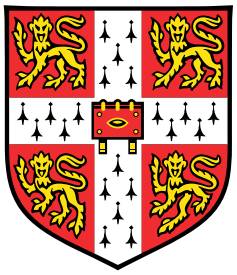
Second-Order Optimisation and Imbalanced Class Distribution in Emotional Analysis



Andrej Jovanović

Department of Engineering
University of Cambridge

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

Darwin College August 2024

"Objectives are well and good when they are sufficiently modest, but things get a lot more complicated when they're more ambitious. In fact, objectives actually become obstacles towards more exciting achievements, like those involving discovery, creativity, invention, or innovation—or even achieving true happiness. In other words (and here is the paradox), the greatest achievements become less likely when they are made objectives. Not only that, but this paradox leads to a very strange conclusion—if the paradox is really true then the best way to achieve greatness, the truest path to "blue sky" discovery or to fulfill boundless ambition, is to have no objective at all."

- Kenneth O. Stanley, Why Greatness Cannot be Planned: The Myth of Objective

Declaration

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

The project began as a *tabula rasa* where I implemented all code, unless otherwise stated. The code itself was developed using Python utilising the `PyTorch` (Paszke et al., 2019), `TorchVision` (maintainers and contributors, 2016), `scikit-learn` (Pedregosa et al., 2011), `wandb` (Biewald, 2020), `transformers` (Wolf et al., 2020) and `peft` (Mangrulkar et al., 2022) libraries, in addition to the following resources:

1. Xiaodong Wu, my supervisor, provided the implementation for both the iEF (Wu et al., 2024b)[1] and GDN (Benzing, 2022) optimisers that were used throughout this thesis. Similarly, we rely on the work of George et al. (2018) for the implementation of the EKFAC preconditioner.

2. I used the public repositories of Cao et al. (2019), Zinnen and Salhab (2023) and Park et al. (2021) to implement the baseline loss correction methods that were standard in the literature.

We make our code available at the following repository.

The word count, excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices is given below.
Word Count: $14,966$

Andrej Jovanović
August 2024

[1] At the time of writing, the *iEF* codebase has not been released publicly; as such, we provide an anonymous link to our materials.

Acknowledgements

Firstly, I would like to express my heartfelt gratitude to my supervisors Xiaodong Wu, Wen Wu and Dr. Brian Sun. Your timely advice and insightful discussions have not only positively impacted the quality of this thesis, but my own personal satisfaction throughout. Thank you for providing a thought-provoking question that has exposed me to a new field of machine learning research.

To my study group: whilst a journey in academia is oftentimes a lonely one, it certainly helps when you have colleagues-turned-friends along for the ride. Thank you for many hours of collaborative work and encouragement.

To Anka, my partner: you, perhaps more intimately than anyone, understands both the highs and lows that this degree provided. Thank you for being my emotional rock - I am not too sure what I would have done without you.

Finally, to my parents Aleksandra and Siniša: it is because of your sacrifice that I was able to pursue further study at this incredible institution. To you, I will be forever indebted; I hope that I made you proud.

Abstract

Supervised learning, in its standard formulation, typically assumes that all classes are represented *roughly* equally; this is crucial for models that are trained to minimise the empirical risk. In many naturally-occurring datasets, however, this condition is not met, where instead datasets suffer from *class imbalance*: certain class(es) make up an overwhelming majority of samples in the training data. This pathology is not necessary novel, where many solutions have either corrected for the training disparity through re-weighting the loss function, or augmenting the training data through sampling methods.

Second-order methods have attractive theoretical guarantees that have a natural applicability to the class imbalance problem. These optimisation methods ensure that each sample, irrespective of its relative prevalence in the mini-batch, has an equal loss reduction in the gradient step. Previous work in the literature has alluded to this fact, but second-order methods have never been applied to the class imbalance problem in practice due to the severe computational overhead in realistic machine learning problems. However, recent work has shown that sophisticated approximations to the preconditioning matrix can still provide the convergence benefits typically associated with second-order methods, whilst being computationally tractable for large neural networks.

To this end, our work makes three novel contributions. Firstly, we prove the suitability of second-order optimisers as a class-imbalance correction mechanism, which are able to outperform many loss-level corrections whilst being inherently simpler to tune. Secondly, we provide a *unified* mathematical framework through which we can understand many solutions that have been offered for the class imbalance problem. Finally, we rigorously evaluate our claims over three datasets which span two modalities: vision and speech. In particular, we focus our attention on the speech emotion recognition task to create a new domain of evaluation for class imbalance solutions.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Machine learning, in an ideal mathematically formalised setting, rests on a few assumptions, namely:

i All data is independent and has been drawn from the same underlying data generating process (I.I.D).

ii *Enough* data (in the view that it sufficiently approximates the underlying data distribution $p(x)$) has been gathered to model the task sufficiently well.

iii In supervised learning tasks, there is an *equal* or *almost equal* representation amongst classes.

Even if points i) and ii) are met, point iii) is often not fulfilled in practical scenarios, where certain classes occur more infrequently than others (Dua and Graff, 2017; Everingham et al., 2010; Lin et al., 2014; Liu et al., 2019; Van Horn et al., 2018). In these settings, the underlying data suffers from a *class imbalance* (CI): there are distinct majority and minority class groupings, where samples in the majority class are significantly more represented than that of the latter (Johnson and Khoshgoftaar, 2019). This poses a problem when building a classifier, albeit naively, as the model will learn to over-classify the majority class, while undermining the performance of the minority class which is often the class of interest[1] - we consider this to be the case for the scope of this thesis. As such, we have to perform well across all classes, and cannot simply ignore minority samples in the view that they are rare and insignificant. The detrimental influence of CI is illustrated in Figure 1.1 for a small toy problem. While a simple problem, it clearly illustrates that in cases of large data imbalance, our classifiers learn poor decision boundaries where the model has not learnt

[1] In the case of predicting a rare cancer, for example, we are more interested in the cases where the disease is present than where it is is not.

to discriminate between positive and negative samples. This is exacerbated in real-world contexts, for example in the medical domain, where identifying minority classes is of critical importance.

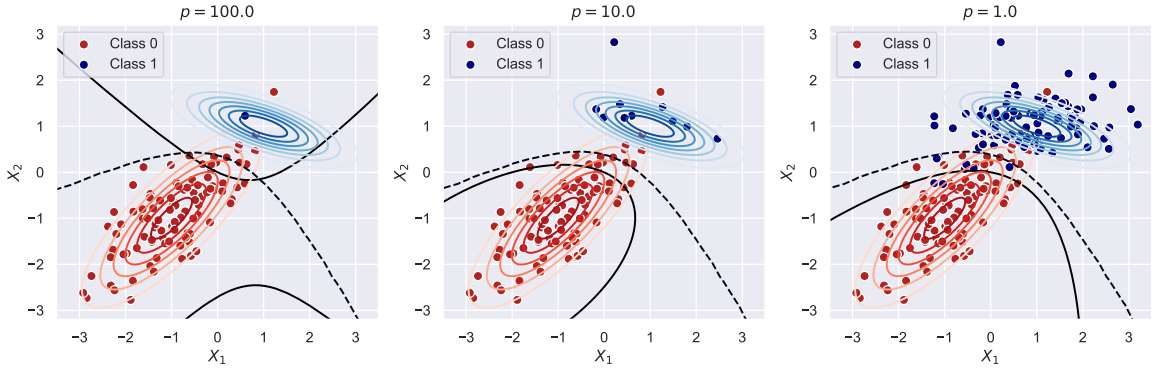


Fig. 1.1 Pathological effect induced by CI on a binary classifier's boundary for a synthetic two-class problem. Underlying data is generated from two multivariate Gaussian distributions with known means and covariance matrices. With this knowledge, we derive the optimal parabolic decision boundary (black dotted line) analytically (see Appendix A.1 for proof). Without the knowledge of the underlying data distribution $p(x)$, we train a classifier (a logistic regression with a parabolic basis function) that approximates the decision boundary $p(y|x)$ (solid black line). As the amount of CI increases (denoted by $p = \frac{\max_i n_i}{\min_i n_i}$ where n_i indicates the number of samples in class i.) the model learns worse decision boundaries relative to the theoretical decision boundary, leading to poor generalisation on the true underlying data distribution.

CI, however, is not a novel problem, and it has received much attention in various contexts (Chawla et al., 2004; He and Garcia, 2009; Japkowicz, 2000; Johnson and Khoshgoftaar, 2019). Typically, proposed solutions in the literature can be grouped into two categories: *data-level* or *algorithm-level* methods (Buda et al., 2018; Japkowicz, 2000; Johnson and Khoshgoftaar, 2019). These remedies aim to correct against the dominance in loss, and hence the gradient update, induced by the majority class which leads to poor generalisation on the minority classes for standard first-order optimisers such as SGD (Anand et al., 1993; Francazi et al., 2024; Kunstner et al., 2024; Robbins and Monro, 1951). In this work, we propose an alternate approach to this, viewed through the lens of optimisation. Second-order optimisers (Benzing, 2022; George et al., 2018; Petersen et al., 2023; Wu et al., 2024b), unlike first-order methods, have an attractive property in that for a particular gradient update, each sample, regardless of its class, has an equal loss reduction (Wu et al., 2024b). This has a natural applicability to the CI problem as the the correction mechanism needed to alleviate the difficulties in model optimisation is baked into the gradient update itself. Although this fact has been alluded to in previous work (Park et al., 2021), it has never been explored in

practice due to the computational overhead of second-order methods. Our work aims to fill this gap.

1.1 SER as a Class Imbalance Benchmark

Standard benchmarks for CI problems have been restricted entirely to the computer vision domain (Cao et al., 2019; Cui et al., 2019; Guo et al., 2016; Krizhevsky, 2009; Van Horn et al., 2018). Instead, we introduce a novel focus on the SER domain (Cao et al., 2014; Lotfian and Busso, 2019). SER is inherently class-imbalanced where either the vast majority of common daily conversations are emotionally neutral (Lotfian and Busso, 2019), and/or the scripted emotional expressions in datasets are prototypical and exaggerated, rather than mimicking everyday emotional responses (Cao et al., 2014). A classifier trained to identify an emotion class associated to an audio recording will learn to over-prioritise frequently occurring emotions (*neutrality*) or emotions that have very distinct audio features (e.g. *anger*) in the naïve case. Tackling the CI problem, in this setting, lies at the heart of creating models that are robust across the entire range of emotions, which is a quality we would expect from performant SER systems.

1.2 Research Questions

In this thesis, our objective is to answer the following research questions (RQ):

RQ.1: Can second-order optimisers be used to alleviate the class-imbalance problem *during training*?

RQ.2: If the aforementioned hypothesis is validated experimentally, under what conditions do second-order optimisers allow the network to learn a more fair representation?

RQ.3: Are class-imbalance corrections effective for the SER domain, and if so, which are the most performant?

1.3 Contributions

The major contributions of this project are as follows:

1. Experimental evidence that, under the correct settings, second-order optimisers can be used as an effective class-imbalance correction mechanism supported by theoretical motivations. Furthermore, we show these methods can out-perform loss-level corrections whilst being easier to *tune*.

2. A unified mathematical framework through which all CI correction methods can be viewed and more clearly understood.

3. Rigorous experimentation of various class-imbalance-correcting methods across three datasets which span two modalities, vision and the novel speech emotion recognition domain.

1.4 Structure of Thesis

The remainder of this thesis is structured as follows: **Chapter 2** provides the necessary background needed to understand both SER and CI methods. **Chapter 3** focuses on the mathematical preliminaries of the thesis, establishing a unified mathematical framework across various CI correction methods. **Chapter 4** introduces the methodology along with the results for CI experiments on a standard vision benchmark and two SER datasets. **Chapter 5** provides a detailed discussion of the results, providing a deeper insight into second-order optimisers as CI methods. **Chapter 6** summarises the main findings of this thesis, and suggests promising directions for future work.

Chapter 2

Background and Related Work

In this chapter, we provide an overview of the literature required to understand the work put forward in this thesis. We begin by understanding various methods that are that are used to tackle the CI problem, irrespective of the application domain. Next, we provide an overview of second-order optimisers using natural gradient descent, highlighting their applicability to CI. We conclude by introducing the SER task, and the various solutions tackling the problem to date.

2.1 Methods to Tackle Class Imbalance

In the CI literature, solutions that occur *during training* are typically grouped into two categories: *data-level* and *algorithm-level* methods. In addition, there are *post-training* correction methods that have been explored (Buda et al., 2018; Japkowicz, 2000; Johnson and Khoshgoftaar, 2019).

2.1.1 Data-Level

Data-level methods, intuitively, can be thought of as data augmentations, where the aim is to create a balanced dataset from one that was imbalanced (see Figure 2.1). In particular, each method can be viewed as an *over-sampling* or *under-sampling* technique (Buda et al., 2018; Japkowicz, 2000; Johnson and Khoshgoftaar, 2019). The former induces a balanced dataset through augmenting the data with new minority samples generated from the pre-existing data, whilst the latter removes samples from the majority class randomly. Whilst shown to be performant in some cases (Masko and Hensman, 2015; Van Hulse et al., 2007), their applicability is limited. Undersampling could remove very important majority class samples, whilst over-sampling could cause models to overfit to the training data due to a lack of

data diversity (Cao et al., 2019; Cui et al., 2019). Furthermore, oversampling would be computationally infeasible for data that has a large memory footprint as is the case in SER (Johnson and Khoshgoftaar, 2019).

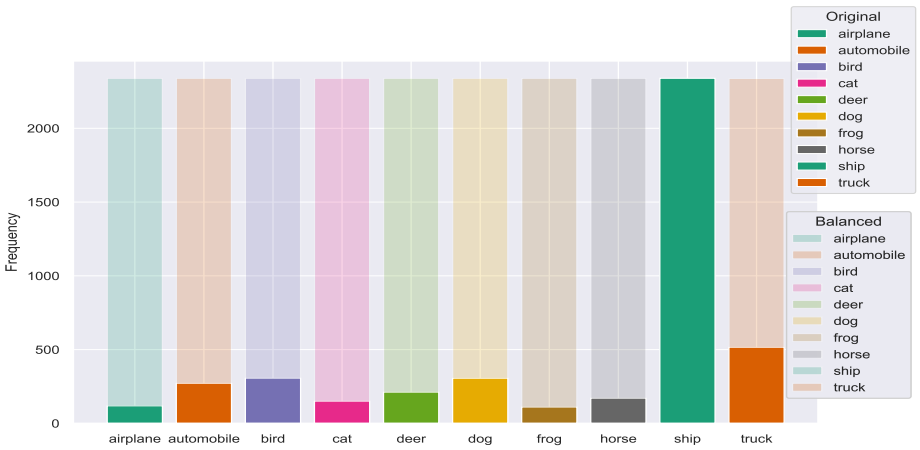


Fig. 2.1 Example of *data-level* imbalance correction using SMOTETomek (Batista et al., 2003). The minority classes have been over-sampled, indicated by the shaded background for each class.

2.1.2 Algorithm-Level

Algorithm-level methods do not adjust the underlying training data, but rather modify the *training* process itself to account for the data imbalance (Johnson and Khoshgoftaar, 2019). These methods can be thought of as cost-sensitive learning (or importance-weighting (Byrd and Lipton, 2019)) as they prioritise minority samples by assigning larger weights to these samples in conjunction with the existing loss function[1]. When choosing how to assign the corrective weight, we can perform this at a global level, where each sample in a class receives the same weight. A common strategy is to use the inverse, or inverse square-root, of the class frequency (Huang et al., 2016; Mahajan et al., 2018; Mikolov et al., 2013; Wang et al., 2017). Cui et al. (2019) proposed an extension with a hyperparameter-controlled class weight inspired by effective samples. This allows the class weight to exist on a spectrum between no correction and the inverse class frequency as before.

Alternatively, one can approach the class weight from a local, per-sample point-of-view. Intuitively, we assume that majority samples, due to their prevalence in the dataset, will be well-classified by the model. The opposite is true for minority samples. In this framework,

[1]This is a useful remedy in supervised learning as typical loss functions, such as the cross-entropy loss, treat every class equally irrespective of its representation in the training data.

we wish to assign a higher priority (weight) to those samples that have been poorly classified; we can interpret these as minority samples, but this will also extend to hard examples more generally. Lin et al. (2018) follow this idea by introducing a focal loss which modulates the per-sample loss by how well it is classified. This local correction can be applied in conjunction with the global prior-based re-weighting (Huang et al., 2016; Wang et al., 2017). Cao et al. (2019) introduce a label-distribution-aware margin loss which encourages the classifier to learn per-class margins, where the minority class will have larger margins. As we will see in Chapter 3, this creates a more complex regularisation of the majority-class gradient-contribution as opposed to a scalar factor introduced by other methods in the literature. Most similar to our work, Park et al. (2021) introduce an influence-balanced loss that is a loose first-order approximation of influence functions (Markatou and Ronchetti, 1997; Robinson, 1984). The intuition behind their approach is to re-weight each sample's loss proportional to the inverse of the influence of the sample. In practice, this results in the loss being scaled by the inverse of the magnitude of the gradient vector. Much like other work in the literature, the influence-balanced loss attempts assign lower importance to majority-class samples through attenuating the loss and hence the gradient. Broadly speaking, algorithm-level methods will form the baseline against which we compare for this thesis.

2.1.3 Post-Training Calibration

Buda et al. (2018) have shown that models can be corrected for class-imbalance *post-training* where the output decision threshold is adjusted to reflect the class priors. Even though this brings improvements over the baseline model, we do not consider these methods for the purposes of this thesis. These *post-training* methods do not change what the model has learned, which is the question of interest in the context of this thesis.

2.2 Second-Order Optimisers

Nowadays, machine learning models are typically trained to minimise a differentiable loss function via gradient descent methods and their stochastic variants (Kingma and Ba, 2017; Loshchilov and Hutter, 2018; Robbins and Monro, 1951). To further accelerate training and improve generalisation by taking into account the curvature of the loss landscape, the gradient update can be preconditioned with the Hessian, resulting in Newton's method (Galántai, 2000). However, computing the Hessian for large models that are commonplace in ML places a prohibitive computational overhead on their adoption (Kunstner et al., 2020; Park et al.,

2021)[2]. Instead, many have turned to using NGD (Amari, 1998) as a second-order method, which uses the Fisher information matrix as the preconditioner. In particular, the K-FAC method by Martens and Grosse (2020) has made this approach tractable for neural networks through sophisticated approximations of the Fisher [3]. Improving on the computational cost further, many methods approximate the true Fisher matrix with the empirical Fisher (EF) matrix, which is computed directly from the gradients of the involved training samples (George et al., 2018; Roux et al., 2007; Yang et al., 2022; Zhang et al., 2022a). This is despite having known theoretical limitations (Kunstner et al., 2020), which Wu et al. (2024b) have corrected in their work introducing the improved Empirical Fisher (iEF) method by addressing the *inversely-scaled projection issue* (Kunstner et al., 2020) (see Chapter 3).

Nevertheless, an interesting property of these methods related to the CI problem is that they theoretically guarantee that the proposed parameter update direction ensures a decrease in loss across all samples; for iEF this is strictly according to each involved sample's corresponding gradient norm (Wu et al., 2024b). This is not the case for typical first-order methods such as SGD (Robbins and Monro, 1951); it is believed that being able to resolve conflicting gradient update directions is what drives the superior performance of second-order methods (Benzing, 2022; Wu et al., 2024b). Park et al. (2021) alluded to the use of second-order optimisation in their work; however, they proceed with an approximation applied at the loss level due to the prohibitive cost of the Hessian. With tractable methods to compute the second-order updates for neural networks, we posit that the optimisation step taken by these approximate NGD methods will embed the *importance* of each sample into the training regime, preventing the gradient domination of majority classes, allowing for both minority and majority samples to be learnt equally well. This application will be the focus of this thesis.

2.3 Methods to Tackle SER

SER is the task where we wish to train a model to predict the emotional state expressed by a speaker based on their spoken utterance (Busso et al., 2004). As mentioned in Chapter 1.1, SER is inherently class-imbalanced where the vast majority of common daily conversations are emotionally neutral (Lotfian and Busso, 2019), and/or the scripted emotional expressions in the datasets are prototypical and exaggerated, rather than mimicking everyday emotional responses (Cao et al., 2014). Typically, proposed solutions involve training transformer-style

[2]The Hessian matrix is $O(P^2)$ in space, and the inversion of the Hessian is $O(P^3)$. As the number of parameters scale, this becomes prohibitive.

[3]We point the reader to the original work for a more in-depth discussion of how these approximations are derived.

models (such as vision transformers (Dosovitskiy et al., 2020), conformers (Gulati et al., 2020), SepTr (Ristea et al., 2022) etc.) from scratch (Croitoru et al., 2022; Kim and Lee, 2023; Zhang et al., 2022b), or fine-tuning speech foundational models (Wagner et al., 2023) such as WavLM (Chen et al., 2022), HuBERT (Hsu et al., 2021), and wav2vec2.0 (Baevski et al., 2020) for the downstream classification task (Chen et al., 2024; Feng and Narayanan, 2023; Lashkarashvili et al., 2024). In particular, each of the models accepts a representation of the audio signal (either the raw signal or a Log-Mel Spectrogram) and is trained to predict the correct emotion from a finite set. However, in relation to the CI literature, these works generally have two limitations. Firstly, many methods claiming *supposed* SoTA performance on SER datasets fail to account for the fact that the underlying audio data is imbalanced (Croitoru et al., 2022; Kim and Lee, 2023; Zhang et al., 2022b). This raises significant doubt about whether the models have truly learned a good representation of the minority class data, or have simply learned to reflect the majority class. The second limitation is that when the imbalance is taken into account, classification is not the downstream task (Li et al., 2021; Wagner et al., 2023). It is instead a regression task to predict emotional attributes such as arousal, valence and dominance, or the attention of the work is not focused solely on the problem of CI (Feng and Narayanan, 2023; Lashkarashvili et al., 2024). Chen et al. (2024) address both of these in their submission for the Odyssey Emotion Challenge[4] submission. However, their work did not cover a broad sweep over the CI literature but focused only on inverse re-weighting and focal loss. While our main contribution is the novelty of second-order optimisers as CI correctors, we additionally perform a rigorous sweep across commonly used imbalance-correction methods on the SER task to assist future work in this domain.

Summary: In this chapter, we provided the necessary background needed to understand the work presented in this thesis. We provided an overview of various methods used to counteract the CI problem. Additionally, we introduced second-order optimisers, specifically focusing on the approximate NGD optimisers that will be applied in the context of this thesis. Finally, we provided an overview of how the SER task has been tackled thus far, and their limitations related to the CI problem. In the next chapter, we focus our attention on providing a unified mathematical framework across which we can understand both algorithm-level methods and second-order optimisers as CI correctors.

[4]https://www.odyssey2024.org/emotion-recognition-challenge

Chapter 3

Mathematical Preliminaries

In this chapter, we elucidate many of the mathematical details behind this thesis. In particular, we provide a *unified* framework to understand how CI affects *learning*, where we show how the various baseline methods that we consider in this thesis can be understood therein. Finally, we interpret second-order optimisers through this lens, and demonstrate their benefit for CI.

3.1 Introducing Supervised Learning

In supervised learning, we assume that we have some data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ that we wish to model, where each d-dimensional sample $\mathbf{x}_n \in \mathbb{R}^d$ has an associated label $y_n \in \{c | c = 1, 2, \dots, \mathcal{C}\}$ from a set of possible classes $\mathcal{C}$. Additionally, we denote that each class c has n_c samples allocated to it, comprising a proportion of $\pi_c = \frac{n_c}{N}$ of the total dataset. We assume that we have a model f parameterised by $\boldsymbol{\theta}$ that is trained to predict y_n given $\mathbf{x}_n$. Namely, $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^{\mathcal{C}}$, where $\mathbf{z}_n = f_{\boldsymbol{\theta}}(\mathbf{x}_n)$ where $\mathbf{z}_n \in \mathbb{R}^{\mathcal{C}}$ are the logits for each class. Applying a softmax activation to the logits, the model defines a categorical probability distribution on the output space $p_{\boldsymbol{\theta}}(y_n | \mathbf{x}_n)$ where $p_{\boldsymbol{\theta}}(y_n = c | \mathbf{x}_n) = \frac{e^{z_n^c}}{\sum_{c'=1}^{\mathcal{C}} e^{z_n^{c'}}} = \hat{y}_n^c$. Assuming that our N training data points are *i.i.d*, we train the model by minimising the cross-entropy loss as is standard in the literature, where y_n^c is an indicator variable:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{\mathcal{C}} y_n^c \log \hat{y}_n^c \tag{3.1}$$

Using first-order optimisers, the parameters $\boldsymbol{\theta}$ are optimised iteratively using gradient descent algorithms (Kingma and Ba, 2017; Robbins and Monro, 1951) which update the parameters in the direction resulting in the largest change in loss per unit change in the

parameters, as measured by the Euclidean distance (Martens and Grosse, 2020), where η is the learning rate.

