

Imperial College of Science, Technology and Medicine  
Department of Computing

**Extracting and comparing Machine Learning-based  
representations across natural and medical modalities**

Xavier Cadet

Submitted in part fulfilment of the requirements for the degree of  
Doctor of Philosophy in Computing and  
the Diploma of Imperial College, October 2024



# Declaration of Originality

I hereby confirm that, to the best of my knowledge, this thesis is my own work, and, except indicated appropriately by references, describes my own research. This thesis has not been submitted for any other purposes.

Xavier Cadet

October 2024

# Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

## Abstract

Machine learning has led to breakthroughs across many domains, such as Computer Vision, Natural Language Processing, and Speech Recognition. Given these breakthroughs, machine learning methods are being applied to more and more fields, such as healthcare, education, and legal services. However, the data required to train such models can be sensitive and limited, making it challenging to develop models in such fields. To ensure the applicability of Artificial Intelligence in such sensitive fields, we need efficient, reliable methods that consider privacy preservation. In this thesis, I aim to address these challenges by comparing representations, evaluating their robustness to incomplete data, and preserving privacy through machine unlearning methods. I introduce several key contributions across these domains: I propose the Feature Impact Balance (FIB) score, a metric to assess error distribution, and the Min-Max Relative Change Quadrant (MMRCQ) plot, a visualization tool to monitor changes in extreme values. I explore the class-wise impact of data augmentation in image-based models, showing that augmentations affect different classes unequally. Data augmentations can make classes less distinguishable, and such an effect is class-dependent. I evaluate the transferability of foundation models trained on non-medical data to domains with limited labeled data, demonstrating their potential for medical applications such as dysarthria automated assessments. I show that foundation models trained on large-scale speech from healthy individuals can be used to detect dysarthria, classify words, and classify intelligibility. I also assess the robustness of self-supervised learning (SSL) models to incomplete data. Finally, I benchmark machine unlearning methods, making models "forget" data points without losing overall performance. This benchmark shows that unlearning algorithms must be compared against stronger baselines with extensive hyper-parameter searches. This thesis advances the development of more robust, reliable, and privacy-conscious machine learning models.



## Acknowledgements

First and foremost, I would like to thank my parents and my sister for all the love and support that made me who I am.

I want to express my gratitude to my supervisors, Professor Hamed Haddadi and Doctor Sara Ahmadi-Abhari, for their guidance, support, and encouragement throughout my Ph.D. journey. I am thankful for the opportunity they gave me to work in such a wonderful lab and to interact with such great people. I would like to thank all of my talented co-authors and colleagues, Ranya Aloufi, Yushan Huang, Mohammad Malekzadeh, Anastasia Borovykh, Lorena Qendro, Sandra Siby, Miloud Bessafi and Alain Miranville, whose contributions and guidance were indispensable for the work presented here. I would like to thank all my colleagues at NetSys for all the great times we had together and the atmosphere they brought to the lab.

Lastly, I would like to thank all my friends and family for being there in good times and bad times and for always supporting me in my endeavors.

During my research, I received support from the UK Research and Innovation (UKRI Centre for Doctoral Training in AI for Healthcare, grant number EP/S023283/1).

# Contents

|          |                                                                                           |           |
|----------|-------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                                       | <b>22</b> |
| 1.1      | Motivation and Objective . . . . .                                                        | 22        |
| 1.2      | Contributions . . . . .                                                                   | 23        |
| 1.2.1    | Metric for Error Distribution: Feature Impact Balance (FIB) . . . . .                     | 23        |
| 1.2.2    | Visualization of Extreme Value Shifts: Min-Max Relative Change Quadrant (MMRCQ) . . . . . | 23        |
| 1.2.3    | Class-wise Evaluation of Data Augmentations in Image Data . . . . .                       | 24        |
| 1.2.4    | Transferability of Foundation Models from Healthy to Dysarthric Individuals . . . . .     | 24        |
| 1.2.5    | Robustness of Self-Supervised Learning (SSL) Models . . . . .                             | 25        |
| 1.2.6    | Mechanisms of Unlearning . . . . .                                                        | 25        |
| 1.3      | Publications . . . . .                                                                    | 25        |
| 1.4      | Thesis outline . . . . .                                                                  | 26        |
| <b>2</b> | <b>Background</b>                                                                         | <b>28</b> |
| 2.1      | Introduction . . . . .                                                                    | 28        |
| 2.1.1    | A brief introduction to Machine Learning . . . . .                                        | 28        |
| 2.1.2    | What is Deep Learning? . . . . .                                                          | 30        |

