
| | 116 | 0.759 | 0.868 | 0.811 | 0.861 | 0.847 | 0.979 |
| C1-5-SEQ | 15 | 0.758 | 0.817 | 0.815 | 0.927 | 0.856 | 0.965 |
| | 29 | 0.701 | 0.769 | 0.794 | 0.841 | 0.836 | 0.963 |
| | 59 | 0.767 | 0.827 | 0.840 | 0.877 | 0.866 | 0.968 |
| | 116 | 0.769 | 0.836 | 0.847 | 0.867 | 0.870 | 0.967 |
| C6-SEQ | 15 | 0.641 | 0.643 | 0.676 | 0.906 | 0.810 | 0.969 |
| | 29 | 0.643 | 0.731 | 0.685 | 0.877 | 0.806 | 0.944 |
| | 59 | 0.700 | 0.790 | 0.698 | 0.890 | 0.834 | 0.967 |
| | 116 | 0.760 | 0.827 | 0.840 | 0.907 | 0.876 | 0.968 |

Table 6.6: Bias mitigation experiment results for BayDeepLabV3+$L_{RCE}$ for different number of in-distribution samples (IDS) which for the minority group added into the training data.