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{3.2}$$

As has been shown by Francazi et al. (2024) and Kunstner et al. (2024), this training paradigm assumes that the representation across classes in the training data is roughly equal; in the presence of CI, the optimisation dynamics become increasingly difficult. Below, we will illustrate this in the case of a simple linear model inspired by the derivations of Kunstner et al. (2024).

3.2 The Learning Dynamics of Class Imbalance

At a high level, CI disrupts the dynamics of optimisation when training a model (Francazi et al., 2024; Kunstner et al., 2024). More specifically, the gradients (and the norms thereof) of the majority class far exceed those of the minority classes. As such, the learning of the minority classes proceeds far slower. To show this effect, we take a linear model where $f_{\boldsymbol{\theta}}(\mathbf{x}) = W\mathbf{x} = \mathbf{z}$ where $\boldsymbol{\theta} = \texttt{vec}(W)$. While we focus on GD in this case, we can make similar arguments for stochastic versions, where data can be translated to analogous batch-level quantities. That being said, the effect of CI may be more severe in the stochastic case as the minority samples may not appear in the batch (see Francazi et al. (2024) for a more in-depth discussion on this point). It can be shown that the gradient of the loss w.r.t $\mathbf{w}_c$ (full derivation in Appendix A.2):

$$\nabla_{\mathbf{w}_c} \mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} [\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n \tag{3.3}$$

Expanded further, the first derivative can be re-written a sum of per-class components, where ■ indicates the c^{th} class component:

$$\nabla_{\mathbf{w}_c} \mathcal{L}(\theta) = \frac{\pi_c}{n_c} \sum_{n=1:y_n=c}^{N} [\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n + \sum_{j=1:j\neq c}^{\mathcal{C}} \frac{\pi_j}{n_j} \sum_{n=1:n=j}^{N} (\hat{y}_n^{y_n})\mathbf{x}_n \tag{3.4}$$

Taking the average across all data points in a class where $\bar{\mathbf{x}}^c = \frac{1}{n_c}\sum_{n=1:y_n=c}^{N} \mathbf{x}_n$ and assuming $\hat{y}_n^{y_n} = p$ is the same for all data points:

$$\nabla_{\mathbf{w}_c} \mathcal{L}(\theta) = \underbrace{\pi_c}_{\text{Data Proportion}} \cdot \underbrace{[p-1]\bar{\mathbf{x}}^c}_{\text{Data fit}} + \sum_{j=1:j\neq c}^{\mathcal{C}} \frac{\pi_j}{n_j} \sum_{n=1:n=j}^{N} (\hat{y}_n^{y_n})\mathbf{x}_n \tag{3.5}$$

In Eq. 3.5, we clearly observe that the per-class gradient magnitude is determined by two components: the data proportion (**C1**) and data-fit (**C2**). Focusing on the data proportion component: in the case that $\pi_c >> \pi_j$ and where the predictions for the other classes are near random, we find that the c^{th} class component will dominate the gradient update through its greater relative representation. How this effects the overall gradient update step is illustrated in Figure 3.1. In the imbalanced case, the direction and magnitude of the gradient update, $\nabla\mathcal{L}(\boldsymbol{\theta})$, is biased towards the majority-class component $\nabla_{\mathcal{C}_1}\mathcal{L}(\boldsymbol{\theta})$, whilst there is little to no contribution from the minority-class component $\nabla_{\mathcal{C}_2}\mathcal{L}(\boldsymbol{\theta})$. When adding two vectors together, vectors with larger magnitudes will dominate the update. Benzing (2022) showed for stochastic optimisers such as SGD (Robbins and Monro, 1951), this effect occurs as the averaging scheme of SGD is not, inherently, able to deal with the conflicting gradient directions. In the balanced (or if gradient magnitudes are corrected as we will see below) case, we find that there is an equal contribution from each class, and the gradient update takes a step that is favourable to both classes. In this case, there are no conflicting gradient directions as the vectors from both classes have equivalent norms. As such, the model is able to make progress on both classes, where differences in performance can attributed to the inter-class difficulty itself, rather than on poor representation (Francazi et al., 2024).

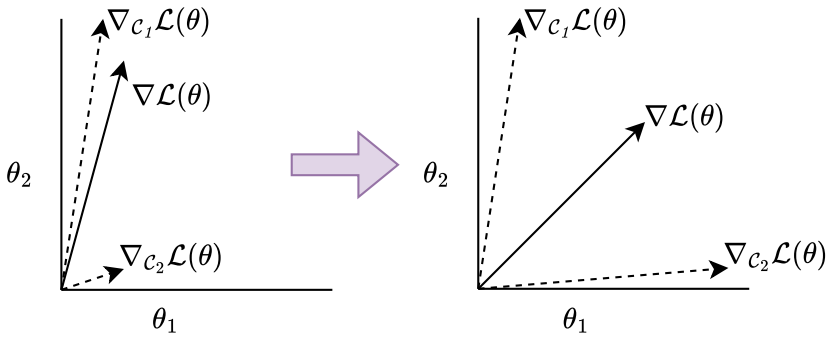


Fig. 3.1 Viewing $\nabla\mathcal{L}(\theta)$ as the sum of per-class vector components in the imbalanced (**Left**, where $\mathcal{C}_1$ is the majority class) and balanced (**Right**) training data in a two class problem. Inspired by Anand et al. (1993).

Addressing the data fit term, related to the saturation regions of the softmax non-linearity: as the majority class samples are better classified, the per-class gradient norm tends to zero. This provides evidence as to why data complexity is important for CI: if the underlying data is too simple, the model can simply learn a good representation across all classes (Japkowicz, 2000; Johnson and Khoshgoftaar, 2019). The gradient magnitudes for the majority class will

be suppressed, allowing the minority classes to contribute. However, this condition is not readily met in more difficult problems, which is the focus of our thesis.

3.3 How to Correct the Dynamics?

As introduced in Chapter 2.1.2, algorithm-level methods correct for CI through a cost-sensitive approach. We can interpret these methods as *first-order* corrections, in the sense that they directly modify the gradient update by a scalar factor, and do not incorporate any second-order information. We provide a unified mathematical framework to understand each of these methods below. Focusing on supervised classification, the family of cost-sensitive approaches corrects the standard cross-entropy loss (Eq. 3.1) as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{\mathcal{C}} \boldsymbol{\alpha_n^c}\, y_n^c \log \hat{y}_n^c \tag{3.6}$$

where the class-dependent weighting factor is $\boldsymbol{\alpha_n^c}$[1] indicated by ▬. This stabilises the learning dynamics as this correction is propagated to the gradient level where Eq. 3.3 and Eq. 3.4 are corrected in Eq. 3.7 and 3.8, respectively:

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\alpha_n^c}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n \tag{3.7}$$

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \frac{\pi_c}{n_c}\sum_{n=1:y_n=c}^{N} \boldsymbol{\alpha_n^c}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n + \sum_{j=1:j\neq c}^{\mathcal{C}} \frac{\pi_j}{n_j}\sum_{n=1:n=j}^{N}(\hat{y}_n^{y_n})\mathbf{x}_n \tag{3.8}$$

Assuming that $\boldsymbol{\alpha_n^c}$ is constant for all members of the class (indeed in some cases (Huang et al., 2016; Mahajan et al., 2018; Mikolov et al., 2013; Wang et al., 2017)), we find that Eq 3.5 becomes:

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \boldsymbol{\alpha_n^c}\cdot \underbrace{\pi_c}_{\text{Data Proportion}} \cdot \underbrace{[p-1]\bar{\mathbf{x}}^c}_{\text{Data fit}} + \sum_{j=1:j\neq c}^{\mathcal{C}} \frac{\pi_j}{n_j}\sum_{n=1:n=j}^{N}(\hat{y}_n^{y_n})\mathbf{x}_n \tag{3.9}$$

As seen in Eq. 3.9, in the case where $\pi_c >> \pi_j$, given an appropriate choice of $\boldsymbol{\alpha_n^c}$, the signal of the majority class is attenuated such that each class has a roughly equal contribution to the gradient update (**Right** in Fig. 3.1). Referring to the data fit term, we see that the particular choice of correction factor can modulate the data fit term, allowing the model to focus on

[1] The factor is in bold for emphasis; it is a scalar.

examples that have been poorly classified during training. These corrections, therefore, allow the model to learn a better representation across all classes, and not just the majority, or easily-classifiable samples.

3.4 Choosing the Loss-Level Correction Factor

In line with the above framework, we introduce the various baseline methods that we consider for this thesis, and how the adjustments they make can be viewed as a different choice of the correction factor $\alpha_{\mathbf{n}}^{\mathbf{c}}$.

Baseline training uses the cross-entropy loss naively as in Eq. 3.1, without taking into account that there is a CI in the training data.

$$\alpha_n^c = 1$$

Re-weighting (Huang et al., 2016; Wang et al., 2017) uses the inverse class frequency to calculate the per-class cost. The inverse frequency is normalised to ensure that the class weights sum to one in the batch on average.

$$\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{\mathcal{C}} n_c^{-1}}$$

This is a global correction as it addresses the gradient norm issue in Eq. 3.9 by removing the influence of the data proportion term (**C1**).

Class-Balance (Cui et al., 2019) builds on the aforementioned re-weighting method, but allows the class weight to exist on a continuum from no correction to the the inverse class frequency, controlled by a hyperparameter β.

$$\alpha_n^c = \frac{1-\beta}{1-\beta^{n_c}}$$

Like re-weighting, this method is a global correction on the data proportion term (**C1**).

Focal Loss (Lin et al., 2018) is a local (per-sample) performance-based correction mechanism for CI training calculated as follows:

$$\alpha_n^c = (1 - \log p_\theta(y_n = c|\mathbf{x}_n))^\gamma$$

Observing Eq. 3.9, focal loss addresses the CI problem through applying an additional modulation factor to the data fit term (**C2**). This suppresses the gradient norm by increasing

the saturation rate of this factor without *explicit* knowledge of the global class-level imbalance; it focuses training on hard examples, which are by proxy minority-class samples.

α-Weighted Focal Loss is an extension to Focal Loss (Lin et al., 2018) inspired by Cui et al. (2019)'s work which now includes the global class-level imbalance explicitly. Specifically, this variation uses the inverse class frequency re-weighting scheme (Huang et al., 2016; Wang et al., 2017) which suppresses the data proportion term (**C1)** in Eq. 3.9 along with the data fit term (**C2**) already provided by Focal Loss.

$$\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{\mathcal{C}} n_c^{-1}}(1 - \log p_{\boldsymbol{\theta}}(y_n = c|\mathbf{x}_n))^{\gamma}$$

LDAM (Cao et al., 2019) introduces a loss than can be used for classification tasks instead of cross-entropy. LDAM is label-aware, which means that it applies larger margins to minority classes.

$$\mathcal{L}_{LDAM}(\boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{c=1}^{\mathcal{C}}[y_n^c \log \tilde{y}_n^c] \text{ and } \tilde{y}_n^c = \frac{e^{z_n^c - \Delta_c}}{e^{z_n^c - \Delta_c} + \sum_{c'=1, c' \neq c}^{\mathcal{C}} e^{z_n^{c'}}} \text{ and } \Delta_c = \frac{C}{n_c^{\frac{1}{4}}}$$

As mentioned in Chapter 2.1.2, LDAM provides a more involved regularisation to the gradient relative to the scalar adjustment used by other baselines considered herein. Particularly, this is because Δ_c is applied at the logit level - this cannot be extracted out of the softmax non-linearity. However, we can still consider this as a correction to the standard cross-entropy (Eq. 3.1) that tackles the data fit term (**C2**), but not the data proportion (**C1**).

LDAM-DRW builds on LDAM (Cao et al., 2019) by introducing a term that provides *explicit* knowledge of the class-level imbalance. However, Cao et al. (2019) showed that introducing this information is helpful, albeit not at the start of training. In particular, they make use of a **d**eferred **r**e-**w**eighting (**DRW**) scheme where training is split into two potions. In the first part, the network is trained using only the LDAM loss for 50% of the total training time. After this point, the model is trained on a re-weighted loss using the inverse class frequency which addresses the data proportion (**C2**).

$$\mathcal{L}_{LDAM-DRW}(\boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{c=1}^{\mathcal{C}}\alpha_n^c[y_n^c \log \tilde{y}_n^c] \text{ and } \alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{\mathcal{C}} n_c^{-1}}$$

Influence Balanced (Park et al., 2021), corrects each sample by the inverse of its influence (Markatou and Ronchetti, 1997; Robinson, 1984). In particular, they show that the influence of a sample is given by $\mathcal{I}(\mathbf{x}, \boldsymbol{\theta}) = -H^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(y, f_{\boldsymbol{\theta}}(\mathbf{x}))$ where $H = \frac{1}{N}\sum_{n=1}^{N}\nabla_{\boldsymbol{\theta}}^2\mathcal{L}(y, f_{\boldsymbol{\theta}}(\mathbf{x}))$.

However, since evaluating the entire Hessian would be too computationally expensive for large neural networks as we will consider in this thesis, Park et al. (2021) instead approximate the conditioning factor to be $\frac{1}{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(y,f_{\boldsymbol{\theta}}(\mathbf{x}))\|_1} \approx \frac{1}{\|f(\mathbf{x}_n;\boldsymbol{\theta})-\mathbf{y}_n\|_1 \cdot \|\mathbf{h}_n\|_1}$, where they only focus on the last fully connected layer of the neural network in the case of cross-entropy loss. This conditioning factor addresses the data fit (**C2**), where there is an additional inverse class frequency included as a re-weighting factor to address the data proportion (**C1**), along with an additional hyperparameter ζ [2].

$$\alpha_n^c = \frac{\zeta \cdot n_c^{-1}}{\sum_{c=1}^{\mathcal{C}} n_c^{-1} \cdot \|f(\mathbf{x}_n;\boldsymbol{\theta}) - \mathbf{y}_n\|_1 \cdot \|\mathbf{h}_n\|_1}$$

Similar to LDAM-DRW (Cao et al., 2019), the influence balanced re-weighting is only applied in the second half of training, where the network is initially trained using the standard cross-entropy loss.

3.5 Enter Second-Order Optimisation

Instead of performing first-order gradient descent as in Eq. 3.2, second-order optimisers precondition the gradient update such that:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \underbrace{\eta F^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})}_{\text{Descent direction } \Delta\boldsymbol{\theta}_{Fisher}} \tag{3.10}$$

where F is a matrix of a particular form such as the Hessian or approximations thereof. For this thesis, our focus is on the Fisher, where

$$F := \sum_{n=1}^{N} \mathbb{E}_{y\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_n)}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y|\mathbf{x}_n) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y|\mathbf{x}_n)^\top] \tag{3.11}$$

where y is a sample from the model's output distribution. The descent direction $\Delta\boldsymbol{\theta}_{\text{Fisher}}$ is now given by the largest change in the loss per unit of change in the parameters as per the KL-divergence (Martens and Grosse, 2020). As pointed out by Wu et al. (2024b) and Kunstner et al. (2020), this can be estimated by Monte-Carlo Sampling (Martens and Grosse, 2020), or it can be approximated further as:

$$\tilde{F} := \sum_{n=1}^{N} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y_n|\mathbf{x}_n) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y_n|\mathbf{x}_n)^\top \tag{3.12}$$

[2] This was α in the original paper; we have changed the symbol to avoid notation overloading.

where now both $\mathbf{x}_n$ and y_n are empirical samples (i.e. the data we have access to). The above approximated matrix will henceforth be referred to as the empirical Fisher (EF) (Kunstner et al., 2020; Wu et al., 2024b). Wu et al. (2024b) showed that the $P \times P$ EF matrix in Eq. 3.12 can be reformulated as:

$$\tilde{F} := \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell} \tag{3.13}$$

where $\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}$ is the $N \times P$ Jacobian matrix of the per sample loss *w.r.t* model parameters. Practically speaking, a damping factor λ is added to the diagonal of the Fisher matrix to facilitate inversion in large models (Wu et al., 2024b), which results in the descent direction $\Delta\boldsymbol{\theta}_{EF}$:

$$\begin{gathered} \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \underbrace{\eta(\tilde{F} + \lambda \mathbb{I})^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})}_{\Delta\boldsymbol{\theta}_{EF}} \\ \text{where } \Delta\boldsymbol{\theta}_{EF} = -\eta \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top (\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell} \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top + \lambda \mathbb{I})^{-1} \mathbf{1} \end{gathered} \tag{3.14}$$

by the Woodbury identity (Petersen and Pedersen, 2012; Wu et al., 2024b). A particular property of second-order optimisers that we wish to explore is that they ensure that the parameter update update ensures a loss reduction on every involved sample, which is not the case for SGD (Benzing, 2022; Wu et al., 2024b). In particular, the loss induced on every sample by the use of the EF, shown by Wu et al. (2024b), is:

$$\Delta\boldsymbol{\ell}_{EF} = -\eta \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell} \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top (\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell} \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top + \lambda \mathbb{I})^{-1} \mathbf{1} \approx -\eta \mathbf{1} \tag{3.15}$$

provided that the Jacobian is full row rank and η and λ are negligible. This allows the second-order optimiser to resolve the conflicting gradient directions, as illustrated in Figure 3.2. However, the issue with the above, as pointed out by Kunstner et al. (2020) and elucidated by Wu et al. (2024b), is that the magnitude of the "new" updates (in the direction determined by EF) are inversely proportional to the magnitude of the gradient for that sample (referred to the *inversely-scaled projection issue* (Kunstner et al., 2020; Wu et al., 2024b) seen in Fig. 3.2). This means that the optimiser is inherently biased towards well-trained samples (which have smaller gradient norms) as seen below:

$$(\kappa_n)_{EF} = \Delta\boldsymbol{\theta}_{EF}^\top \frac{\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}_n}{\|\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}_n\|_2} = -\frac{\eta}{\|\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}_n\|_2} \tag{3.16}$$

To correct for this, Wu et al. (2024b) introduced the *improved empirical Fisher* (iEF) which re-scales the projection with the logit-level gradient norm. The projection is now aligned

with how well each sample is fit.

$$(\kappa_n)_{iEF} - \frac{\eta \|\nabla_{z_n} \ell_n\|_2^2}{\|\nabla_{\boldsymbol{\theta}} \ell_n\|_2} \tag{3.17}$$

which redefines Eq. 3.14 as:

$$\begin{aligned} \Delta\boldsymbol{\theta}_{EF} &= -\eta \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top (\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell} \nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}^\top + \lambda \mathbb{I})^{-1} \mathbf{s}_{iEF} \\ \text{where } \mathbf{s}_{iEF} &= [\|\nabla_{z_1} \ell_1\|_2^2 \cdots \|\nabla_{z_n} \ell_n\|_2^2] \end{aligned} \tag{3.18}$$

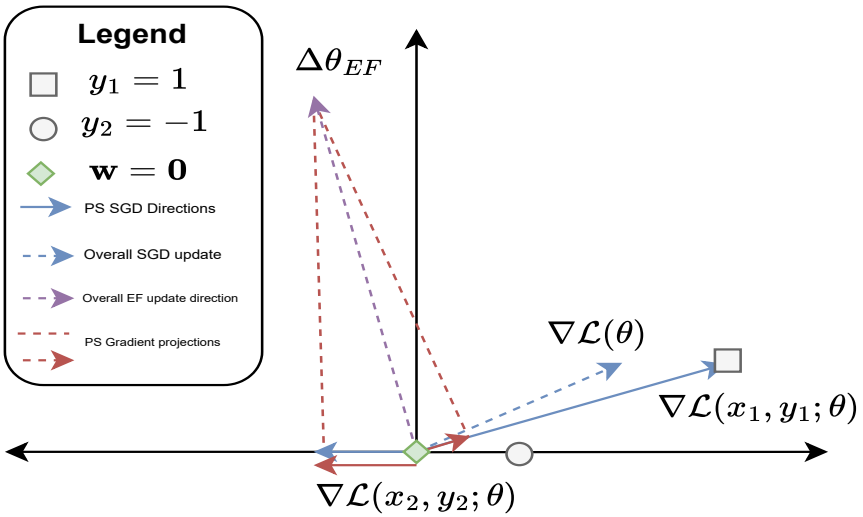


Fig. 3.2 Visualisation of how second-order optimisers act as a local CI correction inspired by Benzing (2022). PS = Per Sample. Like in Figure 3.1 (**Left**), we see that the SGD update (blue vector) is biased towards the first data point as this sample has a larger gradient norm. The gradient update will increase the loss of the second data point. However, the EF gradient direction decreases the loss on both samples as shown by the purple vector; the loss induced per sample is normalised by the gradient norm. While this is on a per-sample level, we can view this as a local correction mechanism that re-normalises conflicting gradient directions analogously to Figure 3.1 (**Right**), allowing the model to learn an adequate representation for both samples

Connecting this to our unified mathematical framework, second-optimisers use Eq. 3.16 and Eq. 3.17 as the analogous quantity to the correction scalar α_n^c which we visualise in Fig. 3.2. In particular, this can be viewed as a correction term addressing the data fit (**C2**) through the normalising the loss by how well the sample is fit in the projected direction. As such, we see that second-order methods can be thought of as imbalance-correction methods that have been applied at the gradient step level, rather than at the loss level, as has been done in the literature previously. Inspired by this theoretical backing, we posit that iEF (Wu et al., 2024b)

and equivalent methods (Benzing, 2022; George et al., 2018), with the equal loss reduction property and corrected projection, will be able to alleviate the difficulties in the learning dynamics caused by the CI problem. In particular, it will stabilise the learning dynamics, allowing the model to make progress both on majority and minority samples during training.

Remark: We wish to point out the inherent similarity between Influence Balance and second-order optimisers. The definition of the influence function, in a sense, can be viewed as a preconditioned gradient step for a single sample (as would be achieved during training with our second-order optimisers), where the Hessian has a very similar definition to our EF matrix. In fact, the Fisher is considered to be an approximation of the true Hessian (Kunstner et al., 2020). Furthermore, in the case of the true EF projection (George et al., 2018) in Eq. 3.16, the loss per-sample is normalised by the norm of the sample's gradient. This is practically the same adjustment made by Influence Balanced (Park et al., 2021). In light of this, we posit that Influence Balance succumbs to the same *inverse-scaling projection issue* (Kunstner et al., 2020; Wu et al., 2024b); normalising the loss of each sample by $\frac{1}{\|f(\mathbf{x}_n;\boldsymbol{\theta})-\mathbf{y}_n\|_1 \cdot \|\mathbf{h}_n\|_1}$ causes the gradients to be biased towards well-trained samples as in Eq. 3.16.

Summary: In this chapter, we provided a *unified* mathematical framework through which we can understand how the various methods presented in this thesis account for the CI problem. In particular, we show the propagation of the re-weighting factor applied at the loss level down to the gradient, allowing for more stable optimisation dynamics across all classes. In effect, this reverts the dynamics to the case where the class representation was balanced. Therein, we showed how each of the baseline methods that we consider throughout this thesis can be interpreted as a particular choice of the correction factor. Finally, we provided a mathematical overview for second-order optimisers, and how, aligned with the framework, they can be interpreted as a local corrective mechanism in the case of CI. In the next chapter, we provide the experimental setup, and the associated results, that we use to verify the theoretical claims that we have set out thus far.

Chapter 4

Experiments

In this chapter, we provide a thorough outline of the experimental setup used for this thesis and the results themselves. In particular, we provide the design decisions with respect to the dataset, model architectur and the evaluation metrics. We conclude by reporting our findings, with some high-level insights.