|          |                                                                                                                        |           |
|----------|------------------------------------------------------------------------------------------------------------------------|-----------|
| 2.1.3    | Further resources . . . . .                                                                                            | 31        |
| 2.2      | Challenges of Machine Learning . . . . .                                                                               | 31        |
| 2.2.1    | Computational cost . . . . .                                                                                           | 31        |
| 2.2.2    | Data availability . . . . .                                                                                            | 31        |
| 2.3      | Towards generalizable representations . . . . .                                                                        | 32        |
| 2.3.1    | Data Augmentations . . . . .                                                                                           | 32        |
| 2.3.2    | Self-Supervised Learning . . . . .                                                                                     | 33        |
| 2.3.3    | Foundation models . . . . .                                                                                            | 34        |
| 2.4      | Model comparison . . . . .                                                                                             | 35        |
| 2.4.1    | Downstream task performance . . . . .                                                                                  | 35        |
| 2.4.2    | Comparing Representations . . . . .                                                                                    | 36        |
| 2.5      | Privacy concerns . . . . .                                                                                             | 37        |
| 2.5.1    | Training set memorization . . . . .                                                                                    | 37        |
| 2.5.2    | Learning to unlearn . . . . .                                                                                          | 38        |
| <b>3</b> | <b>Comparing representations: Feature Impact Balance (FIB) score and Min-Max Relative Change Quadrant (MMRCQ) plot</b> | <b>40</b> |
| 3.1      | Introduction . . . . .                                                                                                 | 40        |
| 3.1.1    | Defining data representations . . . . .                                                                                | 41        |
| 3.1.2    | Why representations matter? . . . . .                                                                                  | 42        |
| 3.1.3    | Example of representation transformation: Distinguishing concentric circles                                            | 42        |
| 3.2      | Assessing balance of contributions: Feature Impact Balance (FIB) . . . . .                                             | 44        |
| 3.2.1    | Error Metrics . . . . .                                                                                                | 44        |

|          |                                                                                                            |           |
|----------|------------------------------------------------------------------------------------------------------------|-----------|
| 3.2.2    | Lack of insight into the error distribution . . . . .                                                      | 45        |
| 3.2.3    | Addressing the lack of insight into the error distribution: Feature Impact<br>Balance (FIB) score. . . . . | 46        |
| 3.2.4    | Limitations . . . . .                                                                                      | 51        |
| 3.2.5    | Perspectives . . . . .                                                                                     | 51        |
| 3.3      | Visualizing relative changes: Min-Max Relative Change Quadrant (MMRCQ) Plot                                | 52        |
| 3.3.1    | Limitations . . . . .                                                                                      | 53        |
| 3.3.2    | Perspectives . . . . .                                                                                     | 54        |
| <b>4</b> | <b>Data Augmentation affects classes differently</b>                                                       | <b>55</b> |
| 4.1      | Introduction . . . . .                                                                                     | 55        |
| 4.2      | Background . . . . .                                                                                       | 56        |
| 4.2.1    | Adding variability to the training data using Data Augmentation . . . .                                    | 56        |
| 4.2.2    | Data augmentation in the feature space rather than the input space . . .                                   | 59        |
| 4.2.3    | The distinction between label-preserving and label-breaking transforma-<br>tions . . . . .                 | 59        |
| 4.2.4    | Adversarial Training . . . . .                                                                             | 61        |
| 4.2.5    | Inter-Class Similarity . . . . .                                                                           | 61        |
| 4.2.6    | Contextual Bias in Image Classification . . . . .                                                          | 61        |
| 4.2.7    | Motivation for Novel Evaluation Metrics . . . . .                                                          | 62        |
| 4.3      | Methodology . . . . .                                                                                      | 62        |
| 4.3.1    | Class aggregated representations . . . . .                                                                 | 62        |
| 4.3.2    | Computing differences between aggregated representations . . . . .                                         | 63        |
| 4.3.3    | Computing similarities between aggregated representations . . . . .                                        | 63        |