4.1 Setup

As mentioned in Chapter 3, the focus of this thesis is on the supervised classification task where there the data suffers from a CI. Specifically, we train all our models on training sets that have varying degrees of imbalance[1], where we wish to evaluate the model's performance on both majority and minority classes on the held-out test data.

4.1.1 Datasets

We focus on the following datasets for our experimentation, where all data has been standardised prior to processing. We quantify the amount of imbalance on each dataset by $p = \frac{\max_c n_c}{\min_c n_c}$ where n_c indicates the number of samples in class c.

CIFAR-10 (Krizhevsky, 2009) is a dataset typically used as a benchmark for computer vision tasks which features 50,000 training, and 10,000 test examples of 32x32 colour images. The images in both datasets are divided equally among ten classes, examples of which can be seen in Fig. 4.1. From the original training set, we reserve 10% as validation data, where the remaining 90% acts as the true training set. Since all data partitions are perfectly balanced (i.e. $p = 1$), we imbalance it manually. In particular, we define a symmetric distribution

[1] Corrections for said imbalance may have been applied

$Dir(\alpha)$ parameterised by the concentration parameter α over the ten classes. We then draw a sample $\boldsymbol{\gamma} \sim Dir(\alpha)$ where $\lfloor \boldsymbol{\gamma}_c * n_c \rfloor = n_c^*$ indicates the new number of samples belonging to the class c. Furthermore, we ensure that $n_c^* > 100$ for all classes to ensure that each class has some lower bound of representation in the dataset. α acts as a proxy for the amount of imbalance: as $\alpha \to \infty$, the Dirichlet distribution tends to a uniform distribution, which means that $\boldsymbol{\gamma}_c = \boldsymbol{\gamma}_j, \forall j \in \mathcal{C}$ in the limit, meaning that $p \to 1$. As $\alpha \to 0$, $p \to \infty$ as the distribution tends to a δ function over a particular class c, which will be the majority. We consider $\alpha \in \{0.1, 1, 10\}$, and the original balanced CIFAR-10 training dataset in particular, visualised in Fig. 4.2, where all dataset statistics can be seen in Table B.1. The reason behind this splitting strategy is that we want to test across a wide range of imbalance levels, where the type of imbalance is akin to that of CREMA-D and MSP-Podcast. Furthermore, as described in Chapter 1, we wish to focus on the particular form of CI where good performance across all classes is essential, rather than prioritising certain majority classes.

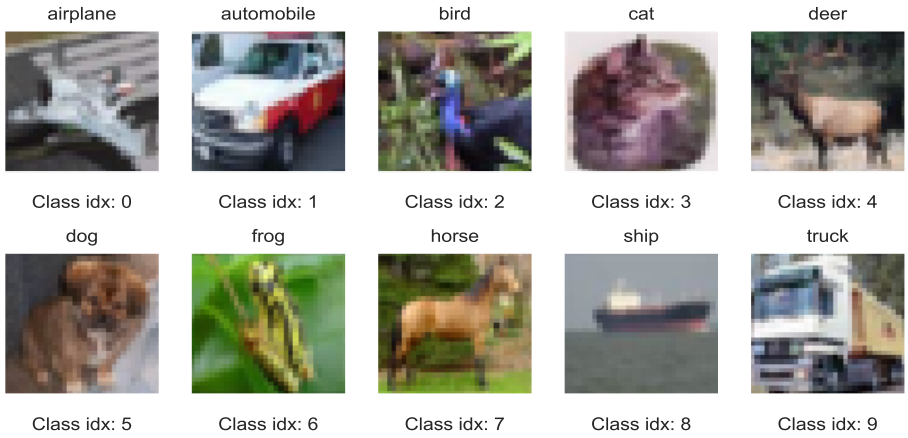


Fig. 4.1 Examples from the training set for each of the ten CIFAR-10 classes.

CREMA-D (Cao et al., 2014) is a speech emotion dataset that contains 7,442 clips from 91 professional actors. In particular, these actors have been tasked with demonstrating a prototypical expression for a selection of 12 predefined sentences in a particular emotion and at a particular emotional level/intensity. The dataset, when created, was *balanced* across the *intended* expressed emotions: anger, disgust, fear, happiness, neutral and sad. However, when Cao et al. (2014) assigned human annotators to evaluate the perceived emotion, there was a large degree of inter-annotator disagreement with the intended emotions. In particular, many of the example recordings were assigned to be neural, rather than their indented class. As such, we use the majority *human-agreed* emotion as the label rather than the *intended* emotion. In the case where there is a tie across the human-agreed emotion, we did not include these samples in our training data as they led to an unstable training setup. This left the six

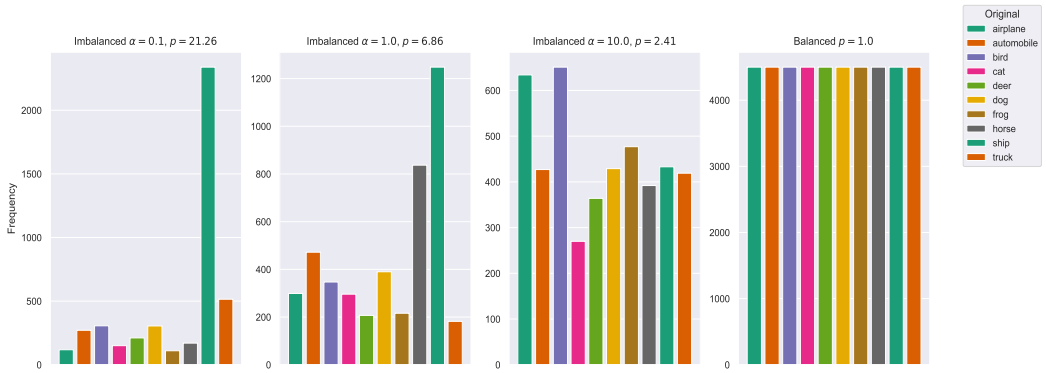


Fig. 4.2 Visualisation of the four variations of the CIFAR-10 dataset used for experimentation. We observe the effect of there being more imbalance as $\alpha \to 0$, confirmed by $p = \frac{\max_c n_c}{\min_c n_c}$.

emotion classes as per Table B.2. This is unlike (Chen and Rudnicky, 2023; Feng et al., 2022; Feng and Narayanan, 2023; Pepino et al., 2021) who use the four most frequently occurring emotions; we posit that our six-way classification task is more realistic and thus more difficult. The data set was divided into train, validation, and test in the ratio $70/10/20$, ensuring that the samples of each speaker occur in a single split to prevent information leakage (Kapoor and Narayanan, 2022; Nisbet et al., 2018). The visualisation of the dataset CI is seen in Fig. 4.3, where `Neutral` is the majority class. For the purposes of this thesis, whilst CREMA-D (Cao et al., 2014) is a multimodal dataset, we only consider the *audio* samples as input.

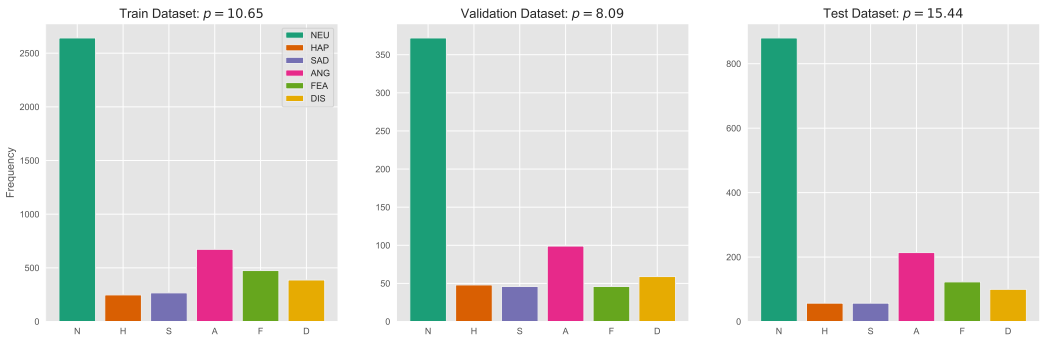


Fig. 4.3 Distribution of samples across the six emotion classes in CREMA-D (Cao et al., 2014) for each data partition with amount of imbalance p indicated.

MSP-Podcast (Lotfian and Busso, 2019) is another speech emotion dataset that aims to build on the work of Cao et al. (2014). In particular, an issue with the CREMA-D (Cao et al., 2014) dataset is that the emotional expressions displayed by the actors are prototypical. They do not generalise well to real, nuanced emotions (Lotfian and Busso, 2019). MSP-Podcast (Lotfian and Busso, 2019) aims to provide natural examples of emotion which are gathered

from podcast data, and annotated using crowdsourcing techniques. In this thesis, we use version 1.10 which contains ∼166 hours of audio[2] (De Oliveira et al., 2023), where each of the audio samples is assigned to one of ten classes. We make use of the suggested partitioning scheme by Lotfian and Busso (2019) to ensure that the datasets are speaker-independent. This results in four data partitions: a train and validation set, along with two test sets with differing levels of imbalance. To create a similar experimental setup to the CREMA-D dataset, we

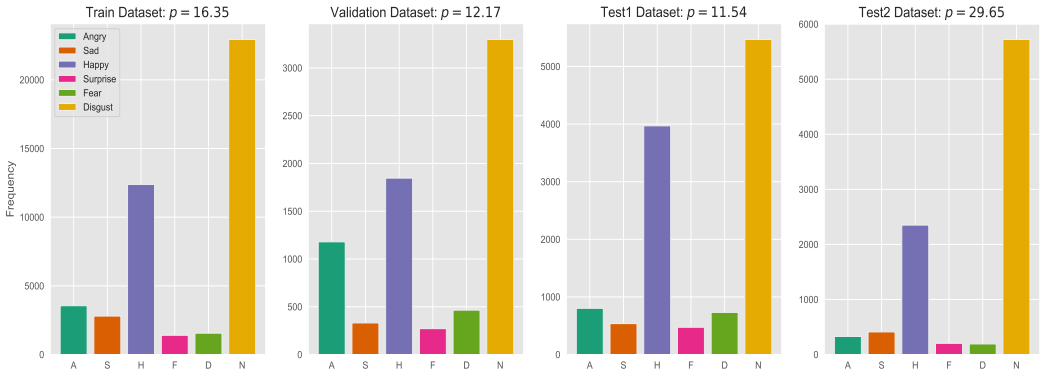


Fig. 4.4 Distribution of samples across the six emotion classes in MSP-Podcast (Lotfian and Busso, 2019) for each data partition with amount of imbalance p indicated.

consider a six-way classification for this thesis, where the classes were chosen to overlap with those in CREMA-D as seen in Table B.3. This is done instead keeping the four most frequent emotions (Chen and Rudnicky, 2023; Feng et al., 2022; Feng and Narayanan, 2023; Pepino et al., 2021) for the same reasons as with CREMA-D. We posit that this experimental design will allow us to make more accurate claims about the increase in stability moving from a small training set in CREMA-D to MSP-Podcast.

4.1.2 Model Architectures

CIFAR-10

The CIFAR-10 (Krizhevsky, 2009) experiments use a VGG model (a form of deep convolutional neural network) originally developed by Simonyan and Zisserman (2015). In particular, we choose the 11 layer version with batch norm (Ioffe and Szegedy, 2015) [3], although scaling the model for suitability with CIFAR-10 (seen in Fig. 4.5) (Krizhevsky, 2009) as opposed to ImageNet. For example, the final fully-connected layer in the network is reduced from 1000

[2]This is considerably larger than CREMA-D, addressing another limitation.

[3]We point the reader to view Configuration A in Simonyan and Zisserman (2015) for additional model details.

dimensions to 10 to match the CIFAR-10 (Krizhevsky, 2009) classification task. The model itself has $28,149,514$ trainable parameters which we train from scratch.

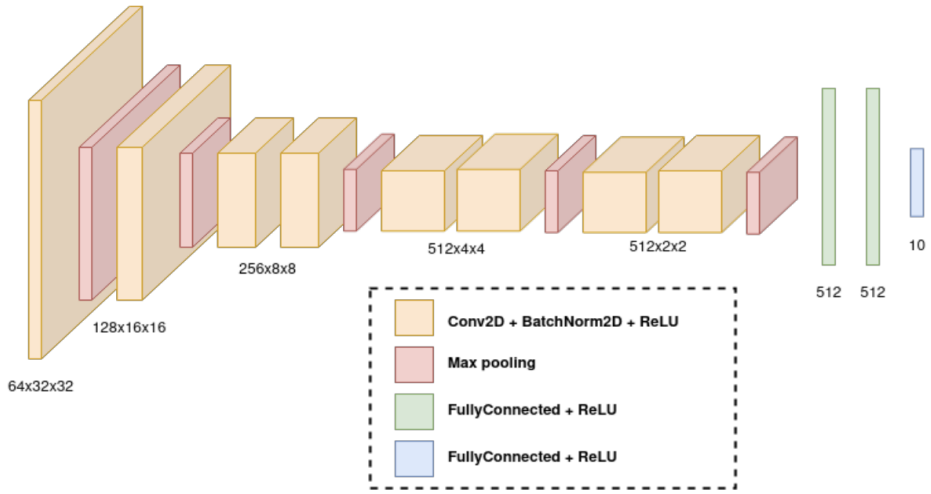


Fig. 4.5 Diagram illustrating the architecture of the `VGG11_bn` model used for the CIFAR-10 (Krizhevsky, 2009) experiments. Image taken from Bezzam et al. (2022)

SER

For both the CREMA-D (Cao et al., 2014) and MSP-Podcast (Lotfian and Busso, 2019) datasets, we make use of a pre-trained foundational model for speech, in particular HuBERT `Base`[4](Hsu et al., 2021), which we fine-tune for the downstream emotion classification task. This model was chosen as it provided good performance for SER on the SUPERB benchmark (Yang et al., 2021), whilst allowing for a relative reduction in the computational demands. In particular, the HuBERT model features 12 transformer layers, which has an embedding dimension of 768 along with 12 attention heads. Additionally, the output from HuBERT is fed into a projection matrix which reduces the dimensionality to 256, after which mean pooling is applied across the heads. Finally, we attach a classification layer to further project the hidden representations to the output space. Instead of fine-tuning the entire model, or specific layers, we opt to follow techniques from the parameter-efficient-fine-tuning literature (Feng and Narayanan, 2023; He et al., 2022; Houlsby et al., 2019). In particular, we use LoRA (Hu et al., 2021), which defines a low-rank bottleneck architecture to approximate the weight updates of a weight matrix $\mathbf{W}_{\text{New}} = \mathbf{W}_{\text{Old}} + \mathbf{W}_{\text{Down}}\mathbf{W}_{\text{Up}}$. Instead of fine tuning the full $d \times k$ matrix $\mathbf{W}$, we fine-tune the two low-rank, $d \times r$ and $r \times k$ matrices, $\mathbf{W}_{\text{Down}}$ and $\mathbf{W}_{\text{Up}}$, which is far more efficient. We apply LoRA to the query and value matrices of

[4]HuBERT `Base`

the 12 transformer layers in the HuBERT model, as recommended in the original work by Hu et al. (2021). Specifically, we use a rank $r = 8$ with LoRA $\alpha = 16$, as this was found to be performant empirically by Feng and Narayanan (2023). We leave the projection and classification layers to be fully trainable, resulting in 493,056 trainable parameters, 0.52% of the total 95,062,912 parameters. A figure describing the model can be seen in Fig. 4.6.

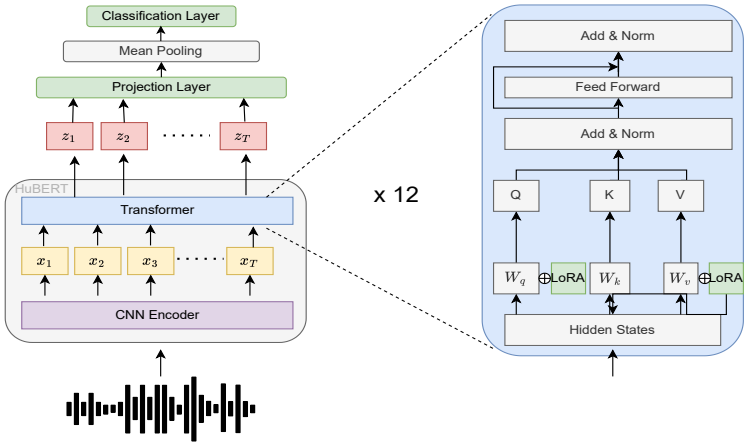


Fig. 4.6 Diagram illustrating the architecture of the `HuBERT` model used for the SER experiments. All parameters coloured in green are trained during the fine-tuning procedure, whilst everything else remains frozen.

Remark: We note that indeed it may be possible to drive up the performance of our model if it were replaced with a larger capacity (e.g. HuBERT `Large` or other speech foundation model variants (Chen et al., 2022; Radford et al., 2022)) model. However, the objective of this work is not to produce SoTA results on the SER task. Rather, we wish to highlight the importance of CI for the task, and provide a suite of easily-applicable correction methods that boost downstream performance. We posit that the conclusions brought by our work have cross-cutting applicability across model architectures, irrespective of their capacity.

4.1.3 Metrics of Interest

An important consideration in CI problems is the choice of metric used to report the final performance of the model. In particular, *accuracy* is not a suitable metric as it does not reflect the CI - the performance portrayed is dominated by samples from the majority class as this metric is class-agnostic (Johnson and Khoshgoftaar, 2019). Instead, more suitable metrics can be found in the *balanced accuracy* or *unweighted average recall* (UAR), and the *macro F1 score* (see Appendix A.3 for calculation details).

For the purposes of this thesis, we make use of all three metrics. Where we evaluate our models on balanced test sets, as in Table 4.2, we report only the accuracy. However, in the case of imbalanced test sets for our SER experiments, we report all aforementioned metrics. We show the *accuracy* to highlight how the supposed performance is a very misleading proxy for the true underlying decision-making of the model, whilst the *balanced accuracy* will serve to reflect a more objective view of the model's performance on both majority and minority classes. The macro F1 score is used to checkpoint the best performing models as in Chen et al. (2024), where we provide test macro F1 in Appendix B.3. For all metrics, the largest value is indicated by , the second largest by and the third largest by .

4.1.4 Training Recipes

To train our models, we will use gradient descent methods (Kingma and Ba, 2017; Robbins and Monro, 1951) to optimise our underlying model architectures, where the loss, unless explicitly stated otherwise, is quantified by the cross-entropy loss as described in Eq. 3.1.

For the **CIFAR-10** experiments, aligned with previous work (Cao et al., 2019; Cui et al., 2019; Park et al., 2021), we use SGD (Robbins and Monro, 1951) as our optimiser for the first-order baselines in Table 4.1. The learning rate has been selected on the best performing validation results for each method (see Appendix B.2 for more information on each baseline's hyperparameters). This is with the exception for second-order methods which use their respective optimiser. Additionally, no learning rate scheduler nor momentum term was used; this was done to remove any unnecessary confounders during our experimentation. The models are trained for 100 epochs across all experiment variations, where the best model is determined by the lowest loss on the validation set.

For both the **CREMA-D** and **MSP-Podcast** experiments, we use AdamW (Loshchilov and Hutter, 2018) as the optimiser for the first-order baselines in Table 4.1, where $\beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 1 \times 10^{-8}$ and the weight decay has been set to 0.1 for all experiments (except for second-order methods which use their respective optimiser). This is the typical optimiser choice for SER tasks (Chen et al., 2024; Feng and Narayanan, 2023). As in the CIFAR-10 experiments, we determine the optimal learning rate for each method based on their validation set performance. For all *first-order* correction experiments, a linear learning rate scheduler was used, as is typical for SER tasks (Chen et al., 2024; Feng and Narayanan, 2023); second-order optimisers do not use a scheduler as this was shown to be more performant empirically. All models were trained for 30 epochs on the **CREMA-D** data and for 20 epochs on the **MSP-Podcast** dataset. These design decisions were made based on the available computational resources. For the experiments on both dataset, the best

performing model was selected based on the best (highest) scoring macro F1 score as done by Chen et al. (2024).

4.1.5 Baseline Methods

For the purposes of this thesis, we consider the CI correction methods that we have introduced in Chapter 3.4 that will form our comparative baseline. These methods are either applied with, or replace, the traditional cross-entropy loss that is used for supervised learning. We provide a summary of the methods in Table 4.1, summarising the correction mechanism for each baseline for convenience. Furthermore, we provide a detailed elaboration of the various hyperparameter values used for each method in Appendix B.2.

Second-Order Optimisers: For the purposes of our experiments we consider two classes of second-order optimisers as introduced in Chapter 3.5. These optimisers will replace the first-order optimisers applied to the baseline methods for experimentation. Firstly, we wish to examine a second-order optimiser that exhibits the *inverse-scaling projection issue*: in our case, we will consider **EKFAC** (George et al., 2018). This will be evaluated on the CIFAR-10 experiments[5]. Additionally, we wish to use a second-order optimiser that has corrected the aforementioned issue. We will use (**iEF**) by Wu et al. (2024b) and the *Gradient Descent by Neurons* (**GDN**) algorithm (Benzing, 2022). These will be used for both the CREMA-D and MSP-Podcast experiments. For each experiment, we list the hyperparameters that were used for the optimisers in Appendix B.2, where we do not use any momentum as this provided better results empirically.

For our second-order optimisers, we additionally consider four sub-variants to create a thorough ablation over their effectiveness for CI problems.

> **Raw** - We apply the second-order optimiser directly to the underlying model without any changes to the underlying training mechanism.
>
> **Sampler** - Selecting samples for a mini-batch at random places no guarantee of on the type of samples that will be included. The batch could be made up of entirely majority samples due to the CI that occurs in the datasets that we consider for this thesis. To counter for this effect, we use a custom batch sampler to prescribe that each batch of samples includes minority samples in conjunction with the second-order optimiser.

[5]This decision is partially motivated by the fact that it is known that *iEF* (Wu et al., 2024b), currently, is unstable for configurations that involve *trainable* convolutional layers and batch normalisation. However, we still believe that the conclusions we draw in the CIFAR-10 experiments are valid and, theoretically speaking, *iEF* would only improve on EKFAC under good hyperparameter settings.

IRW - As was noted in many of the first-order corrections, addressing the data proportion term (**C1**) is crucial. We conduct experiments on whether this is useful for second-order optimisers by using **I**nitial **R**e-**W**eighting, where the inverse class frequency is applied to the cross-entropy loss at the beginning of training. This modifies the objective that the second-order optimiser will use to train the network.

DRW - As an alternative to IRW, we consider **D**eferred **R**e-**W**eighting, where the inverse class frequency is applied to the cross-entropy in the second half of training.

	Loss Correction	**GC**	**LC**	**DS**
Baseline	$\alpha_n^c = 1$	✗	✗	✗
Focal Loss (Lin et al., 2018)	$\alpha_n^c = (1 - \log p_\theta(y_n = c\|\mathbf{x}_n))^\gamma$	✗	✓	✗
α-Weighted Focal Loss (Lin et al., 2018)	$\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1}}(1 - \log p_\theta(y_n = c\|\mathbf{x}_n))^\gamma$	✓	✓	✗
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	$\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1}}$	✓	✗	✗
Class Balance (Cui et al., 2019)	$\alpha_n^c = \frac{1-\beta}{1-\beta^{n_c}}$	✓	✗	✗
LDAM (Cao et al., 2019)	$\mathcal{L}_{LDAM}(\boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{c=1}^{C}[y_n^c \log \tilde{y}_n^c]$ and $\tilde{y}_n^c = \frac{e^{z_n^c - \Delta_c}}{e^{z_n^c - \Delta_c} + \sum_{c'=1, c' \neq c}^{C} e^{z_n^{c'}}}$ and $\Delta_c = \frac{C}{n_c^{\frac{1}{4}}}$	✗	✓	✗
LDAM-DRW (Cao et al., 2019)	$\mathcal{L}_{LDAM-DRW}(\boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{c=1}^{C}\alpha_n^c[y_n^c \log \tilde{y}_n^c]$ and $\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1}}$	✓	✓	✓
Influence Balanced (Park et al., 2021)	$\alpha_n^c = \frac{\zeta * n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1} \cdot \|f(\mathbf{x}_n;\boldsymbol{\theta}) - \mathbf{y}_n\|_1 \cdot \|\mathbf{h}_n\|_1}$	✓	✓	✓
	Optimiser and Loss Correction	**GC**	**LC**	**DS**
Raw (Park et al., 2021)	$(\kappa_n)_{EF} = -\frac{\eta}{\|\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}_n\|_2}$ or $(\kappa_n)_{iEF} - \frac{\eta \|\nabla_{\mathbf{z}_n} \boldsymbol{\ell}_n\|_2^2}{\|\nabla_{\boldsymbol{\theta}} \boldsymbol{\ell}_n\|_2}$ and $\alpha_n^c = 1$	✗	✓	✗
Sampler (Park et al., 2021)	—"—and $\alpha_n^c = 1$	✗	✓	✗
IRW (Park et al., 2021)	—"—and $\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1}}$	✓	✓	✗
DRW (Park et al., 2021)	—"—and $\alpha_n^c = \frac{n_c^{-1}}{\sum_{c=1}^{C} n_c^{-1}}$	✓	✓	✓

Table 4.1 Summary of the various baseline methods introduced in this thesis. GC - Global correction, LC - local correction and DS - deferred scheme.

4.2 Results

In this section, we explore the experimental results for the various configurations that we have presented thus far in this thesis. In particular, all experiments are repeated across three random seeds to ensure that our results are significant. In the various tables, we report the mean metric, along with the standard error to show the variance in performance across runs.

4.2.1 CIFAR-10

In Table 4.2, we present the results for our experiments on the CIFAR-10 dataset. A first trend to notice is that with increased imbalance, the performance of all models globally tends to decrease as expected (Cao et al., 2019; Cui et al., 2019; Park et al., 2021). We were also able to replicate the results seen in (Cao et al., 2019; Cui et al., 2019; Park et al., 2021), where we find that Re-weighting, Class-Balance and Focal Loss all produced models that were

worse than the baseline in terms of their test performance. Observing the confusion matrices in Fig. C.1, while the aforementioned methods are able to boost the performance of the minority classes on average, this is done at the expense of modelling the majority data well. This is unlike performant methods such as Influence Balanced and EKFAC Raw which are able to prioritise both objectives. One unusual result that we obtained is that LDAM-DRW is worse than LDAM when $\alpha \in \{0.1, 1.0\}$: Cao et al. (2019) showed that LDAM-DRW in fact boosts performance relative to the LDAM baseline. We posit that this is due to the fact that we did not use a learning rate scheduler: Cao et al. (2019) necessitate that the re-weighted LDAM loss is applied with a smaller learning rate.

	Imbalanced CIFAR-10			**CIFAR-10**
Imbalance (α)	0.1	1.0	10.0	
Imbalance (p)	21.26	6.86	2.41	1.00
Baseline	62.33 ± 1.05	66.76 ± 0.45	69.99 ± 0.24	87.82 ± 0.65
Focal Loss (Lin et al., 2018)	61.43 ± 2.38	66.45 ± 0.03	70.37 ± 0.75	88.01 ± 0.15
α-Weighted Focal Loss (Lin et al., 2018)	57.88 ± 1.99	62.01 ± 2.82	66.80 ± 0.60	87.53 ± 0.37
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	60.34 ± 1.20	62.74 ± 1.64	68.01 ± 0.38	87.82 ± 0.65
Class Balance (Cui et al., 2019)	61.22 ± 1.05	64.10 ± 0.72	70.79 ± 0.28	87.82 ± 0.65
LDAM (Cao et al., 2019)	64.29 ± 0.48	63.47 ± 0.33	67.25 ± 0.83	88.81 ± 0.25
LDAM-DRW (Cao et al., 2019)	63.82 ± 0.80	62.94 ± 1.43	67.69 ± 0.40	88.81 ± 0.25
Influence Balanced (Park et al., 2021)	67.27 ± 0.32	69.57 ± 0.78	72.17 ± 0.30	88.53 ± 0.28
EKFAC Raw (George et al., 2018)	66.37 ± 0.52	67.45 ± 0.95	70.88 ± 0.63	88.66 ± 0.24

Table 4.2 Classification accuracy (%) of VGG-11 on imbalanced CIFAR-10 and CIFAR-10 datasets across three seeds for various corrections. Larger α indicates less imbalance.

Observing the performance of our second-order method, we see that across all levels of imbalance, simply optimising the model with EKFAC instead of using SGD with loss-level corrections is able to alleviate much of the difficulties associated with the CI problem. Not only does it beat the vast majority of the first-order methods, but it tends towards the performance of the Influence Balanced method, alluding to the similarity we showed in Chapter 3.5. This provides evidence in support of our **RQ.1** (Chapter 1.2), namely that second-order optimisers, with the local correction mechanism, would alleviate the optimisation difficulties associated with the CI problem. Through optimising an unmodified loss with a preconditioned gradient update[6], we are able to make better progress across all classes unlike in the naïve averaging that occurs with SGD. Using the second-order optimiser provides a somewhat simpler approach; we are able to alleviate the difficulties of the CI problem by choosing an adequate optimiser, rather than considering an array of loss-level corrections. This argument is expanded further in Chapter 5.4

However, while the significance of the result is unchanged, we noticed that compared to some of the performant first-order baselines (e.g. Influence Balanced), the EKFAC method

[6]Which ensures a loss reduction for all involved samples in the batch.

is more unstable with larger standard error. We posit that this is due to the *inverse scaling projection issue* pointed out in Chapter 3.5. If this issue were corrected, for example with iEF (Wu et al., 2024b), we hypothesise that we would see results that were i) more stable, and ii) better than the performance achieved by EKFAC in Table 4.2.

4.2.2 SER Tasks

Below, we present the experimental results for both SER tasks: CREMA-D and MSP-Podcast, which can be seen in Tables 4.3 and 4.4, respectively. Across both tasks, we consider the best configurations that we have achieved; in the case of the second-order optimiser experiments, we provide a detailed ablation in Chapter 5.1.

> **Remark:** We wish to point out the relative instability of the CREMA-D results. This is somewhat expected: the CREMA-D dataset, as shown in Table B.2 only has 4692 training samples compared to the $44,640$ available for training in MSP-Podcast (Table. B.3). As mentioned in Chapter 1, a condition for performant machine learning tasks is that we need *enough* data to model the underlying distribution well. When considering SER with deep learning techniques, CREMA-D exhibits some level of data scarcity, which causes large amounts of instability in training. This is corrected when we introduce more data into the equation, as seen with the MSP-Podcast experiments. To allow for this fact, we judge models based on their mean performance on CREMA-D, where we will take the standard error into account for our analysis only.

Across both the CREMA-D and MSP-Podcast results, we find that, generally, applying a correction mechanism at the loss level improves the performance of the underlying model, albeit at a greater level of stability for the MSP-Podcast models.

In this case, this *improvement* in performance is observed through the inversely proportional relationship between balanced accuracy and accuracy: there is an increase in the balanced accuracy score with a corresponding decrease in accuracy relative to the baseline[7]. In the CREMA-D results in Table 4.3, LDAM (Cao et al., 2019) is the only correction where this does not occur; instead, LDAM exaggerates the fit to the majority class evidenced by a higher accuracy and lower balanced accuracy relative to the baseline model (further shown in Figure C.2). In the MSP-Podcast results (Table 4.4), both LDAM and Focal Loss increase the bias to the majority class, evidenced by each model's respective confusion matrices in Fig. C.3, where `Neutral` and `Anger` are the most well-classified emotion classes, while

[7]Models that are biased towards the majority class have a higher accuracy; models that correct for the CI have a higher balanced accuracy. In an ideal case, the two metrics would be equal.

	CREMA-D ($p = 15.44$)	
	Acc.	Bal. Acc
Baseline	74.33 ± 1.10	54.31 ± 1.05
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	67.53 ± 3.26	60.64 ± 1.08
Class Balance (Cui et al., 2019)	72.72 ± 1.42	57.08 ± 0.57
Focal Loss (Lin et al., 2018)	73.00 ± 0.91	55.10 ± 0.57
α-Weighted Focal Loss (Lin et al., 2018)	68.53 ± 1.35	60.32 ± 1.76
LDAM (Cao et al., 2019)	75.31 ± 0.12	53.91 ± 0.55
LDAM-DRW (Cao et al., 2019)	68.86 ± 0.98	58.29 ± 0.38
Influence Balanced (Park et al., 2021)	70.25 ± 3.59	56.55 ± 4.22
iEF IRW (Wu et al., 2024b)	66.27 ± 1.63	60.69 ± 0.34
GDN Sampler	58.49 ± 2.90	57.38 ± 0.25

Table 4.3 Accuracy and Balanced Accuracy (%) for LoRA fine-tuned HuBERT on CREMA-D across three seeds for various corrections.

simultaneously being the two most-frequent emotions. The performance on the second MSP-Podcast test set is lower, due to it being more difficult inherently (Lotfian and Busso, 2019), and more imbalanced.

Across both datasets, we find that accounting for the data proportion (**C1**) is crucial to counteracting the pathologies of CI. In particular, methods that *only* correct for the data fit (**C1**) (e.g. Focal Loss (Lin et al., 2018) in both the CREMA-D and MSP-Podcast experiments) offer only slight improvements relative to our baseline. Rather, methods that adjust for the class-level imbalance, with or without local performance corrections, provide the best results. In Table 4.3, all of the three best methods in terms of balanced accuracy contain the inverse class frequency in their correction mechanism. Furthermore, adding the inverse class frequency in conjunction with per-sample correction methods (e.g. α-Weighted Focal Loss and LDAM-DRW) boosts the performance of the model relative to the per-sample corrections in isolation. Across both MSP-Podcast test sets, similar trends are seen where Re-Weighting, $\alpha-$Weighted Focal Loss, and iEF DRW post the best balanced accuracy scores. However, unlike in the case of CREMA-D, the argument is more nuanced. In particular, we find that the Class Balance and LDAM-DRW models are not able to overcome the CI problem even though they take into account the class-level disparity. Yet, including the inverse class frequency improved the performance of $\alpha-$Weighted Focal Loss relative to Focal Loss in isolation.

Further supporting **RQ.1** (Chapter 1.2), we find that iEF, a second-order optimiser, is able to overcome successfully the difficulties associated with CI for SER tasks. Across both CREMA-D and MSP-Podcast, iEF consistently produces models that are able to achieve the highest performance in the relevant benchmarks, surpassing many baseline first-order

	MSP-Podcast TS1 ($p = 11.54$)	
	Acc.	Bal. Acc
Baseline	58.21 ± 0.09	30.40 ± 0.32
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	45.96 ± 1.47	35.70 ± 0.36
Class Balance (Cui et al., 2019)	57.83 ± 0.07	30.32 ± 0.53
Focal Loss (Lin et al., 2018)	57.48 ± 0.20	30.54 ± 0.28
α**-Weighted Focal Loss** (Lin et al., 2018)	45.54 ± 1.01	34.93 ± 0.35
LDAM (Cao et al., 2019)	59.23 ± 0.21	28.19 ± 0.15
LDAM-DRW (Cao et al., 2019)	59.07 ± 0.11	28.08 ± 0.25
Influence Balanced (Park et al., 2021)	48.64 ± 5.20	32.20 ± 1.10
iEF DRW (Wu et al., 2024b)	45.09 ± 0.65	34.69 ± 0.09
GDN DRW (Benzing, 2022)	43.09 ± 0.28	32.61 ± 0.36
	MSP-Podcast TS2 ($p = 29.65$)	
	Acc.	Bal. Acc
Baseline	64.16 ± 0.55	24.54 ± 0.15
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	41.35 ± 3.81	29.20 ± 0.32
Class Balance (Cui et al., 2019)	64.05 ± 0.41	24.43 ± 0.25
Focal Loss (Lin et al., 2018)	63.64 ± 0.57	24.36 ± 0.35
α**-Weighted Focal Loss** (Lin et al., 2018)	41.88 ± 2.80	28.28 ± 0.39
LDAM (Cao et al., 2019)	65.03 ± 0.11	24.05 ± 0.11
LDAM-DRW (Cao et al., 2019)	65.04 ± 0.08	24.04 ± 0.17
Influence Balanced (Park et al., 2021)	51.78 ± 6.92	25.88 ± 0.95
iEF DRW (Wu et al., 2024b)	42.98 ± 0.31	27.79 ± 0.45
GDN DRW (Benzing, 2022)	38.03 ± 0.86	26.82 ± 0.32

Table 4.4 Accuracy and Balanced Accuracy (%) for LoRA fine-tuned HuBERT on MSP-Podcast across three seeds for various corrections. TS = Test Set.

corrections. In addition, we find that this performance is afforded with increased stability relative to other first-order corrections. In both the CREMA-D and MSP-Podcast results, the iEF-based methods report significantly lower standard errors for the balanced accuracy metric, where it achieved the lowest standard error for both CREMA-D and MSP-Podcast TS1.

When compared to its analogous first-order counterpart, Influence Balanced, our second-order optimisers are superior on two axes. Purely observing the mean balanced accuracy of the two methods, iEF-based methods are able to post superior performance. However, the most interesting aspect of the result is the relative stability on the small CREMA-D training set. In particular, we find that Influence Balanced reports a standard error of 4.22, whilst iEF IRW's standard error is only 0.34. In practice, this translates to Influence Balanced having a balanced accuracy score that varies from $48.14\% - 61.48\%$. The two models on the extrema vary from i) further perpetuating the over-fitting to the majority class and ii) encouraging a

more uniform performance across all classes. iEF IRW on the other hand had a far smaller range, $60.03\% - 61.15\%$ where all three models correct for CI. Similar trends are seen in the MSP-Podcast dataset, where the Influence Balanced method is the most unstable (evidenced by the largest reported standard error) across both test sets whilst iEF DRW reported far smaller standard errors. We posit that this erratic behavior in the case of Influence Balanced stems from the *inverse-scaling projection issue*. iEF is inherently able to mitigate this effect through the introduction of the logit-level gradient norm (Eq. 3.17) which produces far more stable optimisation dynamics.

However, we find that not all second-order optimisers are optimal for the CI. In particular, using GDN (Benzing, 2022) seems to systematically harm the representation of the majority class. As stated previously, we expect that there is some trade-off between the balanced accuracy and accuracy of the model. However, when we compare these metrics with the GDN-based models with other methods that obtain a similar result, the GDN methods consistently obtain a lower accuracy than is expected. From this, we conclude that iEF is a better optimiser than GDN, particularly when we are constrained by the difficulties of the CI problem itself. We provide a more rigorous discussion of this fact, and the inclusion of re-weighted objectives in Chapter 5.1.

A final point that we wish to address is that Re-Weighting the loss by the inverse class frequency resulted in perfomant models in both the CREMA-D and MSP-Podcast datasets. We view the opposite trend in the CIFAR-10 results (Table 4.2), where the inclusion of this factor tends to have detrimental effects. We posit that there a two potential factors at play here. Firstly, this could be a result of a change from a relatively simple vision dataset, to a far more complicated SER task. As we explore in Chapter 5.2, there are subtle differences, in terms of which techniques are effective, between the two modalities. This raises the need for a more diverse set of benchmark datasets for the CI problem which move away from the vision domain, and explore viable alternatives. A second reason for the drastic change could be attributed to the difference in the optimiser used: AdamW (Loshchilov and Hutter, 2018) instead of SGD (Robbins and Monro, 1951). Kunstner et al. (2024) showed that adaptive optimisers such as AdamW do not have difficulties w.r.t optimisation dynamics in heavy-tailed CI encountered in language modelling tasks, unlike gradient descent. While using AdamW with no correction (baseline) still overfits to the majority class, we posit that there is some interaction between the optimiser and the Re-Weighting scheme. In particular, we can view AdamW as an approximate second-order method (Loshchilov and Hutter, 2018)[8]. In effect, the initial stabilisation afforded by the inverse frequency correction allows AdamW to

[8]This would further support the notion of second-order optimisers being used a CI correctors.

make progress on all classes more effectively. We explore the class-imbalance-correcting properties of AdamW in our ablation experiments in Chapters 5.1.1 and 5.1.2.

Nevertheless, these results highlight two interesting points. Firstly, similar to the CIFAR-10 experiments, we can conclude that under the correct setup, second-order methods can be considered effective CI correctors. This provides further positive evidence in favour of **RQ.1** (Chapter 1.2), supported by the fact that the methods we present in our results are able to beat both the uncorrected baseline, and an array of other first-order corrections. Furthermore, as we will make concrete in Chapter 5.4, second-order methods offer a simpler, yet more performant, solution to the CI problem when considering the array of first-order baselines one could apply.

> **Summary:** Throughout this chapter, we presented our experimental results for the CI problem on three datasets which spanned two modalities. On CIFAR-10, we found that using a second-order method significantly outperformed not only the uncorrected baseline methods, but many other loss-level correction mechanisms that we consider in this thesis. This trend is continued across both SER tasks, albeit the second-order optimisers had to use a modified loss function. Nevertheless, we find that second-order optimisers, with their local corrective property, can alleviate much of the difficulties associated with the CI problem under the correct experimental settings. In the next chapter, we provide a more in-depth discussion of the various experimental results reported, paying particular attention to how we established our second-order methods and the computational cost associated with training the underlying model.

Chapter 5

Discussion

In Chapters 4.2.1 and 4.2.2, we showed that under a suitable configuration, second-order optimisers were indeed able to overcome the difficulties of the CI problem. In this section, we provide a more detailed discussion of the "creation" of our most performant methods that we introduce in Tables 4.3 and 4.4. We wish to provide a deeper insight into why second-order methods should be applied to solve the CI problem. Furthermore, we provide a discussion on successfully counteracting CI for SER more generally, which includes pertinent considerations such as selecting an appropriate correction mechanism, and computational complexity.

5.1 An Ablation Across Second-Order Optimisers

Given that we have shown that second-order optimisers can be used to solve the CI problem **RQ.1** (Chapter 1.2), we now turn our attention to **RQ.2**: under what conditions do second-order optimisers allow the network to learn a more fair representation? As before, we only consider models that have been trained with their "optimal" learning rate through empirical evidence, where additional experiments can be seen in Appendix B.4.

5.1.1 Raw versus Sampler

We first consider using second-order optimisers with an unmodified training objective (i.e. $\alpha_n^c = 1$) to observe whether they are able to overcome the difficulties of the CI problem in isolation. For the CIFAR-10 experiments, we find that simply using EKFAC (George et al., 2018), irrespective of whether it is implemented as `Raw` or `Sampler`, is able to significantly outperform the baseline model. This effect is confirmed when observing the confusion matrices in Fig. C.1 - second-order optimisers have a better representation across the

diagonal. In cases of a larger imbalance ($\alpha \in \{0.1, 1.0\}$), inducing the `Sampler` variation can provide slight increases in performance relative to `Raw`, where the effect is reversed for balanced datasets. However, Sampler does introduce more instability which limits the generality of our claims. To confirm whether `Sampler` derives any benefit for first-order optimisers, we apply this variant to our baseline experiment with SGD. As expected, we see that ensuring the presence of minority examples in every batch does not achieve a statistically significant improvement, and diminishes performance in some cases. This further reinforces the fact that second-order optimisers are implicit CI correctors, unlike standard fist-order optimisers, as we showed Chapter 4.2.1.

	Imbalanced CIFAR-10			**CIFAR-10**
Imbalance (α)	0.1	1.0	10.0	
Imbalance (p)	21.26	6.86	2.41	1.00
Baseline	62.33 ± 1.05	66.76 ± 0.45	69.99 ± 0.24	87.82 ± 0.65
SGD Sampler	62.75 ± 1.38	65.89 ± 0.23	67.96 ± 1.36	87.35 ± 0.34
EKFAC Raw (George et al., 2018)	66.37 ± 0.52	67.45 ± 0.95	70.88 ± 0.63	88.66 ± 0.24
EKFAC Sampler (George et al., 2018)	67.04 ± 0.67	68.98 ± 1.42	68.41 ± 0.70	88.28 ± 0.45

Table 5.1 Ablation across configurations of second-order optimisers that solve for an unmodified objective on CIFAR-10, across three seeds.

Moving to the SER tasks, we draw similar yet more nuanced findings. In the CREMA-D experiments, neither iEF nor GDN are able to provide superior performance to the baseline if the presence of minority samples in the mini-batch is not ensured. Instead, the local corrective property can only have a positive effect when combined with `Sampler`, allowing both iEF and GDN to beat the baseline and overcome the difficulties of the CI problem, visualised in Fig. C.2. In the MSP-Podcast experiments in Table 5.3, iEF, in both the `Raw` and `Sampler` variations, is able to beat the baseline in terms of balanced accuracy. We can see this effect in the confusion matrices in Fig. C.3 where iEF is able to provide a better representation across the minority classes relative to the baseline. Similar to the CIFAR-10 experiments in Table 5.1, using `Sampler` provided no statistically significant gain over `Raw`. In fact, it diminished performance in the majority class while providing no real gain across all classes on average.

We posit that the difference in conclusions drawn from CREMA-D to MSP-Podcast can be chalked up to the instability when training models on CREMA-D due to the relative data scarcity. With MSP-Podcast having many more batch updates in an epoch, the minority samples will be well considered in the limit accounting for randomness in the batch selection. This quality is not achieved on the small training set with CREMA-D, as such the way we select our batch examples has a more profound impact. Nevertheless, we still show that

in isolation, second-order optimisers, n particular iEF, are able to provide benefit for CI problems.

	CREMA-D ($p = 15.44$)	
	Acc.	Bal. Acc
Baseline	74.33 ± 1.10	54.31 ± 1.05
AdamW Sampler	74.98 ± 0.79	55.77 ± 0.78
iEF RAW (Wu et al., 2024b)	72.19 ± 1.17	54.03 ± 0.75
iEF Sampler (Wu et al., 2024b)	69.49 ± 0.47	57.53 ± 1.37
GDN RAW (Benzing, 2022)	67.71 ± 3.12	51.06 ± 1.94
GDN Sampler (Benzing, 2022)	58.49 ± 2.90	57.38 ± 0.25

Table 5.2 Ablation across configurations of second-order optimisers that solve for an unmodified objective on CREMA-D, across three seeds.

	MSP-Podcast TS1 ($p = 11.54$)	
	Acc.	Bal. Acc
Baseline	58.21 ± 0.09	30.40 ± 0.32
AdamW Sampler	52.55 ± 0.18	31.87 ± 0.10
iEF Raw (Wu et al., 2024b)	55.11 ± 0.53	31.01 ± 0.16
iEF Sampler (Wu et al., 2024b)	50.50 ± 0.49	31.20 ± 0.32
GDN Raw (Benzing, 2022)	55.66 ± 0.42	29.18 ± 0.50
GDN Sampler (Benzing, 2022)	51.99 ± 1.66	29.57 ± 0.93
	MSP-Podcast TS2 ($p = 29.65$)	
	Acc.	Bal. Acc
Baseline	64.16 ± 0.55	24.54 ± 0.15
AdamW Sampler	51.22 ± 0.63	26.10 ± 0.34
iEF Raw (Wu et al., 2024b)	59.88 ± 0.19	24.74 ± 0.29
iEF Sampler (Wu et al., 2024b)	47.70 ± 1.94	24.90 ± 0.46
GDN Raw (Benzing, 2022)	61.22 ± 0.43	24.47 ± 0.68
GDN Sampler (Benzing, 2022)	50.95 ± 3.39	24.91 ± 0.06

Table 5.3 Ablation across configurations of second-order optimisers that solve for an unmodified objective on MSP-Podcast, across three seeds.

However, unlike SGD used in the CIFAR-10 experiments, we find that AdamW does have some implicit CI correction properties. When ensuring that minority samples are always present in each batch, we find that the `Sampler` method improves the balanced accuracy, with a consistent drop in accuracy, in both the CREMA-D and MSP-Podcast results. In this setting, AdamW is now able to counteract the CI like our second-order methods. This provides further

evidence for our claim that adaptive optimisers such as AdamW have inherent properties that allow them to overcome the CI problem, supported by Kunstner et al. (2024)

Focusing on GDN, we find that it is a significantly worse optimiser than iEF. In the CREMA-D experiments, while it is able to beat the baseline in terms of balanced accuracy, it does this at the significant expense of the majority class well, unlike iEF. This poor generalisation on the majority class is continued in MSP-Podcast, while in this case it is unable to beat the Baseline neither with its `Raw` nor its `Sampler` variants. In Figure C.2, we see this issue highlighted in the case of GDN which produces models that perform sub-optimally on the majority class (`Neutral`).