|          |                                                                                                                                                |           |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 4.3.4    | Computing the error balance between aggregated representations . . . .                                                                         | 64        |
| 4.4      | Problem setup . . . . .                                                                                                                        | 65        |
| 4.5      | Experimental Setup . . . . .                                                                                                                   | 65        |
| 4.5.1    | Datasets . . . . .                                                                                                                             | 65        |
| 4.5.2    | Data augmentations . . . . .                                                                                                                   | 66        |
| 4.5.3    | Aggregated representations . . . . .                                                                                                           | 69        |
| 4.6      | Results . . . . .                                                                                                                              | 70        |
| 4.6.1    | Aggregated representations of the original dataset . . . . .                                                                                   | 70        |
| 4.6.2    | How do data augmentations impact the similarities of aggregated representations? . . . . .                                                     | 74        |
| 4.6.3    | Do data augmentations impact all classes the same way? . . . . .                                                                               | 74        |
| 4.7      | Discussion . . . . .                                                                                                                           | 80        |
| 4.7.1    | Conclusion . . . . .                                                                                                                           | 80        |
| 4.7.2    | Limitations . . . . .                                                                                                                          | 81        |
| 4.7.3    | Perspectives . . . . .                                                                                                                         | 81        |
| <b>5</b> | <b>Leveraging foundation models trained on non-medical data to conduct multiple classification tasks for automated dysarthria assessments.</b> | <b>83</b> |
| 5.1      | Introduction . . . . .                                                                                                                         | 83        |
| 5.1.1    | Challenges of medical applications . . . . .                                                                                                   | 83        |
| 5.1.2    | Foundation models and downstream tasks . . . . .                                                                                               | 84        |
| 5.1.3    | Our contributions . . . . .                                                                                                                    | 84        |
| 5.2      | Background . . . . .                                                                                                                           | 85        |
| 5.2.1    | What is Dysarthria? . . . . .                                                                                                                  | 85        |

|       |                                                                                                                                   |           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|-----------|
| 5.2.2 | Towards automated dysarthria assessment . . . . .                                                                                 | 86        |
| 5.2.3 | Learning general-purpose representations via Self-Supervised Learning . .                                                         | 87        |
| 5.2.4 | Transferring self-supervised representations to dysarthria assessment . . .                                                       | 87        |
| 5.3   | Methodology . . . . .                                                                                                             | 88        |
| 5.3.1 | Problem Setup . . . . .                                                                                                           | 88        |
| 5.3.2 | Overview . . . . .                                                                                                                | 89        |
| 5.3.3 | Experimental Settings . . . . .                                                                                                   | 89        |
| 5.4   | Experiments . . . . .                                                                                                             | 92        |
| 5.4.1 | Original: <i>How reliable are classifiers when confronted with unknown speakers?</i> . . . . .                                    | 92        |
| 5.4.2 | Noisy Control: <i>Are models trained on the Original dataset biased by patterns related to the noise in recordings?</i> . . . . . | 95        |
| 5.4.3 | Enhanced: <i>What impact does enhancing the recordings have over the different tasks?</i> . . . . .                               | 95        |
| 5.5   | Discussion . . . . .                                                                                                              | 96        |
| 5.5.1 | Conclusion . . . . .                                                                                                              | 96        |
| 5.5.2 | Limitations . . . . .                                                                                                             | 97        |
| 5.5.3 | Perspectives . . . . .                                                                                                            | 98        |
| 6     | <b>Assessing the Robustness of Self-Supervised Learning Representations to Incomplete Data</b> . . . . .                          | <b>99</b> |
| 6.1   | Introduction . . . . .                                                                                                            | 99        |
| 6.2   | Background . . . . .                                                                                                              | 100       |
| 6.2.1 | Self-Supervised Learning (SSL): learning by leveraging signal from unlabeled data . . . . .                                       | 100       |

|       |                                                                                                                                                                 |     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.2.2 | Learning representation by masking parts of the input . . . . .                                                                                                 | 101 |
| 6.2.3 | Evaluating SSL on downstream tasks . . . . .                                                                                                                    | 102 |
| 6.2.4 | Evaluating the similarity of representations . . . . .                                                                                                          | 102 |
| 6.3   | Methodology . . . . .                                                                                                                                           | 106 |
| 6.3.1 | Problem Setup . . . . .                                                                                                                                         | 106 |
| 6.3.2 | Removing information from images . . . . .                                                                                                                      | 106 |
| 6.3.3 | Variations of the dataset . . . . .                                                                                                                             | 107 |
| 6.3.4 | Linear Evaluation: How well can we classify using the extracted representations? . . . . .                                                                      | 108 |
| 6.3.5 | Representation Similarity: Are variations of the same image leading to similar representations? . . . . .                                                       | 109 |
| 6.3.6 | Representation Neighborhood: Are samples in the representations neighboring samples from the same variants? . . . . .                                           | 109 |
| 6.4   | Experiments . . . . .                                                                                                                                           | 109 |
| 6.4.1 | The representations contain information about the context. . . . .                                                                                              | 110 |
| 6.4.2 | Representations between Original and Center are more similar in Caltech-101 than Original and Foreground based on Canonical Correlation Analysis (CCA). . . . . | 112 |
| 6.4.3 | Foreground and Background are more similar than Original and Foreground for SwAV, VICReg, and MAE based on Centered Kernel Alignment (CKA). . . . .             | 112 |
| 6.4.4 | Representations from different variants are close to each other. . . . .                                                                                        | 113 |
| 6.5   | Discussion . . . . .                                                                                                                                            | 113 |
| 6.5.1 | Conclusion . . . . .                                                                                                                                            | 113 |
| 6.5.2 | Limitations . . . . .                                                                                                                                           | 114 |