5.1.2 Re-weighting: To Defer or Not To Defer

When creating solutions to solve the CI problem, many methods in the literature note the importance of including the inverse class frequency in their scalar gradient adjustments (Cao et al., 2019; Lin et al., 2018; Park et al., 2021). In particular, Park et al. (2021) and Cao et al. (2019) note the benefit of a deferred re-weighting scheme. As introduced in Chapter 4.1.5,

	Imbalanced CIFAR-10			**CIFAR-10**
Imbalance (α)	0.1	1.0	10.0	
Imbalance (p)	21.26	6.86	2.41	1.00
Baseline	62.33 ± 1.05	66.76 ± 0.45	69.99 ± 0.24	87.82 ± 0.65
EKFAC Raw (George et al., 2018)	66.37 ± 0.52	67.45 ± 0.95	70.88 ± 0.63	88.66 ± 0.24
EKFAC IRW (George et al., 2018)	62.13 ± 1.85	69.04 ± 0.71	70.75 ± 0.86	88.63 ± 0.50
EKFAC DRW (George et al., 2018)	66.03 ± 0.36	68.73 ± 1.35	71.02 ± 0.31	88.98 ± 0.30

Table 5.4 Ablation across configurations of second-order optimisers that solve for an re-weighted objective on CIFAR-10, across three seeds.

this scheme allows models to learn a good representation of the underlying feature space initially, where correcting for the inherent CI to promote a more equal performance across minority and majority classes is deferred to a later stage. In this set of ablation studies, we wish to answer whether re-weighting is important for second-order optimisers, and if so, should the re-weighting be deferred to a later stage in the training period.

In the case of the CIFAR-10 experiments in Table 5.4, we find that in settings of more extreme imbalance (i.e. $\alpha = 0.1$), not only does the re-weighted objective not help the baseline model, in some cases it significantly reduces the performance relative to EKFAC Raw. This resembles the Re-Weighting baseline we saw in Table 4.2. In the case of whether the re-weighting should be deferred or not, we find that both schemes have detrimental effects to varying degrees. Where $\alpha = 1.0$, EKFAC IRW significantly reduces the quality of the

	CREMA-D ($p = 15.44$)	
	Acc.	Bal. Acc
Baseline	74.33 ± 1.10	54.31 ± 1.05
iEF IRW (Wu et al., 2024b)	66.27 ± 1.63	60.69 ± 0.34
iEF DRW (Wu et al., 2024b)	65.71 ± 0.43	57.34 ± 1.21
GDN IRW (Benzing, 2022)	62.45 ± 2.82	56.08 ± 0.54
GDN DRW (Benzing, 2022)	65.76 ± 2.21	55.13 ± 0.89

Table 5.5 Ablation across configurations of second-order optimisers that solve for an re-weighted objective on CREMA-D, across three seeds.

underling model, reverting the performance back to the baseline level. This is less extreme in the case of EKFAC DRW, but there is a drop in accuracy nonetheless. Overall, as we decrease the amount of imbalance ($p \to 1.0$), the effect of re-weighting the loss becomes less severe, which is an expected result.

We observe the inverse of these findings for the SER tasks. Starting with CREMA-D in Table 5.5, we see that for both iEF and GDN, introducing the re-weighted objective, in addition to the per-sample normalisation afforded by the optimisers themselves, significantly improves the performance relative to our baseline setup. We notice the same effect in the analogous experiment performed on the MSP-Podcast dataset seen in Table 5.6. Although we cannot find conclusive evidence on whether we should follow the `IRW` or `DRW` scheme - as in the case of CIFAR-10, it depends on a case-to-case basis across the underlying task in question. Generally, re-weighting is helpful on the SER, to the extent that it allows the optimiser to make good progress on all classes, without ensuring the presence of minority samples in each batch.

A key insight is that, across both SER tasks, modifying the loss, whether initially or at a deferred stage, in combination with the second-order optimisers such as iEF is essential to driving the best performance across both majority and minority classes. Intuitively, while second-order optimisers ensure loss step that ensures a loss reduction for all samples involved in a mini-batch (local performance correction), there will inevitably be many more samples belonging to the majority class. As such, over time, the model will still be biased towards the majority class, relatively speaking; although this will be less extreme than using standard first-order optimisers that do not have this local loss reduction quality, as we have pointed out in Chapter 5.1.1. As such, we still need to address the data proportion term (**C1**) (provided by the inverse class frequency, for example) that will allow us to correct for the imbalance across multiple sets of updates. This is true for first-order corrections such as LDAM-DRW, α-weighted Focal Loss and Influence Balance, and it has proved to be true for second-order

optimiser as well. Additionally, we posit that this applies for more difficult tasks, such as SER, where re-weighting has detrimental effects on simpler datasets such as CIFAR-10. A possible reason for this is that due to the simplicity of the problem, the global class-level information may not be necessary to alleviate the CI, provided that adequate local correction mechanisms adjust for this in the optimisation dynamics as in the case of EKFAC in the CIFAR-10 experiments. As we showed in Chapter 3, the majority-class contribution could be supressed by proxy by the data fit term (**C2**). In contrast, the added difficulty of the SER task benefits from this additional class-level information to guide the model to produce better representations across all classes.

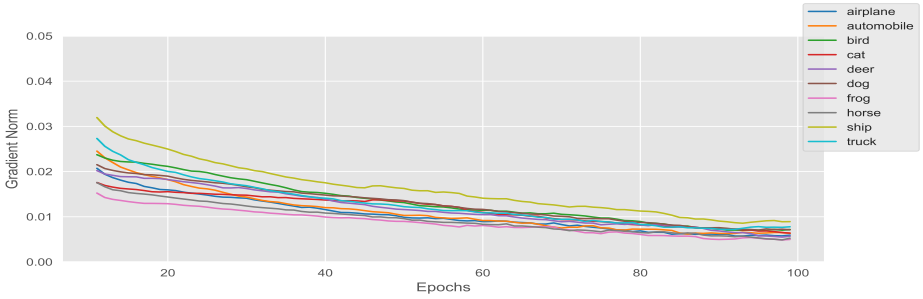
	MSP-Podcast TS1 ($p = 11.54$)	
	Acc.	Bal. Acc
Baseline	58.21 ± 0.09	30.40 ± 0.32
iEF IRW (Wu et al., 2024b)	42.03 ± 0.69	33.44 ± 0.35
iEF DRW (Wu et al., 2024b)	45.09 ± 0.65	34.69 ± 0.09
GDN IRW (Benzing, 2022)	42.96 ± 1.25	32.52 ± 0.08
GDN DRW (Benzing, 2022)	43.09 ± 0.28	32.61 ± 0.36
	MSP-Podcast TS2 ($p = 29.65$)	
	Acc.	Bal. Acc
Baseline	64.16 ± 0.55	24.54 ± 0.15
iEF IRW (Wu et al., 2024b)	40.46 ± 1.58	27.84 ± 0.78
iEF DRW (Wu et al., 2024b)	42.98 ± 0.31	27.79 ± 0.45
GDN IRW (Benzing, 2022)	40.59 ± 2.51	27.30 ± 0.42
GDN DRW (Benzing, 2022)	38.03 ± 0.86	26.82 ± 0.32

Table 5.6 Ablation across configurations of second-order optimisers that solve for an re-weighted objective on MSP-Podcast, across three seeds.

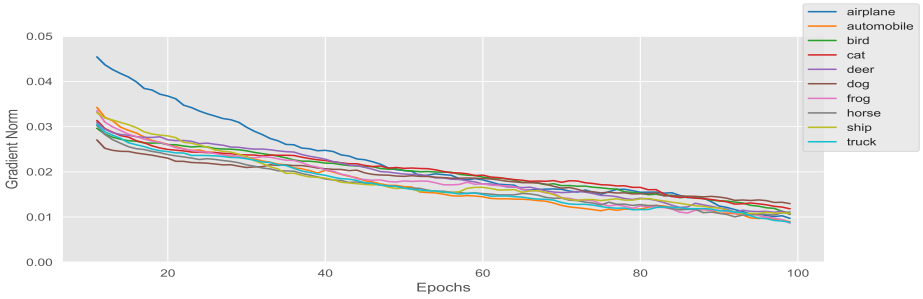
We conclude this discussion by highlighting the benefit of using iEF in terms CI performance. Across our ablations, we view that it is able to accommodate for the multiple competing optimisation objectives, promoting a more equal performance across minority classes without compromising that of the majority class more readily relative to first-order methods. This is an essential quality for the CI problem, particularly in SER, as iEF provides robust performance across all emotions. As mentioned in Chapter 1, this is imperative if we want a functioning SER system. It is for these reasons that we believe iEF is directly suited for CI problems, especially when compared to some first-order corrections.

5.2 Gradient Flows

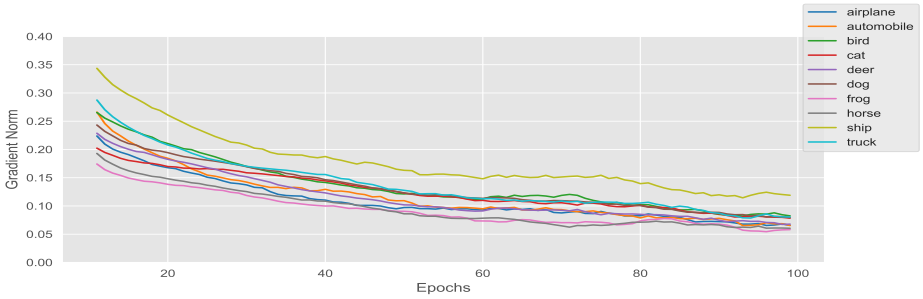
As we showed in Chapter 5.1.2, addressing the data fit (**C1**) had a detrimental effect on the underlying models for the CIFAR-10 experiments, whilst it significantly improves the performance in the case of the SER tasks. To gain a deeper understanding of why this occurs, we observe the L_2 norm of the per-class gradient for the final linear layer (Eq. 3.3 and 3.7) for the Baseline, Re-weighted and second-order method for all three datasets we have explored[1].



(a) Baseline



(b) Re-weighting (Huang et al., 2016; Wang et al., 2017)



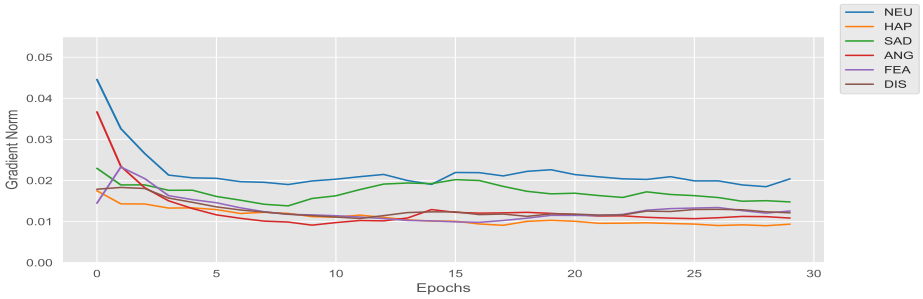
(c) EKFAC Raw (George et al., 2018)

Fig. 5.1 L_2 norm of the per-class gradient of the final linear layer for three models trained on CIFAR-10, $\alpha = 0.1$.

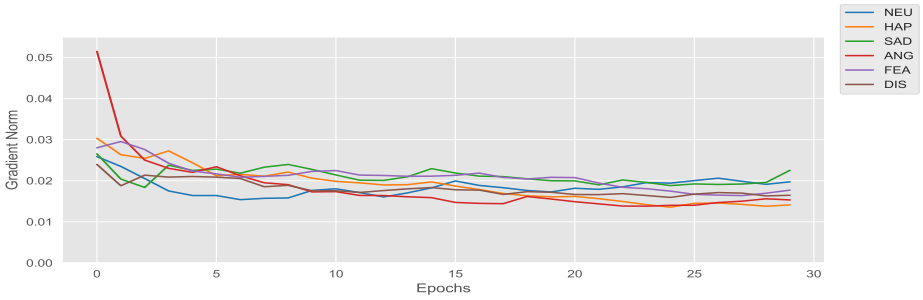
From the gradient plots in Figures 5.1, 5.2 and 5.3, we first observe some of the theoretical properties of our derivations in Chapter 3. As the model continues to train, the per-class

[1] In the case of CIFAR-10, we focus on $\alpha = 0.1$ for convenience.

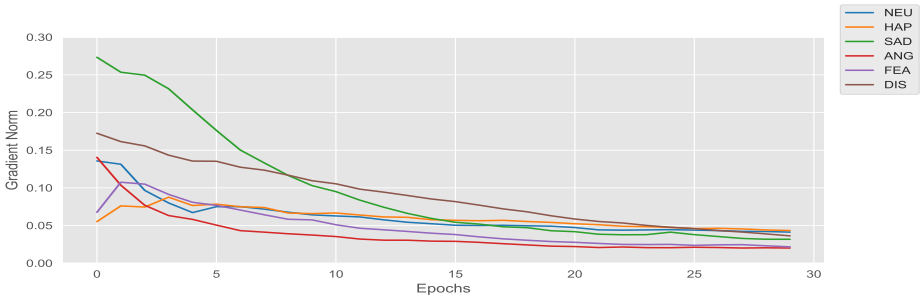
gradient norms tend to decrease; this is the influence of the data fit (**C2**) term becoming smaller. We also observe that the gradient of the majority class (`Ship` in the vision tasks, and `Neutral` in the SER tasks) is the one that dominates in the case where no re-weighting has been applied, where the data proportion term (**C1**) has not been addressed, or it has been done at deferred state where there is only a slight correction in the relative gradient norms. These curves also illuminate why, although well-motivated theoretically, the re-weighting



(a) Baseline



(b) Re-weighting (Huang et al., 2016; Wang et al., 2017)

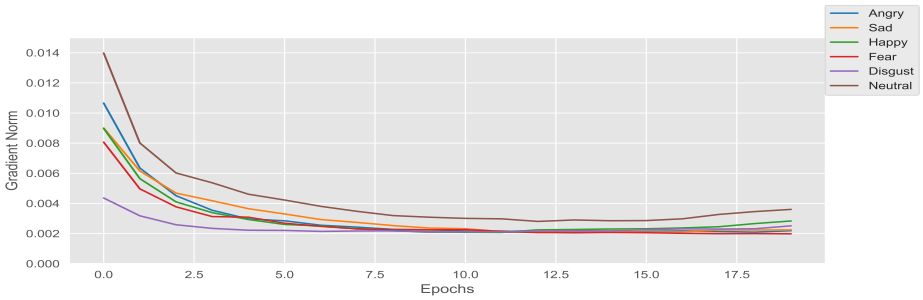


(c) iEF IRW (Wu et al., 2024b)

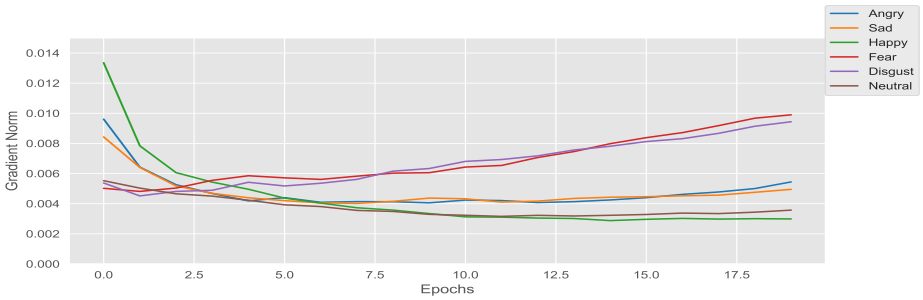
Fig. 5.2 L_2 norm of the per-class gradient of the final linear layer for three models trained on CREMA-D.

scheme fails for the CIFAR-10 experiments, but is successful for the SER tasks. Comparing Baseline and Re-weighting for CIFAR-10 (Fig. 5.1), we see that although Re-weighted has managed to remove the domination of the majority class, it seems to have converged less to a local minimum within the 100 epochs of training relative to Baseline. As such, in the

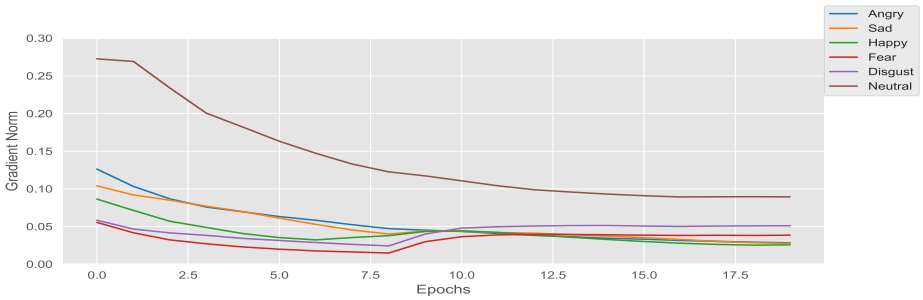
case of Re-weighting with SGD, we may need to train for longer to observe the performance benefit afforded through addressing the CI. In the SER tasks, we find that AdamW is able to



(a) Baseline



(b) Re-weighting (Huang et al., 2016; Wang et al., 2017)



(c) iEF DRW (Wu et al., 2024b)

Fig. 5.3 L_2 norm of the per-class gradient of the final linear layer for three models trained on MSP-Podcast.

better cope with the introduction of the re-weighted objective. When comparing Baseline and Re-weighting in Fig. 5.2, we find that while the majority-class gradient dominance is suppressed, the gradients in both plots display the same level of convergence. This shows that AdamW, combined with its re-weighted objective, is able to converge to a more optimal local minimum (in terms of performance across all classes), without compromising on the speed of convergence as was in the case with SGD. An exception is that of Fig. 5.3b, where the gradient norms for the two minority classes continue to increase as training progresses although the training loss decreases. This may indicate that improvement could be gained

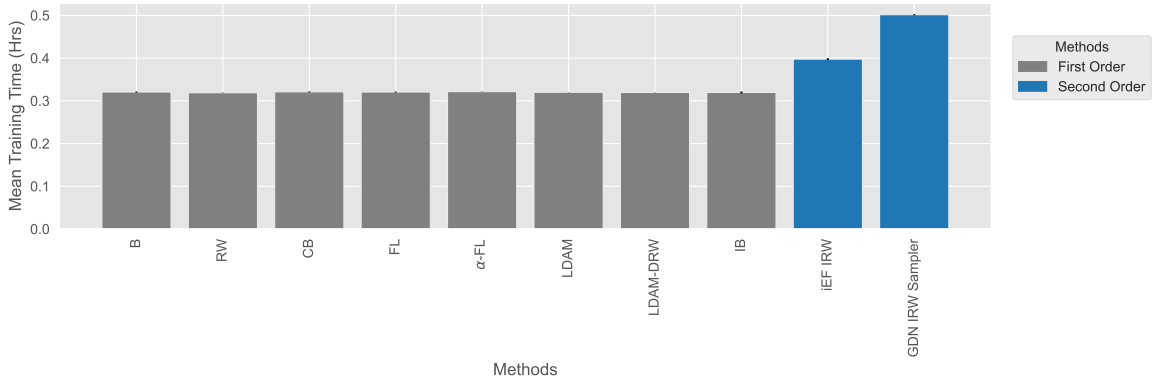
from longer training; we did not have the computational resources at hand to thoroughly examine this setting. The interaction between the re-weighted objective and AdamW ought to be explored further.

Finally, these figures shed light on why second-order optimisers are performant as CI correctors. When using EKFAC or iEF, there is a new gradient direction defined in Eq. 3.14, where the magnitude of the updates have been scaled by the inverse of the gradient norm (Eq. 3.16 and Eq. 3.17, respectively). In the gradient plots for all three datasets, we observe this effect as the L_2 norms for each class's gradient is approximately a factor of ten larger that in the case of Baseline and Re-weighting. While the majority class still dominates in the case Fig 5.1c and Fig. 5.3c, the local corrective property of this second-order optimisation leads to every single class gradient having a larger signal during training relative to using first-order methods in isolation. As such, we see that this allows the underlying model to learn a more robust representation across all classes, leading to better performance in the CI problem.

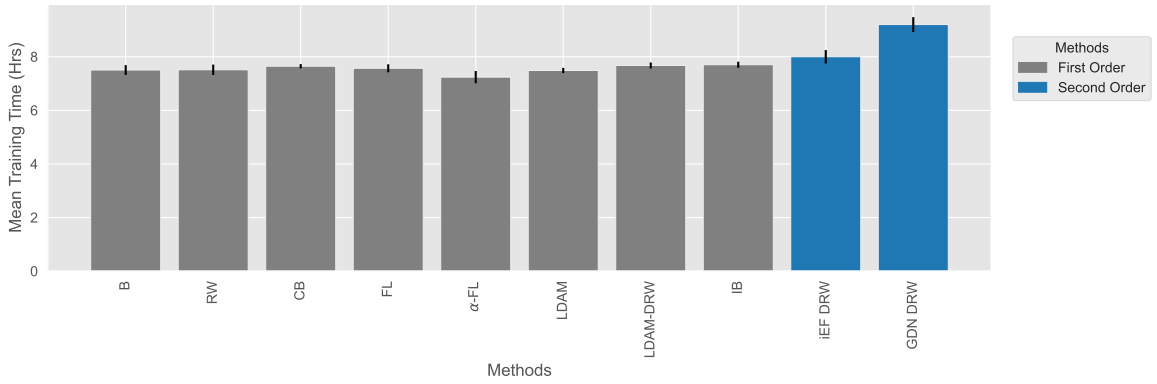
5.3 Computational Expense

A commonly raised criticism against the use of second-order methods is that they pose significant computational overhead when training very large models (Agarwal et al., 2017; Park et al., 2021; Xu et al., 2020). However, due to the work of Martens and Grosse (2020) and others (Benzing, 2022; George et al., 2018; Petersen et al., 2023; Wu et al., 2024a), using sophisticated approximations to the true Fisher information matrix as introduced in Chapter 3, we are able to bypass this overhead, while still affording many of the benefits associated with second-order methods such as faster convergence (Agarwal et al., 2017). Below, we illustrate that this is indeed the case of second-order methods on CREMA-D and MSP-Podcast experiments in Fig. 5.4

For both the CREMA-D and MSP-Podcast experiments , we find that on average, all the first-order correction methods take the same amount of time for training to complete. As expected, the second-order methods take slightly longer. However, particularly in the case of iEF (Wu et al., 2024b), the added computational complexity is relatively on par with first-order methods, while itself being a second-order method. These results reinforce that due to the work done by (Benzing, 2022; George et al., 2018; Martens and Grosse, 2020; Petersen et al., 2023; Wu et al., 2024a), second-order optimisers are both computationally tractable for large models such as HuBERT and are capable of superior performance relative to first-order loss corrections.



(a) CREMA-D



(b) MSP-Podcast

Fig. 5.4 Average training times across three seeds reported for CREMA-D and MSP-Podcast experimental results in Tables 4.3 and 4.4, respectively. First-order corrections are indicated in grey, and second-order methods in blue.

5.4 Just Choose a Suitable Optimiser

The final research question that we have yet to answer is **RQ.3** (Chapter 1.2): are CI corrections effective for the SER domain, and if so, which are the most performant? If we were to interpret the results in Chapter 4 naïvely, three methods emerge as obvious candidates: Re-Weighting, α-Weighted Focal Loss and iEF (I/D)RW - a method that we have proposed in this thesis. While all of the aforementioned corrections are roughly equally performant, they all have their own trade-offs, and each will learn a slightly different decision boundary. This result is somewhat reflective of the experimental decisions in the work of Chen et al. (2024), who used an ensemble system with seven independently trained models, which featured the Re-Weighting and α-Weighted Focal Loss corrections.

However, given that we have a constrained compute budget (i.e. we are limited to using a *single* model), how do we determine the most performant model across a number of domains

and benchmark loss-correcting techniques? The answer we give to this question is simple: *just choose a suitable optimiser*.

Perhaps the most pertinent takeaway of this work is that we find that across all three datasets, overcoming the problem of CI boils down to choosing an optimiser that has implicit class-imbalance-correcting properties. In the CIFAR-10 experiments, simply preconditioning the gradient update using EKFAC (George et al., 2018) significantly outperformed many of the first-order baselines. Similarly on the SER tasks, both AdamW[2] and iEF, coupled with a loss that was re-weighted by the inverse class frequency, proved to give the most optimal results amongst a suite of alternate correction methods. In light of this, we claim that the unique benefit of using second-order methods for CI is two-fold. Firstly, in all three benchmarks, we showed that using a second-order method, whilst placing no additional strain on the computational complexity, was always able to match, and sometimes beat, the performance of many first-order corrections. If we were simply choosing a corrector based on its empirical performance in isolation, second-order optimisers have a compelling case to win this argument. However, we find that this performance is afforded whilst being significantly easier to *tune*. One issue associated with the first-order loss corrections is that each method comes with its own set of hyperparameters. If we were to use one of these methods in conjunction with an optimiser, tuning the hyperparameters for the specific task would involve not only those of the optimiser, but of the loss correction too. This increases the number of variables we have to consider, each of which might interact with the other variables in an unpredictable way. Second-order optimisers, on the other hand, are considerably simpler; we only need to consider the hyperparameters of the optimiser itself. Given that second-order optimisers have theoretical loss-correcting properties (Chapter 3), we have shown that achieving optimal performance on a given task is equivalent to finding the optimal optimiser configuration. In some cases, the loss will need to be re-weighted by the inverse class frequency to boost the performance of the model further. However, this does not create an additional dimension in the hyperparameter space. As such, we conclude that methods such as iEF, coupled with Re-Weighting, should be considered routinely as a baseline method that is able to provide impressive performance on CI problems that emerge from a variety of domains.

Summary: In this chapter, we provided a more specific discussion of the results we presented in Chapter 4. Addressing **RQ.2**, we showed that in all cases, second-order optimisers are effective CI correctors, albeit to varying extents. In the simpler CIFAR-10 experiments, second-order optimisers were able produce a better representation with an unmodified loss. In the more difficult SER tasks, while achieving slight improvement

[2]We can interpret AdamW as an *approximate* second-order method

with an unmodified loss, inducing a re-weighted loss is what led to the significant boost in performance for second-order optimisers. In addition, we showed that there is little to no additional burden placed on the computational complexity when training models with the second-order optimisers we consider in this thesis. Finally, we address **RQ.3** by stating that simple loss-level corrections achieve the best performance on SER tasks, provided that the optimisers have an in-built local corrective property. In the next chapter, we provide our concluding remarks, limitations of our work, and we make suggestion for future directions of research.

Chapter 6

Conclusion

Throughout this thesis, we have investigated whether second-order optimisers can be used as a CI corrector. We first began by creating a *unified* mathematical framework across which we could understand and interpret the benefit that imbalance-correcting mechanisms provide to stabilise the optimisation dynamics in these problems. Within this framework, we motivated the use of second-order optimisers as CI correctors, focusing on their theoretical loss normalisation property which guarantees a loss reduction for all samples involved in the gradient update.

Supported by these theoretical foundations, we validated our claims experimentally across three different datasets which spanned two modalities. In our vision benchmark, we showed that second-order methods, optimising for an unmodified loss, significantly outperform first-order baselines that correct for the CI at a loss level. In the SER tasks, when optimising on an unmodified loss, using second-order methods resulted in an improvement, albeit marginal, over the uncorrected baseline. However, when provided with a re-weighted loss, second-order optimisers significantly overcame the difficulties associated with CI, beating many first-order baselines. This re-weighting, we concluded, was essential for optimal performance on more complex tasks (SER), while it had detrimental effects in the relatively simpler vision benchmark. We additionally showed that the performance achieved by using second-order optimisers in this setting is afforded with little to no computational overhead relative to first-order optimisers. Finally, supported by our experimental evidence, we argued that the best results w.r.t to the CI problem are achieved by using a performant second-order optimiser, coupled with simple loss corrections. By injecting the class-imbalance-correcting mechanism into the gradient step, we are able to produce performant models, superior to many first-order corrections, whilst providing a simplified *tuning* process.

6.1 Limitations

Herein, we wish to identify two limitations with our work. Firstly, while we focused on second-order optimisers that we inspired by NGD, we did not provide an interpretation of the CI properties of first-order optimisers such as AdamW, which approximate second-order methods. These properties have been alluded to by Kunstner et al. (2024); the thesis could have been improved with a more detailed focus on this class of optimiser, and subsequently unified with our mathematical framework. Secondly, due to computational constraints, we were unable to exhaustively test all hyperparameter settings for the various methods we have presented in this thesis. In this regard, we generally chose our hyperparameter sweeps based on values that were shown to perform well empirically in the original papers. In this case, there may be certain settings of hyperparameters that we have missed which would boost the performance of the respective method. However, we ensured that we tuned our algorithms to the same extent as all other baselines, meaning that this claim could apply both ways.

6.2 Future Work

Throughout our investigations that we have put forward in this thesis, we find that there are many interesting research questions that have arisen from our work and are worthy of future exploration.

Traditionally, much of the work in the field of CI has focused on benchmark from the vision domain. However, as we have seen in this thesis, some of the conclusions brought forward do not transfer from one task to another. Particularly, methods that perform well in vision tasks oftentimes do not achieve the same relative level of performance of the SER tasks we have considered in this thesis. As such, we first wish to extend our work directly to a third textual domain to ensure the robustness of our claims. Secondly, there would be great benefit to create a large benchmark study that would collate the results from the vision domain, and make further evaluations on other domains, such as SER, with various class-imbalance-correcting techniques standard practice. This would lift the reliance on vision tasks as the *de facto* benchmark dataset, and would provide practitioners with more fine-grained insights across a number of downstream use-cases. Finally, as addressed in the limitations, a worthy direction of future work would emerge from interpreting AdamW as a class-imbalance-correcting optimiser within the framework we have presented in this thesis.

References

Agarwal, N., Bullins, B., and Hazan, E. (2017). Second-Order Stochastic Optimization for Machine Learning in Linear Time. *Journal of Machine Learning Research*, 18(116):1–40.

Amari, S.-i. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276.

Anand, R., Mehrotra, K., Mohan, C., and Ranka, S. (1993). An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Batista, G. E. A. P. A., Bazzan, A., and Monard, M. C. (2003). Balancing Training Data for Automated Annotation of Keywords: a Case Study.

Benzing, F. (2022). Gradient Descent on Neurons and its Link to Approximate Second-order Optimization. In *Proceedings of the 39th International Conference on Machine Learning*, pages 1817–1853. PMLR. ISSN: 2640-3498.

Bezzam, E., Vetterli, M., and Simeoni, M. (2022). Privacy-Enhancing Optical Embeddings for Lensless Classification. arXiv:2211.12864 [cs, eess].

Biewald, L. (2020). Experiment Tracking with Weights and Biases.

Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 205–211, New York, NY, USA. Association for Computing Machinery.

Byrd, J. and Lipton, Z. (2019). What is the Effect of Importance Weighting in Deep Learning? In *Proceedings of the 36th International Conference on Machine Learning*, pages 872–881. PMLR. ISSN: 2640-3498.

Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., and Verma, R. (2014). CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6.

Chen, L.-W. and Rudnicky, A. (2023). Exploring Wav2vec 2.0 Fine Tuning for Improved Speech Emotion Recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. ISSN: 2379-190X.

Chen, M., Zhang, H., Li, Y., Luo, J., Wu, W., Ma, Z., Bell, P., Lai, C., Reiss, J., Wang, L., Woodland, P. C., Chen, X., Phan, H., and Hain, T. (2024). 1st Place Solution to Odyssey Emotion Recognition Challenge Task1: Tackling Class Imbalance Problem. arXiv:2405.20064 [cs, eess].

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., and Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518. arXiv:2110.13900 [cs, eess].

Croitoru, F.-A., Ristea, N.-C., Ionescu, R. T., and Sebe, N. (2022). LeRaC: Learning Rate Curriculum. arXiv:2205.09180 [cs].

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-Balanced Loss Based on Effective Number of Samples. arXiv:1901.05555 [cs].

De Oliveira, D., Raj Prabhu, N., and Gerkmann, T. (2023). Leveraging Semantic Information for Efficient Self-Supervised Emotion Recognition with Audio-Textual Distilled Models. In *INTERSPEECH 2023*, pages 3632–3636. ISCA.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Dua, D. and Graff, C. (2017). UCI Machine Learning Repository.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.

Feng, T., Hashemi, H., Annavaram, M., and Narayanan, S. S. (2022). Enhancing Privacy Through Domain Adaptive Noise Injection For Speech Emotion Recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7702–7706. ISSN: 2379-190X.

Feng, T. and Narayanan, S. (2023). PEFT-SER: On the Use of Parameter Efficient Transfer Learning Approaches For Speech Emotion Recognition Using Pre-trained Speech Models. In *2023 11th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8. arXiv:2306.05350 [cs, eess].

Francazi, E., Baity-Jesi, M., and Lucchi, A. (2024). A Theoretical Analysis of the Learning Dynamics under Class Imbalance. arXiv:2207.00391 [cs, stat].

Galántai, A. (2000). The theory of Newton's method. *Journal of Computational and Applied Mathematics*, 124(1):25–44.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. (2018). Fast Approximate Natural Gradient Descent in a Kronecker-factored Eigenbasis.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. pages 5036–5040.

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 87–102, Cham. Springer International Publishing.

He, H. and Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. (2022). Towards a Unified View of Parameter-Efficient Transfer Learning. arXiv:2110.04366 [cs].

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. arXiv:1902.00751 [cs, stat].

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 29:3451–3460.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs].

Huang, C., Li, Y., Loy, C. C., and Tang, X. (2016). Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384. ISSN: 1063-6919.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR. ISSN: 1938-7228.

Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54. Number: 1 Publisher: SpringerOpen.

Kapoor, S. and Narayanan, A. (2022). Leakage and the Reproducibility Crisis in ML-based Science. arXiv:2207.07048 [cs, stat].

Kim, J.-Y. and Lee, S.-H. (2023). CoordViT: A Novel Method of Improve Vision Transformer-Based Speech Emotion Recognition using Coordinate Information Concatenate. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4, Singapore. IEEE.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images.

Kunstner, F., Balles, L., and Hennig, P. (2020). Limitations of the Empirical Fisher Approximation for Natural Gradient Descent. arXiv:1905.12558 [cs, stat].

Kunstner, F., Yadav, R., Milligan, A., Schmidt, M., and Bietti, A. (2024). Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models. arXiv:2402.19449 [cs, math, stat].

Lashkarashvili, N., Wu, W., Sun, G., and Woodland, P. C. (2024). Parameter Efficient Finetuning for Speech Emotion Recognition and Domain Adaptation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10986–10990. arXiv:2402.11747 [cs, eess].

Li, M., Yang, B., Levy, J., Stolcke, A., Rozgic, V., Matsoukas, S., Papayiannis, C., Bone, D., and Wang, C. (2021). Contrastive Unsupervised Learning for Speech Emotion Recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6329–6333. ISSN: 2379-190X.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2018). Focal Loss for Dense Object Detection. arXiv:1708.02002 [cs].

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. (2019). Large-Scale Long-Tailed Recognition in an Open World. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2532–2541, Long Beach, CA, USA. IEEE.

Loshchilov, I. and Hutter, F. (2018). Decoupled Weight Decay Regularization.

Lotfian, R. and Busso, C. (2019). Building Naturalistic Emotionally Balanced Speech Corpus by Retrieving Emotional Speech from Existing Podcast Recordings. *IEEE Transactions on Affective Computing*, 10(4):471–483.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. (2018). Exploring the Limits of Weakly Supervised Pretraining. pages 181–196.

maintainers, T. and contributors (2016). TorchVision: PyTorch's Computer Vision library. Publication Title: GitHub repository.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. (2022). PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods.

Markatou, M. and Ronchetti, E. (1997). 3 Robust inference: The approach based on influence functions. In *Handbook of Statistics*, volume 15 of *Robust Inference*, pages 49–75. Elsevier.

Martens, J. and Grosse, R. (2020). Optimizing Neural Networks with Kronecker-factored Approximate Curvature. arXiv:1503.05671 [cs, stat].

Masko, D. and Hensman, P. (2015). The Impact of Imbalanced Training Data for Convolutional Neural Networks.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nisbet, R., Miner, G., and Yale, K. (2018). *Handbook of Statistical Analysis and Data Mining Applications*. Elsevier.

Park, S., Lim, J., Jeon, Y., and Choi, J. Y. (2021). Influence-Balanced Loss for Imbalanced Visual Classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 715–724, Montreal, QC, Canada. IEEE.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs, stat].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pepino, L., Riera, P., and Ferrer, L. (2021). Emotion Recognition from Speech Using wav2vec 2.0 Embeddings. pages 3400–3404.

Petersen, F., Sutter, T., Borgelt, C., Huh, D., Kuehne, H., Sun, Y., and Deussen, O. (2023). ISAAC Newton: Input-based Approximate Curvature for Newton's Method. arXiv:2305.00604 [cs, math, stat].

Petersen, K. B. and Pedersen, M. S. (2012). The Matrix Cookbook.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 [cs, eess].

Ristea, N.-C., Ionescu, R. T., and Khan, F. S. (2022). SepTr: Separable Transformer for Audio Spectrogram Processing. arXiv:2203.09581 [cs].

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407. Publisher: Institute of Mathematical Statistics.

Robinson, A. (1984). Residuals and Influence in Regression. *Royal Statistical Society. Journal. Series A: General*, 147(1):108.

Roux, N., Manzagol, P.-a., and Bengio, Y. (2007). Topmoumoute Online Natural Gradient Algorithm. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs].

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. (2018). The iNaturalist Species Classification and Detection Dataset. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, Salt Lake City, UT. IEEE.

Van Hulse, J., Khoshgoftaar, T. M., and Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 935–942, New York, NY, USA. Association for Computing Machinery.

Wagner, J., Triantafyllopoulos, A., Wierstorf, H., Schmitt, M., Burkhardt, F., Eyben, F., and Schuller, B. W. (2023). Dawn of the transformer era in speech emotion recognition: closing the valence gap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10745–10759. arXiv:2203.07378 [cs, eess].

Wang, Y.-X., Ramanan, D., and Hebert, M. (2017). Learning to Model the Tail. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs].

Wu, W., Li, B., Zhang, C., Chiu, C.-C., Li, Q., Bai, J., Sainath, T. N., and Woodland, P. C. (2024a). Handling Ambiguity in Emotion: From Out-of-Domain Detection to Distribution Estimation. arXiv:2402.12862 [cs].

Wu, X., Yu, W., Zhang, C., and Woodland, P. (2024b). An Improved Empirical Fisher Approximation for Natural Gradient Descent. arXiv:2406.06420 [cs].

Xu, P., Roosta, F., and Mahoney, M. W. (2020). Second-order Optimization for Non-convex Machine Learning: an Empirical Study. In *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*, Proceedings, pages 199–207. Society for Industrial and Applied Mathematics.

Yang, M., Xu, D., Wen, Z., Chen, M., and Xu, P. (2022). Sketch-Based Empirical Natural Gradient Methods for Deep Learning. *Journal of Scientific Computing*, 92(3):94.

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., Lee, K.-t., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., and Lee, H.-y. (2021). SUPERB: Speech processing Universal PERformance Benchmark. arXiv:2105.01051 [cs, eess].

Zhang, L., Shi, S., and Li, B. (2022a). Eva: Practical Second-order Optimization with Kronecker-vectorized Approximation.

Zhang, Y., Park, D. S., Han, W., Qin, J., Gulati, A., Shor, J., Jansen, A., Xu, Y., Huang, Y., Wang, S., Zhou, Z., Li, B., Ma, M., Chan, W., Yu, J., Wang, Y., Cao, L., Sim, K. C., Ramabhadran, B., Sainath, T. N., Beaufays, F., Chen, Z., Le, Q. V., Chiu, C.-C., Pang, R., and Wu, Y. (2022b). BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532. arXiv:2109.13226 [cs, eess].

Zinnen, M. and Salhab, M. (2023). Focal Loss Torch.

Appendix A

Mathematical Details

A.1 Derivation of Optimal Parabolic Decision Boundary

In the example shown in Figure 1.1, we have two sets of Gaussian data, namely: $p(x|y=0) = \mathcal{N}(\mathbf{x}; \mu_\mathbf{0}, \Sigma_0)$ and $p(x|y=1) = \mathcal{N}(\mathbf{x}; \mu_\mathbf{1}, \Sigma_1)$. The optimal decision boundary lies where the probability density of these two distributions is equal:

$$
\begin{aligned}
\mathcal{N}(\mathbf{x}; \mu_\mathbf{0}, \Sigma_0) &= \mathcal{N}(\mathbf{x}; \mu_\mathbf{1}, \Sigma_1) \\
\log \mathcal{N}(\mathbf{x}; \mu_\mathbf{0}, \Sigma_0) &= \log \mathcal{N}(\mathbf{x}; \mu_\mathbf{1}, \Sigma_1) \\
-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{0})^\top \Sigma_0^{-1}((\mathbf{x}-\mu_\mathbf{0})) + c &= -\frac{1}{2}(\mathbf{x}-\mu_\mathbf{1})^\top \Sigma_1^{-1}((\mathbf{x}-\mu_\mathbf{1}) + d \\
\mathbf{x}^\top \Sigma_0^{-1}\mathbf{x} - 2\mathbf{x}^\top \Sigma_0^{-1}\mu_\mathbf{0} + \mu_\mathbf{0}{}^\top \Sigma_0^{-1}\mu_\mathbf{0} + c &= \mathbf{x}^\top \Sigma_1^{-1}\mathbf{x} - 2\mathbf{x}^\top \Sigma_1^{-1}\mu_\mathbf{1} + \mu_\mathbf{1}{}^\top \Sigma_1^{-1}\mu_\mathbf{1} + d \\
\underbrace{\mathbf{x}^\top(\Sigma_0^{-1} - \Sigma_1^{-1})\mathbf{x}}_{\text{Quadratic in } \mathbf{x}} &\underbrace{- 2\mathbf{x}^\top(\Sigma_0^{-1}\mu_\mathbf{0} - \Sigma_1^{-1}\mu_\mathbf{1})}_{\text{Linear in } \mathbf{x}} + \underbrace{\mu_\mathbf{0}{}^\top \Sigma_0^{-1}\mu_\mathbf{0} - \mu_\mathbf{1}{}^\top \Sigma_1^{-1}\mu_\mathbf{1} + c - d}_{\text{Scalar}} = 0 \\
\text{where } c = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_0| &\text{ and } d = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_1|
\end{aligned}
$$

The above derivation shows that, in the general case, the optimal decision boundary between two Gaussian distributions is parabolic in $\mathbf{x}$.

A.2 Per-Class Derivation of Cross-Entropy Loss

The following derivation is based on the work of Kunstner et al. (2024) which we include here for the convenience of the reader.

For this task, we assume that we are in the domain of a supervised learning problem (i.e. classification) where we use the cross-entropy loss.

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{\mathcal{C}} y_n^c \log \hat{y}_n^c$$

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{\mathcal{C}} y_n^c (z_n^c - \log \sum_{c'=1}^{\mathcal{C}} z_n^{c'})$$

Taking the first derivative of the loss w.r.t class c, we obtain:

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}(\mathbb{1}[y_n = c]\mathbf{x}_n - \frac{z_n^c}{\sum_{c=1}^{\mathcal{C}} z^{c_n}}\mathbf{x}_n)$$

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}(\mathbb{1}[y_n = c]\mathbf{x}_n - \hat{y}_n^{y_n}\mathbf{x}_n)$$

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n$$

Now we wish to show how the class components can be derived, bearing in mind that $\pi_c = \frac{n_c}{N} \rightarrow \frac{1}{N} = \frac{\pi_c}{n_c}$:

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \frac{1}{N}\sum_{c=1}^{\mathcal{C}}\sum_{n=1,y_n=c}^{N}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n$$

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \sum_{c=1}^{\mathcal{C}}\frac{\pi_c}{n_c}\sum_{n=1,y_n=c}^{N}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n$$

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \frac{\pi_c}{n_c}\sum_{n=1:y_n=c}^{N}[\hat{y}_n^{y_n} - \mathbb{1}[y_n = c]]\mathbf{x}_n + \sum_{j=1:j\neq c}^{\mathcal{C}}\frac{\pi_j}{n_j}\sum_{n=1:n=j}^{N}(\hat{y}_n^{y_n})\mathbf{x}_n$$

Taking the average across all data points in a class where $\bar{\mathbf{x}}^c = \frac{1}{n_c}\sum_{n=1:y_n=c}^{N}\mathbf{x}_n$ and assuming $\hat{y}_n^{yn} = p$ is the same for all data points:

$$\nabla_{\mathbf{w}_c}\mathcal{L}(\theta) = \pi_c \cdot [p-1]\bar{\mathbf{x}}^c + \sum_{j=1:j\neq c}^{\mathcal{C}}\frac{\pi_j}{n_j}\sum_{n=1:n=j}^{N}(\hat{y}_n^{y_n})\mathbf{x}_n$$

A.3 Metrics

A.3.1 Accuracy

$$\textbf{Accuracy} = \frac{\sum_{c=1}^{\mathcal{C}} N_{\text{Correct}}^{c}}{\sum_{c=1}^{\mathcal{C}} n_c} = \frac{\sum_{c=1}^{\mathcal{C}} N_{\text{Correct}}^{c}}{N}$$

A.3.2 Balanced Accuracy

Unweighted arithmetic mean of per-class accuracies.

$$\textbf{Balanced Accuracy} = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} \frac{N_{\text{Correct}}^{c}}{n_c} = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} \text{Accuracy}_c$$

A.3.3 Macro F1

Unweighted arithmetic mean of per-class F1 scores.

$$\textbf{Macro F1} = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} \text{F1}_c = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} \frac{2 \cdot TP_c}{2 \cdot TP_c + FP_c + FN_c}$$

Appendix B

Additional Experiment Details

B.1 Dataset Statistics

	CIFAR-10					
	Train				Validation	Test
α	0.1	1.0	10.0	Balanced		
p	21.26	6.86	2.41	1.00	1.00	1.00
Airplane	118	299	634	4500	500	1000
Automobile	271	472	427	4500	500	1000
Bird	306	347	651	4500	500	1000
Cat	150	296	270	4500	500	1000
Deer	211	207	364	4500	500	1000
Dog	305	390	429	4500	500	1000
Frog	110	216	477	4500	500	1000
Horse	170	837	392	4500	500	1000
Ship	2339	1248	433	4500	500	1000
Truck	515	182	419	4500	500	1000
Total	4495	4494	4496	45000	5000	10000

Table B.1 Statistics for CIFAR-10 (Krizhevsky, 2009).

	CREMA-D		
	Train	Validation	Test
p	10.65	8.09	15.44
Neutral	2642	372	880
Happy	248	48	57
Sad	267	46	57
Angry	673	99	214
Fear	475	46	123
Disgust	387	59	100
Total	4692	670	1431

Table B.2 Statistics for CREMA-D (Cao et al., 2014).

	MSP-Podcast			
	Train	Validation	Test 1	Test 2
p	16.35	12.17	11.54	29.65
Angry	3555	1180	803	329
Sad	2802	331	537	411
Happy	12386	1847	3970	2351
Fear	1403	271	474	204
Disgust	1556	465	732	193
Neutral	22938	3298	5469	5722
Total	44640	7392	11985	9210

Table B.3 Statistics for MSP-Podcast (Lotfian and Busso, 2019).

B.2 Details on Hyperparameters

Symbol	Description of Hyperparameter
η	Learning rate used for various optimisers. $\eta_i \rightarrow \eta_j$ indicates that the learning rate was modified during training, where η_i was the learning rate before any deferred re-weighting occurred, and η_j is the learning rate that was used after deferral.
β	Hyperparameter used in Class Balanced Re-weighting (Cui et al., 2019) used to determine the re-weighting factor. Linked to empirical number of samples.
γ	Modulation factor used for Focal Loss (Lin et al., 2018) and associated methods.
C	Hyperparameter used methods that use LDAM (Cao et al., 2019).
ζ	Hyperparameter for tuning with Influence Balanced (Park et al., 2021).
B	Batch size

Table B.4 Legend for all hyperparameters.