|          |                                                       |            |
|----------|-------------------------------------------------------|------------|
| 6.5.3    | Perspectives . . . . .                                | 114        |
| <b>7</b> | <b>Learning to Unlearn</b>                            | <b>116</b> |
| 7.1      | Introduction . . . . .                                | 116        |
| 7.2      | Background . . . . .                                  | 118        |
| 7.2.1    | Machine Unlearning . . . . .                          | 118        |
| 7.2.2    | Machine Unlearning for Deep Neural Networks . . . . . | 120        |
| 7.2.3    | Post-Hoc Machine Unlearning . . . . .                 | 120        |
| 7.2.4    | Machine Unlearning and Differential Privacy . . . . . | 121        |
| 7.3      | Methodology . . . . .                                 | 121        |
| 7.3.1    | Machine Unlearning Methods . . . . .                  | 124        |
| 7.3.2    | Datasets . . . . .                                    | 126        |
| 7.3.3    | Neural Network Architectures . . . . .                | 128        |
| 7.3.4    | Privacy Evaluation . . . . .                          | 129        |
| 7.4      | Experimental Evaluation . . . . .                     | 130        |
| 7.4.1    | Main Results . . . . .                                | 131        |
| 7.4.2    | Per dataset results . . . . .                         | 134        |
| 7.4.3    | Per architectures rankings . . . . .                  | 144        |
| 7.4.4    | L2 Distances between model weights . . . . .          | 144        |
| 7.5      | Discussion . . . . .                                  | 145        |
| 7.5.1    | Conclusion . . . . .                                  | 145        |
| 7.5.2    | Limitations . . . . .                                 | 147        |
| 7.5.3    | Perspectives . . . . .                                | 147        |

|          |                                                                                                                       |            |
|----------|-----------------------------------------------------------------------------------------------------------------------|------------|
| 7.6      | Requirements . . . . .                                                                                                | 147        |
| <b>8</b> | <b>Contributions and Perspectives</b>                                                                                 | <b>148</b> |
| 8.1      | Conclusion and Contributions . . . . .                                                                                | 148        |
| 8.1.1    | Comparison tools: Feature Impact Balance (FIB) and Min-Max Relative<br>Change Quadrant (MMRCQ) . . . . .              | 149        |
| 8.1.2    | Class-level data augmentation impact . . . . .                                                                        | 149        |
| 8.1.3    | Analysis of the transferability of healthy speech foundation models for<br>automated dysarthria assessments . . . . . | 150        |
| 8.1.4    | Evaluation of the robustness of Self-Supervised visual feature extractors<br>to corrupted inputs . . . . .            | 150        |
| 8.1.5    | Evaluation of the Machine Unlearning methods across datasets, initial-<br>izations . . . . .                          | 150        |
| 8.2      | Insights toward the application of Artificial Intelligence in Healthcare . . . . .                                    | 151        |
| 8.3      | Perspectives . . . . .                                                                                                | 152        |
| 8.3.1    | Feature Impact Balance and Min-Max Relative Change Quadrant . . . . .                                                 | 152        |
| 8.3.2    | Data Augmentations . . . . .                                                                                          | 152        |
| 8.3.3    | Dysarthria Automated Assessments . . . . .                                                                            | 152        |
| 8.3.4    | Self-Supervised Learning Robustness . . . . .                                                                         | 153        |
| 8.3.5    | Machine Unlearning . . . . .                                                                                          | 153        |