B.2.1 CIFAR-10

	η	β	γ	C	ζ	B
Baseline	0.1	-	-	-	-	32
Focal Loss (Lin et al., 2018)	0.1	-	1.0	-	-	32
α**-Weighted Focal Loss** (Lin et al., 2018)	0.1	-	2.0	-	-	32
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	0.1	-	-	-	-	32
Class Balance (Cui et al., 2019)	0.1	0.999	-	-	-	32
LDAM (Cao et al., 2019)	0.01	-	-	0.5	-	32
LDAM-DRW (Cao et al., 2019)	0.01	-	-	0.5	-	32
Influence Balanced (Park et al., 2021)	0.1	-	-	-	1000.0	32
EKFAC (George et al., 2018)	0.01	-	-	-	-	32
EKFAC-DRW (George et al., 2018)	0.01	-	-	-	-	32

Table B.5 Hyperparameters used for the CIFAR-10 experiments.

B.2.2 CREMA-D

	η	β	γ	C	ζ	B
Baseline	5×10^{-4}	-	-	-	-	64
Sampler	5×10^{-4}	-	-	-	-	64
Focal Loss (Lin et al., 2018)	5×10^{-4}	-	1.0	-	-	64
α**-Weighted Focal Loss** (Lin et al., 2018)	5×10^{-4}	-	2.0	-	-	64
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	5×10^{-4}	-	-	-	-	64
Class Balance (Cui et al., 2019)	5×10^{-4}	0.999	-	-	-	64
LDAM (Cao et al., 2019)	5×10^{-4}	-	-	0.5	-	64
LDAM-DRW (Cao et al., 2019)	5×10^{-4}	-	-	0.5	-	64
Influence Balanced (Park et al., 2021)	5×10^{-4}	-	-	-	1000.0	64
iEF Raw (Wu et al., 2024b)	50.0	-	-	-	-	64
iEF Sampler (Wu et al., 2024b)	50.0	-	-	-	-	64
iEF IRW (Wu et al., 2024b)	50.0	-	-	-	-	64
iEF DRW (Wu et al., 2024b)	50.0	-	-	-	-	64
GDN Raw (Benzing, 2022)	10.0	-	-	-	-	64
GDN Sampler (Benzing, 2022)	10.0	-	-	-	-	64
GDN IRW (Benzing, 2022)	10.0	-	-	-	-	64
GDN DRW (Benzing, 2022)	10.0	-	-	-	-	64

Table B.6 Hyperparameters used for the CREMA-D experiments.

B.2.3 MSP-Podcast

	η	β	γ	C	ζ	B
Baseline	1×10^{-4}	-	-	-	-	32
Sampler	1×10^{-4}	-	-	-	-	64
Focal Loss (Lin et al., 2018)	1×10^{-4}	-	1.0	-	-	32
α**-Weighted Focal Loss** (Lin et al., 2018)	1×10^{-4}	-	2.0	-	-	32
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	1×10^{-4}	-	-	-	-	32
Class Balance (Cui et al., 2019)	1×10^{-4}	0.999	-	-	-	32
LDAM (Cao et al., 2019)	1×10^{-4}	-	-	0.5	-	32
LDAM-DRW (Cao et al., 2019)	1×10^{-4}	-	-	0.5	-	32
Influence Balanced (Park et al., 2021)	1×10^{-4}	-	-	-	1000.0	64
iEF Raw (Wu et al., 2024b)	10.0	-	-	-	-	32
iEF Sampler (Wu et al., 2024b)	10.0	-	-	-	-	64
iEF IRW (Wu et al., 2024b)	5.0	-	-	-	-	32
iEF DRW (Wu et al., 2024b)	$10.0 \rightarrow 5.0$	-	-	-	-	32
GDN Raw (Benzing, 2022)	10.0	-	-	-	-	32
GDN Sampler (Benzing, 2022)	10.0	-	-	-	-	64
GDN IRW (Benzing, 2022)	5.0	-	-	-	-	32
GDN DRW (Benzing, 2022)	$10.0 \rightarrow 5.0$	-	-	-	-	32

Table B.7 Hyperparameters used for the MSP-Podcast experiments.

B.3 Test Macro F1 Scores

We report the Macro F1 scores for the CREMA-D and MSP-Podcast test sets for the sake of transparency. Although this metric has been used to report performance in Chen et al. (2024), we feel that it does not adequately capture the trade-off between overfitting on the majority classes versus having a good model across all emotions.

	CREMA-D ($p = 15.44$)
	Macro F1
Baseline	54.95 ± 0.18
AdamW Sampler	56.45 ± 0.56
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	54.86 ± 1.60
Class Balance (Cui et al., 2019)	55.56 ± 0.97
Focal Loss (Lin et al., 2018)	54.21 ± 0.38
α**-Weighted Focal Loss** (Lin et al., 2018)	55.20 ± 1.52
LDAM (Cao et al., 2019)	55.19 ± 0.59
LDAM-DRW (Cao et al., 2019)	53.97 ± 0.37
Influence Balanced (Park et al., 2021)	54.22 ± 1.20

iEF RAW (Wu et al., 2024b)	53.35 ± 1.13
iEF Sampler (Wu et al., 2024b)	54.03 ± 1.50
iEF IRW (Wu et al., 2024b)	53.32 ± 0.67
iEF DRW (Wu et al., 2024b)	51.25 ± 0.85
GDN RAW (Benzing, 2022)	48.35 ± 0.71
GDN Sampler (Benzing, 2022)	49.34 ± 1.10
GDN IRW (Benzing, 2022)	49.40 ± 2.10
GDN DRW (Benzing, 2022)	50.44 ± 0.97

Table B.8 Additional test set Macro F1 scores reported for all CREMA-D experiments.

	MSP-Podcast TS1 ($p = 11.54$)
	Macro-F1
Baseline	30.38 ± 0.36
AdamW Sampler	28.55 ± 0.41
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	30.96 ± 0.25
Class Balance (Cui et al., 2019)	30.48 ± 0.64
Focal Loss (Lin et al., 2018)	30.44 ± 0.28
α**-Weighted Focal Loss** (Lin et al., 2018)	30.55 ± 0.05
LDAM (Cao et al., 2019)	27.40 ± 0.15
LDAM-DRW (Cao et al., 2019)	27.50 ± 0.29
Influence Balanced (Park et al., 2021)	29.25 ± 0.91
iEF Raw (Wu et al., 2024b)	31.41 ± 0.26
iEF Sampler (Wu et al., 2024b)	27.45 ± 0.35
iEF IRW (Wu et al., 2024b)	29.74 ± 0.18
iEF DRW (Wu et al., 2024b)	31.36 ± 0.34
GDN Raw (Benzing, 2022)	29.50 ± 0.36
GDN Sampler (Benzing, 2022)	26.94 ± 0.27
GDN IRW (Benzing, 2022)	29.20 ± 0.53
GDN DRW (Benzing, 2022)	29.57 ± 0.38
	MSP-Podcast TS2 ($p = 29.65$)
	Macro-F1
Baseline	25.03 ± 0.21
AdamW Sampler	22.29 ± 0.39
Re-Weighting (Huang et al., 2016; Wang et al., 2017)	22.28 ± 0.94
Class Balance (Cui et al., 2019)	24.92 ± 0.35
Focal Loss (Lin et al., 2018)	24.64 ± 0.56
α**-Weighted Focal Loss** (Lin et al., 2018)	21.99 ± 0.57
LDAM (Cao et al., 2019)	23.96 ± 0.11
LDAM-DRW (Cao et al., 2019)	24.14 ± 0.10
Influence Balanced (Park et al., 2021)	23.22 ± 0.84
iEF Raw (Wu et al., 2024b)	25.26 ± 0.30

iEF Sampler (Wu et al., 2024b)	20.65 ± 0.65
iEF IRW (Wu et al., 2024b)	23.20 ± 0.58
iEF DRW (Wu et al., 2024b)	23.68 ± 0.31
GDN Raw (Benzing, 2022)	24.53 ± 0.42
GDN Sampler (Benzing, 2022)	21.02 ± 0.80
GDN IRW (Benzing, 2022)	22.19 ± 0.57
GDN DRW (Benzing, 2022)	22.10 ± 0.11

Table B.9 Additional test set Macro F1 scores reported for all MSP-Podcast experiments.

B.4 Additional Second-Order Experiments on CREMA-D

	CREMA-D			
	η	Acc.	Macro-F1	Bal. Acc
iEF RAW	1.0	73.33 ± 0.10	46.96 ± 0.58	44.46 ± 0.49
iEF IRW	1.0	52.60 ± 0.48	47.17 ± 0.69	56.56 ± 1.12
iEF DRW	1.0	56.14 ± 1.55	47.54 ± 0.40	55.52 ± 0.15
iEF Sampler	1.0	73.26 ± 0.08	48.65 ± 0.58	46.29 ± 0.61
iEF RAW	10.0	74.24 ± 0.62	49.70 ± 0.16	48.02 ± 0.32
iEF IRW	10.0	60.00 ± 1.29	51.31 ± 1.05	59.45 ± 1.78
iEF DRW	10.0	62.89 ± 3.32	50.88 ± 1.19	56.28 ± 0.90
iEF Sampler	10.0	73.56 ± 0.43	50.52 ± 1.49	49.23 ± 1.65
iEF RAW	100.0	52.43 ± 4.90	22.99 ± 3.38	26.80 ± 4.36
iEF IRW	100.0	10.69 ± 2.82	9.78 ± 3.41	22.92 ± 2.34
iEF DRW	100.0	46.19 ± 18.96	18.30 ± 5.66	23.87 ± 1.02
iEF Sampler	100.0	26.65 ± 2.13	26.58 ± 2.86	39.55 ± 3.19
GDN RAW	1.0	75.56 ± 0.34	47.05 ± 0.87	44.53 ± 0.59
GDN IRW	1.0	59.14 ± 0.93	48.81 ± 0.55	57.51 ± 0.34
GDN DRW	1.0	60.82 ± 0.49	50.17 ± 0.05	57.87 ± 0.33
GDN Sampler	1.0	75.03 ± 0.06	48.81 ± 1.31	46.32 ± 1.19
GDN RAW	50.0	58.07 ± 3.42	13.22 ± 0.53	16.63 ± 0.03
GDN IRW	50.0	29.75 ± 11.58	12.04 ± 3.15	17.72 ± 0.53
GDN DRW	50.0	53.30 ± 9.56	16.54 ± 1.43	18.33 ± 1.27
GDN Sampler	50.0	34.47 ± 11.82	23.98 ± 9.81	34.01 ± 9.58

Table B.10 Additional test set results of LoRA fine-tuned HuBERT on CREMA-D across three seeds for various second-order optimiser configurations.

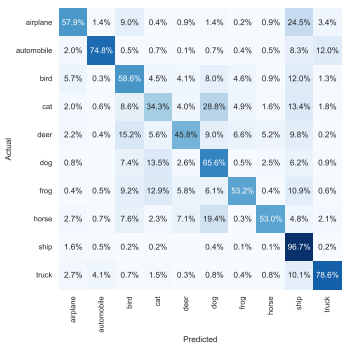
Note: Whilst we did take care to tune the learning rate on MSP-Podcast, we did not perform experiments across a number of seeds for sub-optimal configurations due to limited computational resources.

Appendix C

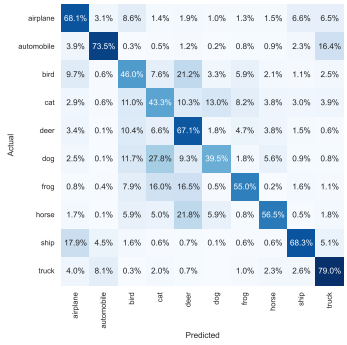
Additional Experiment Visualisations

In this section, we provide additional visualisations for our various experiments. For the sake of space, we only report the CIFAR-10 visualisations where $\alpha = 0.1$

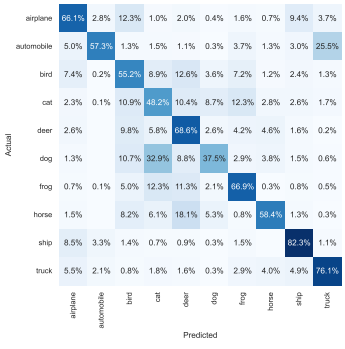
C.1 CIFAR-10 Confusion Matrices



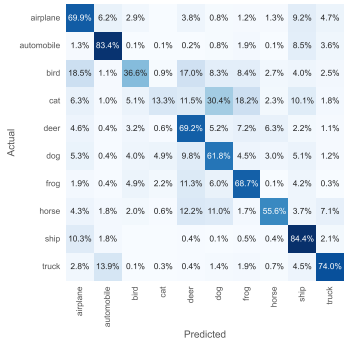
Baseline



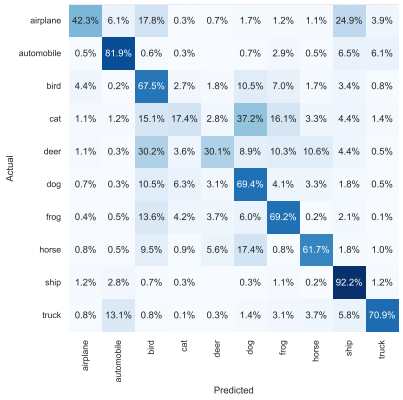
Re-weighting (Huang et al., 2016; Wang et al., 2017)



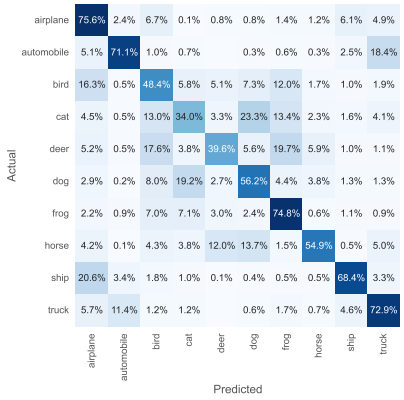
Class Balance (Cui et al., 2019)



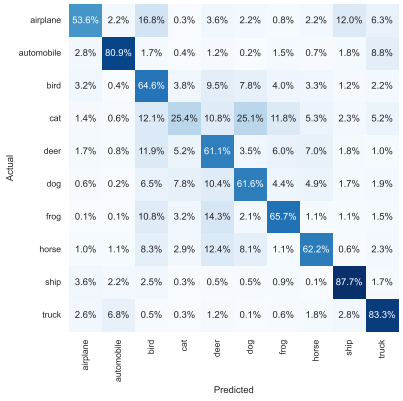
Sampler



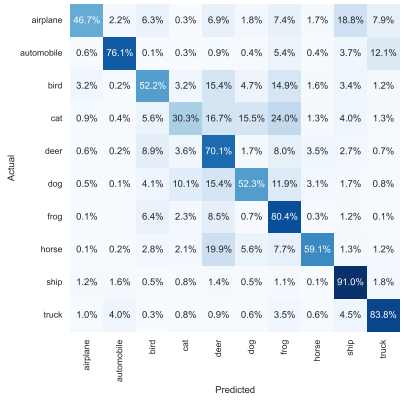
Focal Loss (Lin et al., 2018)



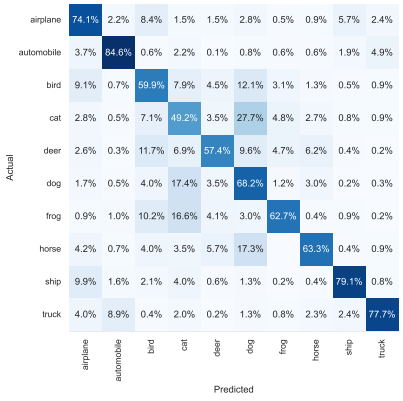
α-Weighted Focal Loss (Lin et al., 2018)



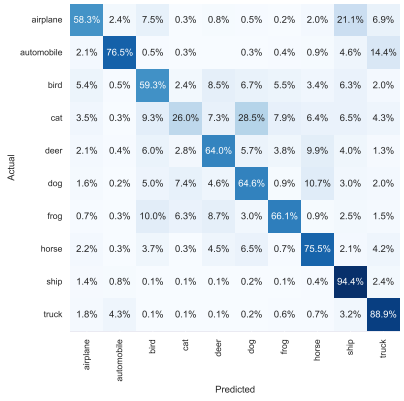
LDAM (Cao et al., 2019)



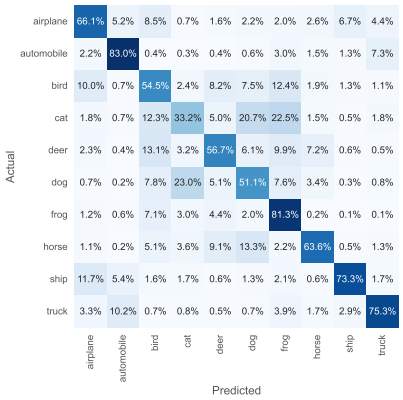
LDAM-DRW (Cao et al., 2019)



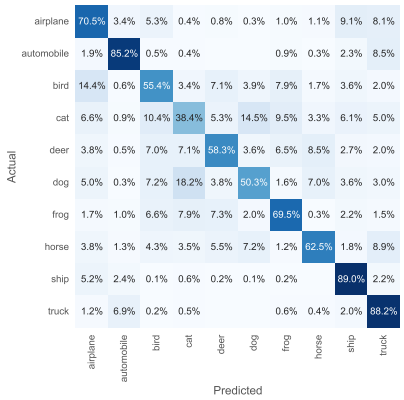
Influence Balanced (Park et al., 2021)



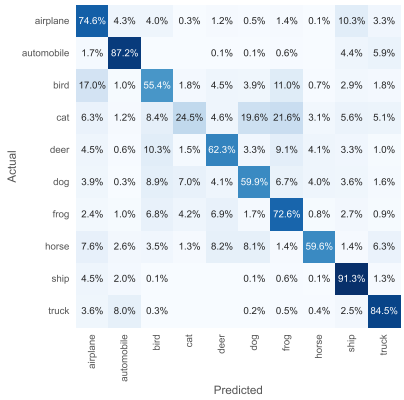
EKFAC Raw (George et al., 2018)



EKFAC IRW (George et al., 2018)



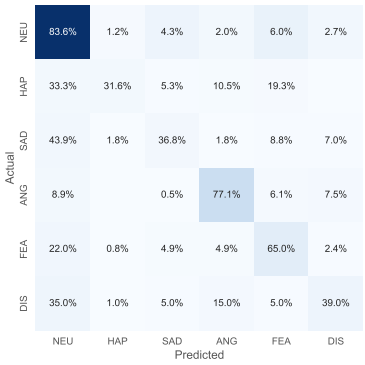
EKFAC DRW (George et al., 2018)



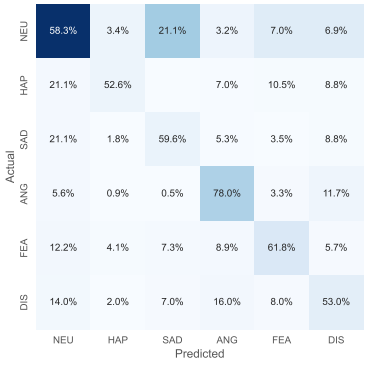
EKFAC Sampler (George et al., 2018)

Fig. C.1 Confusion matrices for the CIFAR-10 test set $\alpha = 0.1$ across various CI correctors.

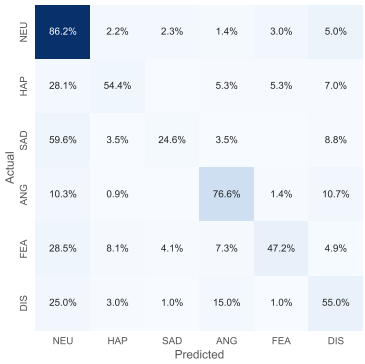
C.2 CREMA-D Confusion Matrices



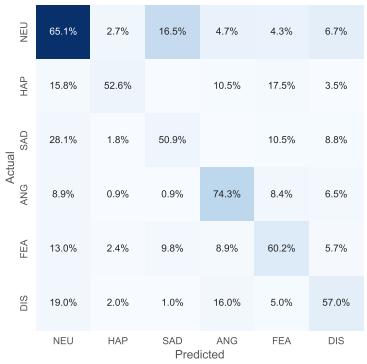
Baseline



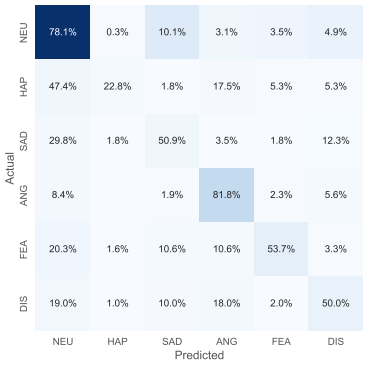
Re-weighting (Huang et al., 2016; Wang et al., 2017)



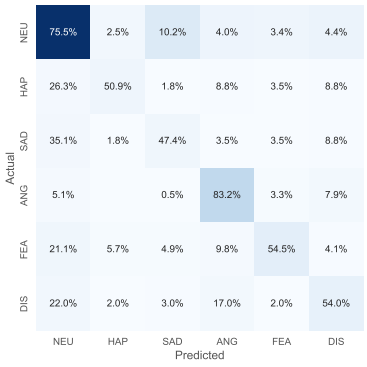
Class Balanced (Cui et al., 2019)



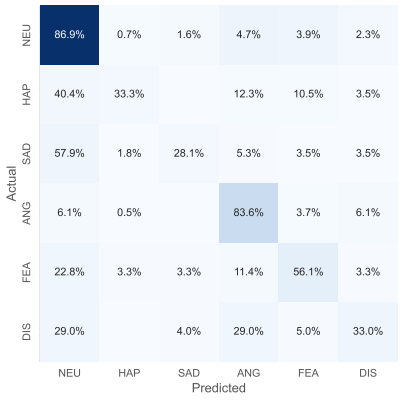
Influence Balanced (Park et al., 2021)



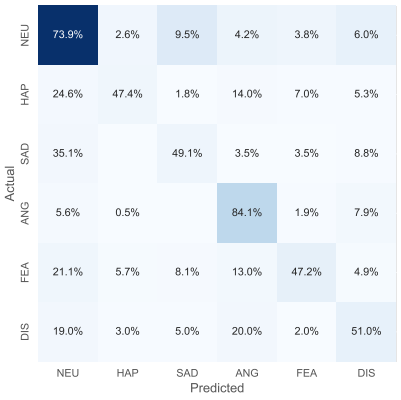
Focal Loss (Lin et al., 2018)



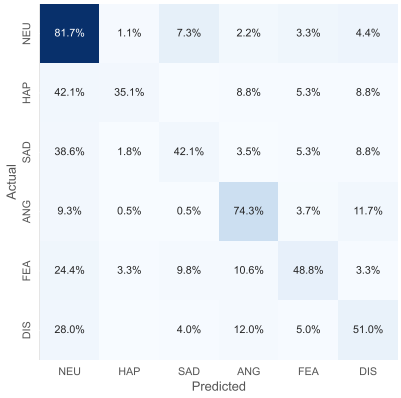
α-Weighted Focal Loss (Lin et al., 2018)



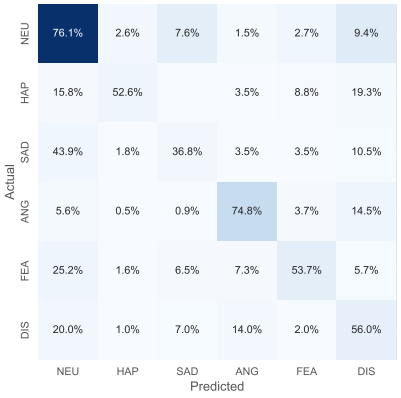
LDAM (Cao et al., 2019)



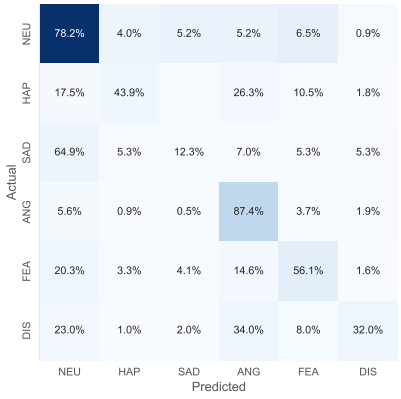
LDAM-DRW (Cao et al., 2019)