# List of Tables

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.1 | Example of Tabular Data Representation . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                       | 41  |
| 4.1 | Quadrant Attribution for Data Augmentation effects of MSE and FIB score on<br>CIFAR-100 . . . . .                                                                                                                                                                                                                                                                                                                                                                                      | 80  |
| 5.1 | Task Performance: results are reported using Accuracy, using the Leave-One-Speaker-Out cross-validation. For the Disease and Words classification problems, results are based on individual recordings. For the Intelligibility task, results are calculated performance at the speaker level. . . . .                                                                                                                                                                                 | 93  |
| 6.1 | Linear Probing: results are 5-fold cross-validated Balanced Accuracy. DINOv2 is consistently the best performing model. Except of Caltech-101, the supervised performs worse than the self-supervised extractors. On every dataset, the <b>Background</b> representations can be used to train classifiers with much higher than random performance. FGVC-Aircraft is the only dataset where the <b>Foreground</b> performance is much higher than the <b>Original</b> images. . . . . | 115 |
| 7.1 | Summary of the number epochs, learning rate, and batch size for each dataset and model used to train the Original and Retrained models. . . . .                                                                                                                                                                                                                                                                                                                                        | 128 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 7.2  | Ranking by performance on Retention Deviation and Indiscernibility across datasets and architectures. We count the number of times each method appears in the Best Performers group (G1), Average performance group (G2) and Worst performers group (G3) (see §7.4). The final rank is computed based on the number of times the method appears in G1—with occurrences in G2 and G3 used to break ties if needed. If a method does not produce any usable models, it is assigned to a Failed group (F). Three methods appear in the top 3 for both performance measures: MSG (1st and 1st), CT (3rd and 1st), and KDE (3rd and 2nd). . . . . | 132 |
| 7.3  | Run Time Efficiency on ResNet for the top performing methods. CT is the fastest on average, MSG runs up to 5x faster than naive retraining. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 133 |
| 7.4  | MNIST - ResNet18: Retain Accuracy (RA), Forget Accuracy (FA), Test Accuracy (TA), Retain Retention (RR), Forget Retention (FR), Test Retention (TR), Performance Retention Deviation (RetDev), Indiscinerbility concerning the Test Set (Indisc), U-MIA on the Test set (T-MIA) and RunTime Efficiency (RTE) . .                                                                                                                                                                                                                                                                                                                             | 135 |
| 7.5  | FashionMNIST - ResNet18 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 136 |
| 7.6  | CIFAR-10 - ResNet18 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 137 |
| 7.7  | CIFAR-100 - ResNet18. CIFAR-100 provides the most visible comparison as there is a large gap in performance between the Retain Set and Test set, this leads to much larger RetDev scores. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                            | 138 |
| 7.8  | UTKFace - ResNet18 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 139 |
| 7.9  | MNIST - TinyViT . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 140 |
| 7.10 | FashionMNIST - TinyViT . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 141 |
| 7.11 | CIFAR-10 - TinyViT . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 142 |
| 7.12 | CIFAR-100 - TinyViT . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 143 |
| 7.13 | Ranking on ResNet . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 144 |
| 7.14 | Ranking of ViT . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 145 |

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | Two concentric circles with different radii using different representations: Cartesian Coordinates (left), Polar Coordinates (center), Feature Combination (right). Each of the representation can be derived from the Cartesian coordinates, nevertheless, using the Polar Coordinates (center) or the Feature Combination (right), one could draw a linear separation between the set of points of both circles. . . .                                                                                                                                                                                                                                                                                                                           | 43 |
| 3.2 | Comparison between Mean Squared Error (MSE) and Feature Impact Balance (FIB), illustrating how differences are distributed across vector positions. Given a reference vector and three other vectors, we compute the MSE between the reference vector and each comparison vector. While the MSE remains the same for all three comparisons, the distribution of differences across vector positions varies. These differences are captured by the FIB, which are 0.0, 0.66, and 1.0 from left to right, respectively. The FIB values indicate that: (left) a single position carries all the difference, (center) a subset of positions carries the differences, and (right) the differences are distributed equally across all positions. . . . . | 46 |
| 3.3 | Visualization of the Feature Impact Balance (FIB) score as a function of feature contributions. The left plot shows the FIB score as a function of the contribution $c_1$ for two features, $c_1$ and $1 - c_1$ , highlighting the maximum FIB score and the balanced point where $c_1 = 0.5$ . The right plot extends this to three features, $c_1$ , $c_2$ , and $1 - c_1 - c_2$ , showing how FIB scores vary as contributions are distributed across multiple features. Higher FIB scores correspond to more balanced feature contributions. . . . .                                                                                                                                                                                           | 51 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.4 | Min-Max Relative Change Quadrant (MMRCQ): Based on the localization of a point in the MMRCQ plot we can infer changes in the minimal and maximal values of a property with respect to a baseline. Points located in the upper quadrants indicate an increase in the maximal value while points located in the lower quadrants indicate a decrease in the maximal value. Points located in the left quadrants indicate lower minimal values, and points located in the right quadrants indicate higher minimal values. . . . . | 52 |
| 4.1 | Label-breaking transformation: Starting from a 9, if we flip both horizontally and vertically, we obtain 6. This highlights that data augmentations requires careful consideration depending on the data. (Images from MNIST[123]) . . . .                                                                                                                                                                                                                                                                                    | 60 |
| 4.2 | Label-preserving transformation: A cat remains a cat after the different transformations. (Images from Oxford IIIT Pet[143]) . . . . .                                                                                                                                                                                                                                                                                                                                                                                        | 60 |
| 4.3 | Flowchart of the representation aggregation pipeline on CIFAR-100 using the strongest HUE perturbation. Given a dataset, we consider a single pass through the data augmentation method. Therefore, we obtain a single version of an image $x$ , its transformed versions $\phi(x)$ . Then, we regroup all images that belong to the same class. Finally, we aggregate the transformed representations for each class.                                                                                                        | 69 |
| 4.4 | Aggregated Representation over CIFAR-10's training set. Each square represents the average representation for a given class. For instance, the top-left element represents the average representation of the "airplane" classes. We can observe that, even based on the average representation, both airplane and boat classes exhibit similar color schemes with more nuances of blue compared to other classes.                                                                                                             | 71 |
| 4.5 | Aggregated Representation over CIFAR-100's training set. Each square represents the average representation for a given class. For instance, the top-left element represents the average representation of the "airplane" classes. We can observe that, even based on the average representation, both airplane and boat classes exhibit similar color schemes with more nuances of blue compared to other classes. . . . .                                                                                                    | 72 |
| 4.6 | Examples of data augmentations applied to CIFAR-10 . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 73 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.7  | Normalized difference between aggregated representations of classes of CIFAR-10. We observe that "Airplane" and "Ship" classes have differences with other classes that are mostly from the blue channel. This is due to the context in which the different pictures are taken, as these classes more likely to be against the sky or water. . . . .                                                                                                                            | 75 |
| 4.8  | Density of Structural Similarity Index (SSIM) values between aggregate representations on CIFAR-100. Most data augmentations do not change the similarity between aggregated representations. The distribution are left skewed with peaks within $[0.6, 0.8]$ . . . . .                                                                                                                                                                                                         | 76 |
| 4.9  | CIFAR-10 MMRCQ plot between the original data and the data after applying vertical flipping. When considering the MSE scores, the "truck" class has the strongest relative changes in its minimal and maximal distances from other classes. . . . .                                                                                                                                                                                                                             | 77 |
| 4.10 | CIFAR-100 Relative Change Quadrant plot between the original and the training sets post vertical flip augmentation. When considering the MSE scores, the truck class has the strongest relative impact on its differences with the aggregated representation of other classes. . . . .                                                                                                                                                                                          | 78 |
| 5.1  | Our evaluation pipeline. We start by generating the different versions of the dataset then use different feature extractors to obtain representations of the audio recordings. Using these representations we train diverse classification models and predict different properties. Finally, we perform patient level aggregation, for instance for the intelligibility level we consider the intelligibility class that is the most associated with the given patient. . . . . | 89 |
| 5.2  | Audio Settings: Example of the same recording under different audio settings. . . . .                                                                                                                                                                                                                                                                                                                                                                                           | 92 |
| 5.3  | Speaker Intelligibility across datasets (columns) and features (rows). All predictions are made using logistic regression-based classifiers. While the performance based on HuBERT-extracted features is higher than acoustic, for a given speaker, there are major mis-classifications at the recording level. . . . .                                                                                                                                                         | 94 |

|     |                                                                                                                                                                                                                                                                                                                                                                    |     |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.1 | Examples of the 5 variations explored. On the left we show the <b>Original</b> , <b>Foreground</b> , and <b>Background</b> from top to bottom, on CIFAR-100, Caltech101, FGVC Aircraft. On the right we show the <b>Original</b> , <b>Center</b> , <b>Border</b> from top to bottom, CIFAR-100, DTD and DTD . . . . .                                              | 107 |
| 6.2 | Mean Squared CCA correlation $\mathbb{R}_{CCA}^2$ . The first row is based on CIFAR-100, the second row on Caltech101. From left to right, the matrices are associated with the representation extracted by the Supervised SwAV, VICReg, MAE, and DINOv2 models. <b>Original</b> , <b>Foreground</b> , <b>Background</b> , <b>Center</b> , <b>Border</b> . . . . . | 110 |
| 6.3 | CKA scores. The first row is based on CIFAR-100, the second row on Caltech101. The matrices are associated with Supervised, SwAVm VICReg, MAE, and DINOv2 from left to right. <b>Original</b> , <b>Foreground</b> , <b>Background</b> , <b>Center</b> , <b>Border</b> . . . . .                                                                                    | 110 |
| 6.4 | The first row indicates how each variant neighboring points are related to other variants. These observations are based on the representations extracted on CIFAR-100. We consider the Supervised, DINOv2, and MAE models from left to right. The second row depicts the UMAP obtained on 10% of the representations.                                              | 111 |
| 7.1 | U-LiRA on CIFAR-10 on ResNet18 models. Both CT and MSG, which ranked first against U-MIA, showed great resilience against the U-LiRA attack. . . . .                                                                                                                                                                                                               | 134 |
| 7.2 | L2 Distance between the Unlearned ResNet18 models, the Original and Retrained models. None of the unlearned models gets close to the Retrained model's weights; most unlearned Models are closer to the Original model than the Retrained model. . . . .                                                                                                           | 146 |