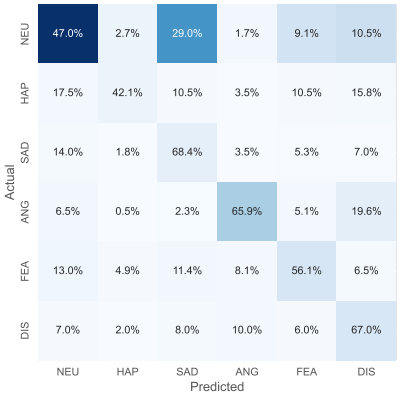
iEF Raw



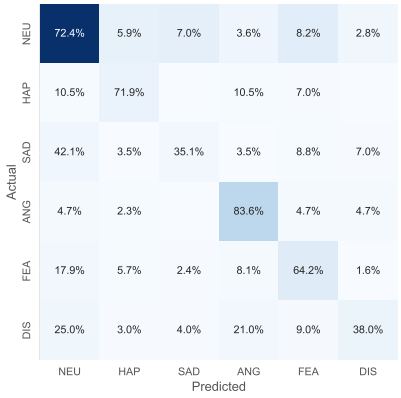
iEF Sampler



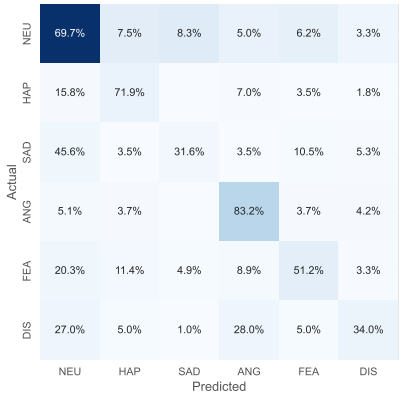
GDN Raw



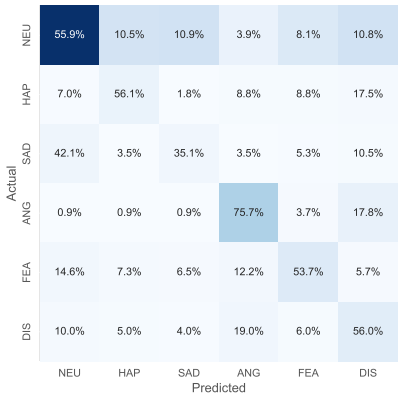
GDN Sampler



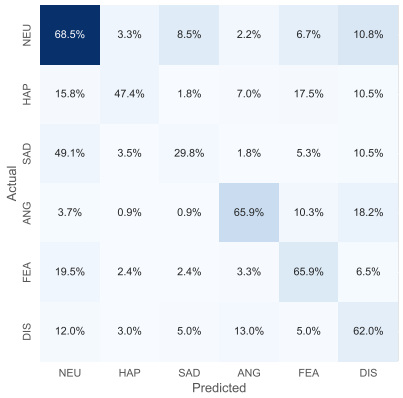
iEF IRW (Wu et al., 2024b)



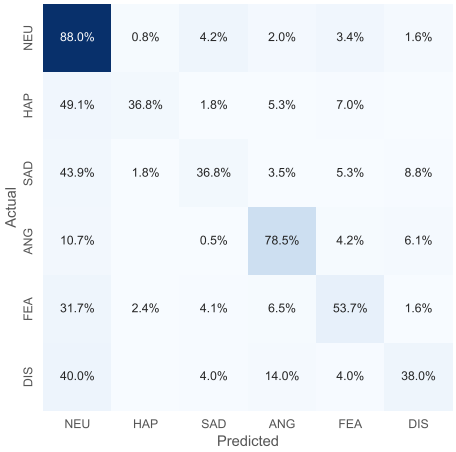
iEF DRW (Wu et al., 2024b)



GDN IRW (Benzing, 2022)



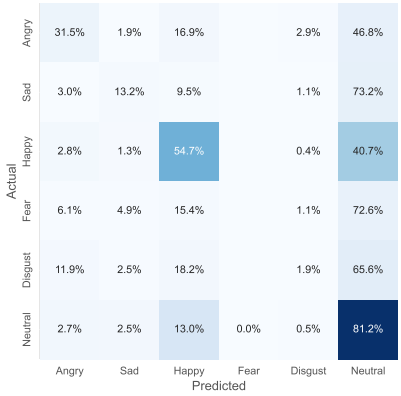
GDN DRW (Benzing, 2022)



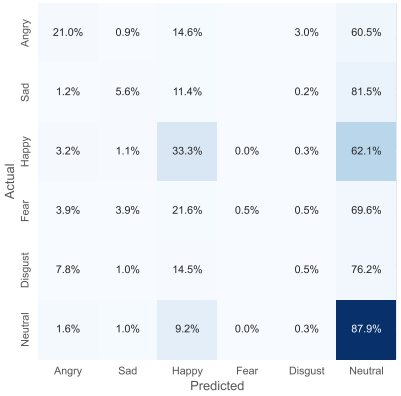
Sampler

Fig. C.2 Confusion matrices for the CREMA-D test set $\alpha = 0.1$ across various CI correctors.

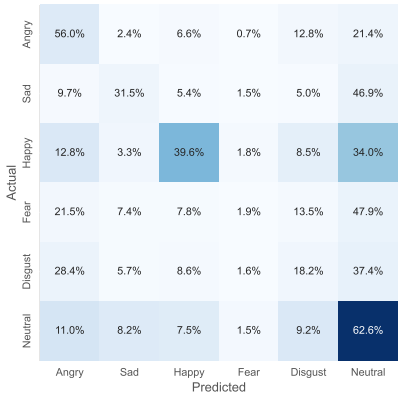
C.3 MSP-Podcast Confusion Matrices



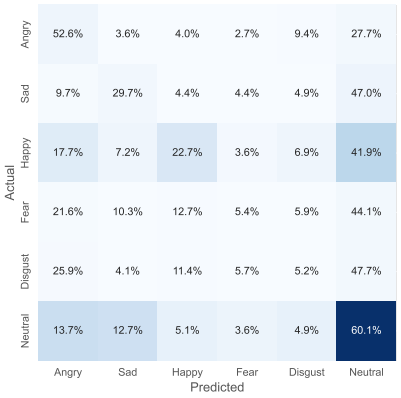
Baseline TS1



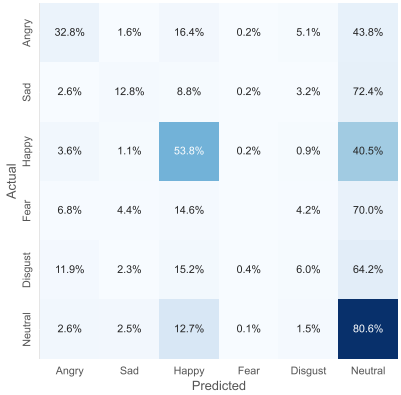
Baseline TS2



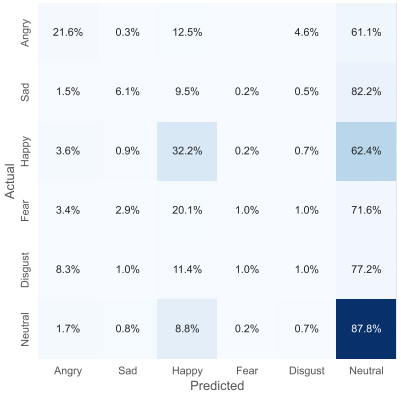
Re-weighting (Huang et al., 2016; Wang et al., 2017) TS1



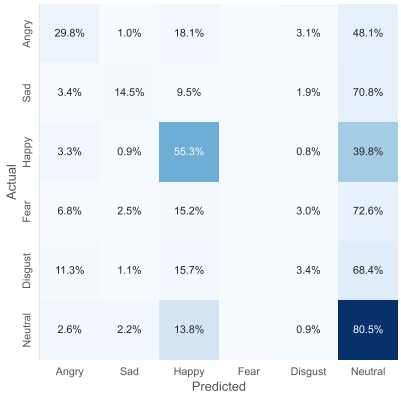
Re-weighting (Huang et al., 2016; Wang et al., 2017) TS2



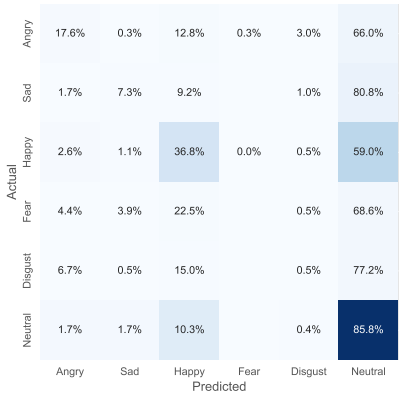
Class Balance (Cui et al., 2019) TS1



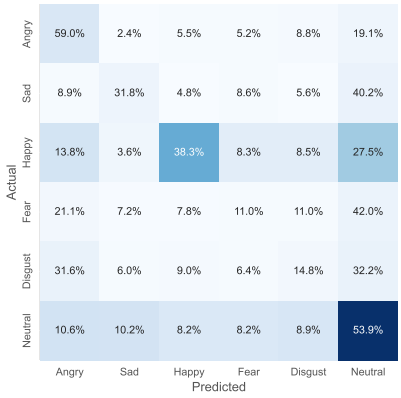
Class Balance Cui et al. (2019) TS2



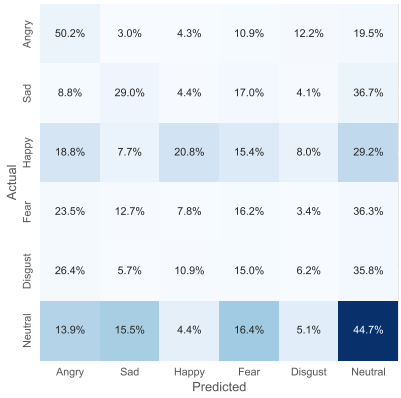
Focal Loss (Lin et al., 2018) TS1



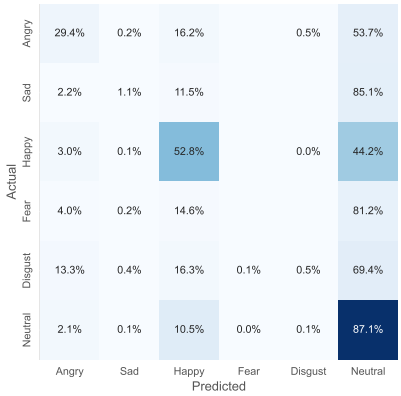
Focal Loss (Lin et al., 2018) TS2



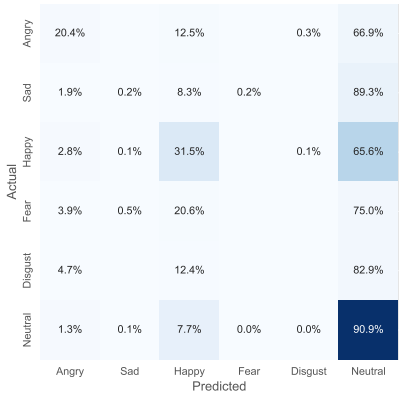
α-Weighted Focal Loss (Lin et al., 2018) TS1



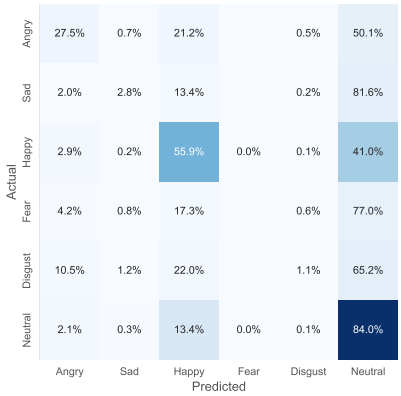
α-Weighted Focal Loss (Lin et al., 2018) TS2



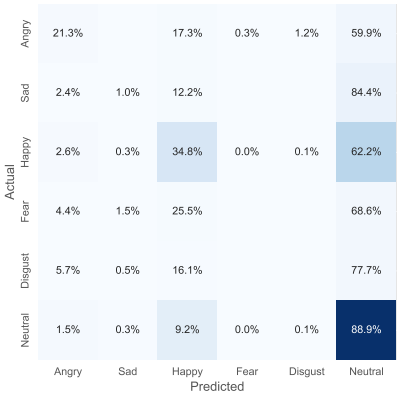
LDAM (Cao et al., 2019) TS1



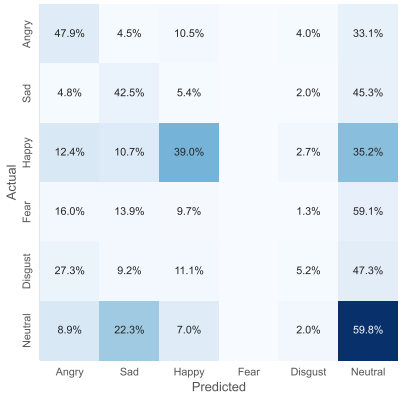
LDAM (Cao et al., 2019) TS2



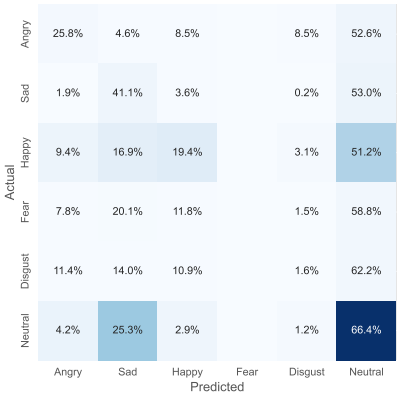
LDAM-DRW (Cao et al., 2019) TS1



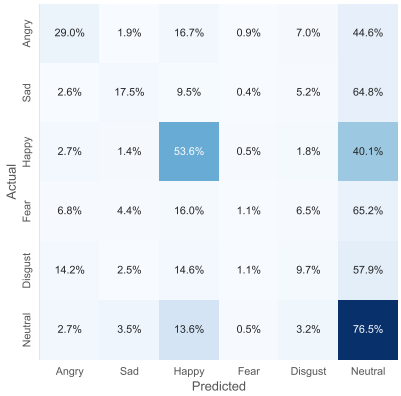
LDAM-DRW (Cao et al., 2019) TS2



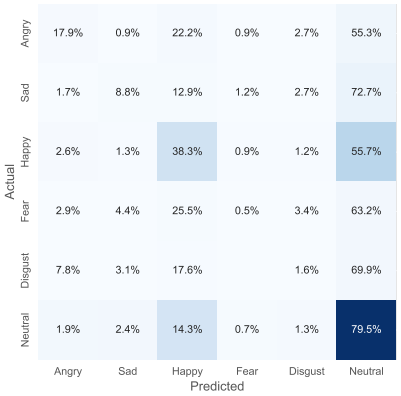
Influence Balanced (Park et al., 2021) TS1



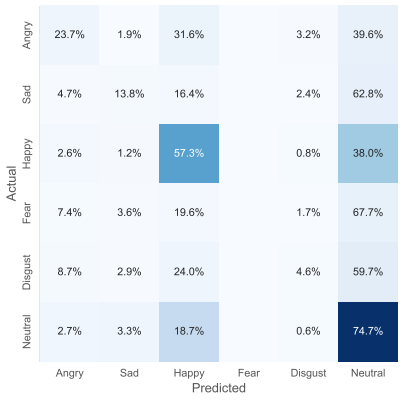
Influence Balanced (Park et al., 2021) TS2



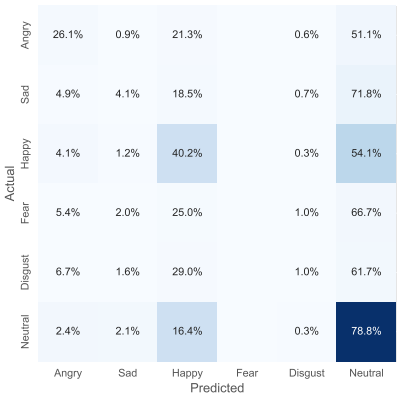
iEF Raw TS1



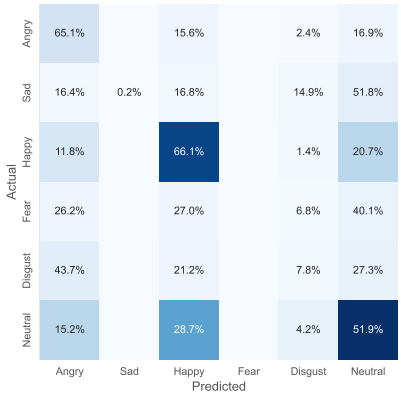
iEF Raw TS2



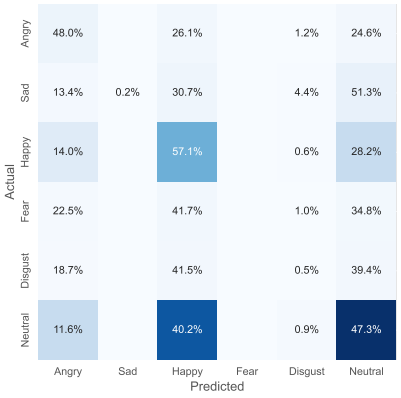
GDN Raw TS1



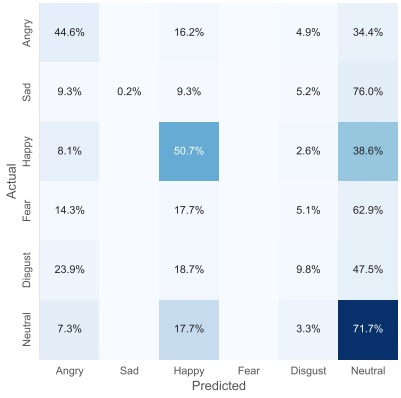
GDN Raw TS2



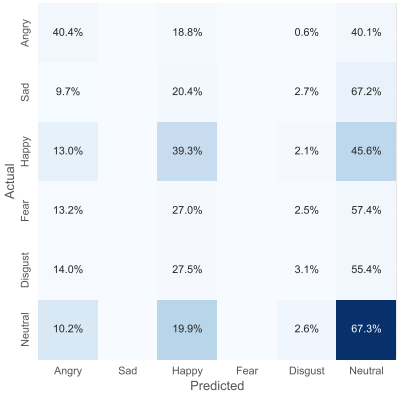
iEF Sampler TS1



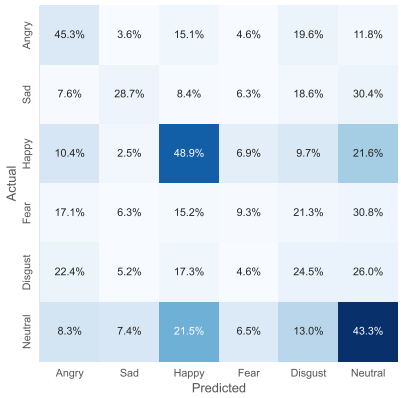
iEF Sampler TS2



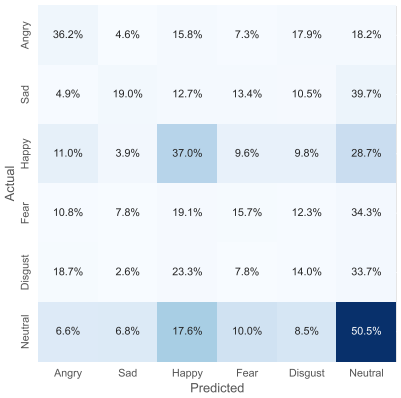
GDN Sampler TS1



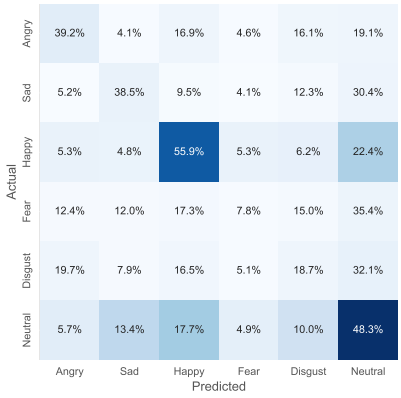
GDN Sampler TS2



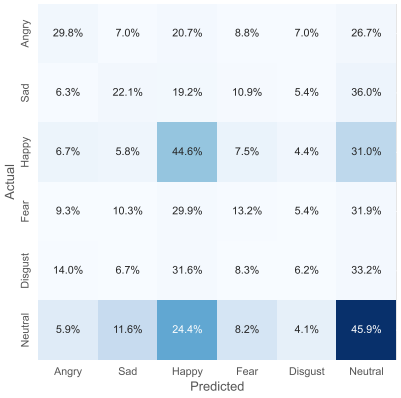
iEF IRW(Wu et al., 2024b) TS1



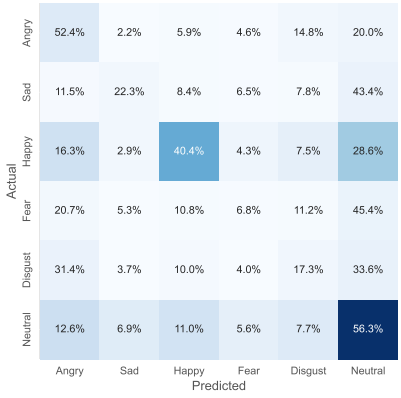
iEF IRW (Wu et al., 2024b) TS2



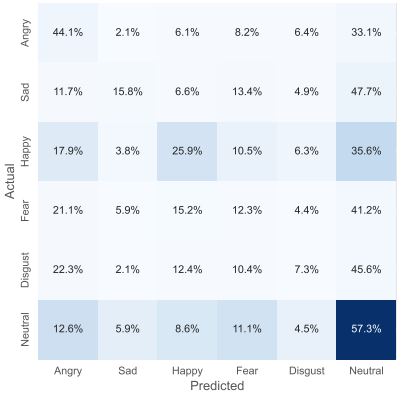
iEF DRW(Wu et al., 2024b) TS1



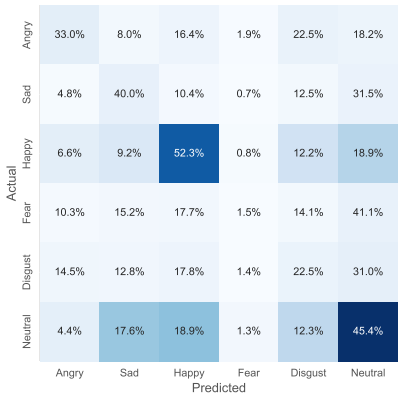
iEF DRW (Wu et al., 2024b) TS2



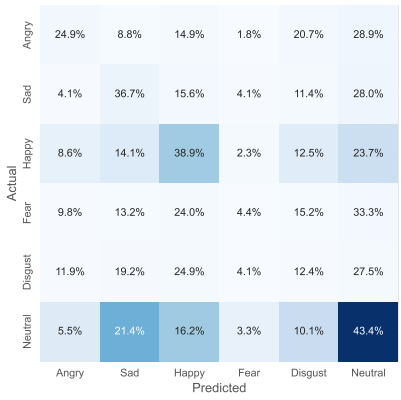
GDN IRW(Benzing, 2022) TS1



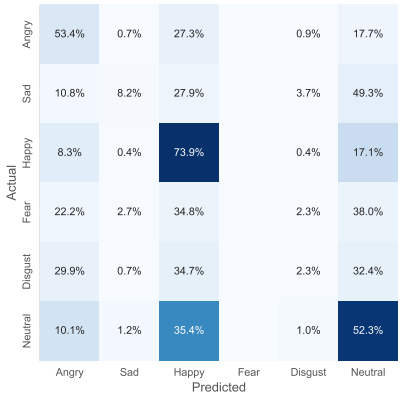
GDN IRW (Benzing, 2022) TS2



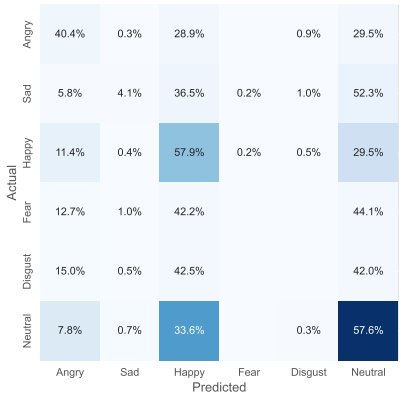
GDN DRW(Benzing, 2022) TS1



GDN DRW (Benzing, 2022) TS2



Sampler TS1



Sampler TS2

Fig. C.3 Confusion matrices for the MSP-Podcast test set $\alpha = 0.1$ across various CI correctors.