# Chapter 1

## Introduction

### 1.1 Motivation and Objective

Machine learning has rapidly advanced fields such as Computer Vision, Natural Language Processing, and Speech Recognition, enabling transformative applications in assistive technologies [122, 121], healthcare domains such as automated medical image segmentation [176, 129, 55], cancer detection [62], and clinical disciplines like radiology [178] and ophthalmology [50]. Part of the underlying progress comes from innovations in representation learning and model robustness, yet the need for privacy-preserving machine learning has also increased.

This thesis aims to address these challenges by investigating critical aspects of machine learning models, focusing on the effect of data augmentations, analyzing the robustness of self-supervised models to incomplete data, and evaluating the efficacy of machine unlearning methods to preserve privacy. We present new methodologies and evaluations of these aspects.

We explore the following research question: *How can machine learning models be improved regarding representation learning, robustness to incomplete data, and privacy preservation through machine unlearning?*

To answer this question, we address the following in different chapters:

- **Representation Comparison:** How can we effectively assess and compare the quality of data representations learned by different models? (Chapters 3, 4 and 6)

- **Foundation Model Transfer:** Can non-medical foundation models be successfully adapted for medical tasks, especially with limited labeled data? (Chapter 5)
- **Robustness of Self-Supervised Models:** How resilient are self-supervised learning representations to incomplete or corrupted input data? (Chapter 6)
- **Machine Unlearning for Privacy:** How can machine unlearning help models forget specific data points while maintaining overall performance and, as such, address privacy concerns? (Chapter 7)

Through these analyses, we advance machine learning toward more reliable, robust, and privacy-aware models, particularly in domains that face limited data availability and privacy concerns.

## 1.2 Contributions

### 1.2.1 Metric for Error Distribution: Feature Impact Balance (FIB)

The first contribution of this thesis is the development of Feature Impact Balance (FIB), a novel metric designed to assess the distribution of errors between two vectors. I derived this metric from the observation that not all errors are equally important: small, equally distributed errors have a different impact than fewer but more significant differences. FIB supplements traditional metrics like the Mean Squared Error (MSE) by providing a more nuanced view of error distribution, offering a more comprehensive understanding of data representation. The score is bounded between  $[0, 1]$ , ensuring intuitive interpretation. Additionally, it can be customized to suit different use cases by selecting appropriate contribution vectors. We introduced this metric in Cadet et al. [20], and describe it in detail in Chapter 3, and its application is demonstrated in Chapter 4.

### 1.2.2 Visualization of Extreme Value Shifts: Min-Max Relative Change Quadrant (MMRCQ)

The second contribution is the Min-Max Relative Change Quadrant (MMRCQ), a visualization tool developed to track changes in the range of extreme values across different settings, such

as time points or datasets. This tool provides insights into shifts in extreme values within the data. It can be helpful in fields such as finance or climate studies, where monitoring rare but significant changes matter. Chapter 3 explains the development of MMRCQ, and a practical application in evaluating the impact of data augmentation on images is presented in Chapter 4. The method is adaptable and transferable to various domains, as it can be used to track extreme values over time or across datasets in diverse fields.

### 1.2.3 Class-wise Evaluation of Data Augmentations in Image Data

The third contribution involves the evaluation of the class-wise impact of data augmentations on image data. Data augmentation is a widely used technique in computer vision. It enhances the generalization of machine learning models by increasing dataset variability. Nonetheless, augmentations can degrade model performance for specific classes. Traditional machine learning pipelines often apply augmentations uniformly without considering their effect on individual class representations. In Chapter 4, we demonstrate that data augmentations can alter class similarity by bringing the representations of different classes closer, potentially reducing their distinguishability. We use both the FIB and MMRCQ metrics to assess these effects across a range of commonly used augmentations.

### 1.2.4 Transferability of Foundation Models from Healthy to Dysarthric Individuals

The fourth contribution investigates the transferability of large-scale foundation models from general populations to dysarthric individuals, a domain with limited data availability. Obtaining reliable deep-learning models in medical settings is challenging due to the need for labeled data. However, pre-trained foundation models on large non-medical datasets offer the potential for transferring generalized representations. We evaluated several foundation models to assess their effectiveness in dysarthric speech analysis. This work is presented in Cadet et al. [21], and discussed in Chapter 5. This work shows that one can successfully transfer these models to perform automated dysarthria detection, word recognition, and intelligibility classification despite being trained on non-medical data.

### 1.2.5 Robustness of Self-Supervised Learning (SSL) Models

Another contribution is the evaluation of the robustness of Self-Supervised Learning (SSL) models. SSL models are hypothesized to generalize better than their supervised counterparts by leveraging unlabeled data and avoiding overfitting specific tasks. To test this, we evaluated the impact of removing the object from the images, assuming that the classification performance should decrease without the object of interest. Our analysis, presented in Cadet et al. [22] and detailed in Chapter 6, shows that in some cases, high performance is still achievable, suggesting that dataset-specific characteristics, such as lack of diversity in the background, play a significant role in model performance.

### 1.2.6 Mechanisms of Unlearning

The final contribution concentrates on understanding machine unlearning in deep neural networks. We focus on aspects relevant to privacy regulations requiring data removal from trained models. Unlearning is an emerging field driven by the need to ensure that models can reliably forget sensitive data without degrading overall model performance. This research, discussed in Chapter 7 and presented in Cadet et al. [23], benchmarks various machine unlearning techniques and assesses their efficiency in removing specific data points while preserving the accuracy of the model.

## 1.3 Publications

#### First-Author

- Xavier F. Cadet, Sara Ahmadi-Abhari, and Hamed Haddadi. FIB: A Method for Evaluation of Feature Impact Balance in Multi-Dimensional Data, July 2022. arXiv:2207.04500 [20]
- Xavier F. Cadet, Ranya Aloufi, Sara Ahmadi-Abhari, and Hamed Haddadi. A study on the impact of self-supervised learning on automatic dysarthric speech assessment. In 2024 IEEE international conference on acoustics, speech, and signal processing workshops (ICASSPW), pages 630–634, 2024. [21]

- Xavier F. Cadet, Ranya Aloufi, Alain Miranville, Sara Ahmadi-Abhari, and Hamed Haddadi. Evaluating The Robustness of Self-Supervised Representations to Background/Foreground Removal, June 2023. arXiv:2306.01398 [22]
- Xavier F. Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep Unlearn: Benchmarking Machine Unlearning, October 2024. arXiv:2410.01276 [23]
- Xavier F. Cadet, Manyue Hu, Hamed Haddadi and Sara Ahmadi-Abhari for the Alzheimer’s Disease Neuroimaging Initiative. Enhanced Prediction of Cognitive Decline Using Structural MRI. (Under writing)

## Co-Author

- Yushan Huang, Ranya Aloufi, Xavier Cadet, Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. Low-Energy On-Device Personalization for MCUs, October 2024. arXiv:2403.08040 [103]

## 1.4 Thesis outline

This thesis is organized as follows:

- Chapter 2 provides the reader with the general background for representation comparison. Nonetheless, each following chapter also introduces more relevant aspects of its domain.
- Chapter 3 presents the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) plot, providing an overview of their design and how they can be used to compare representations.
- Chapter 4 covers the issue of data augmentation and how its application can impact classes more than others.
- Chapter 5 covers the transferability of foundation models to a new domain and how they can be used to improve performance on limited data.

- Chapter 6 covers the robustness of self-supervised learning models when faced with corrupted inputs with missing parts of the image.
- Chapter 7 covers the issue of machine unlearning and how it can be used to remove specific data points from a model and provide a benchmark of contemporary approaches.
- Chapter 8 concludes the thesis and provides an overview of the main findings and contributions.

# Chapter 2

## Background

In the following chapter, we provide a general introduction to Machine Learning and Deep Learning (Section 2.1); We then discuss two overarching challenges in Machine Learning (Section 2.2); We cover aspects of generalizability of machine learning representations (Section 2.3), model comparison (Section 2.4) and privacy in machine learning (Section 2.5). Then, for each subsequent chapter, we also provide an overview of the key concepts that are the most relevant.

### 2.1 Introduction

The amount of data collected in recent decades has greatly increased, partially due to new means of storing and sharing information, advances in storage capacity, and the development of the Internet. Some fields have greatly benefited from the influx of data shared on the Internet, either through social media platforms or user-generated content, as it can serve to build large-scale training datasets for Machine Learning models. Nonetheless, other domains, such as industrial applications or healthcare and medicine, still face various challenges, such as the difficulty of accessing large-scale curated labeled datasets.

#### 2.1.1 A brief introduction to Machine Learning

Machine Learning has existed for decades and led to breakthroughs in various fields such as Computer Vision (CV) and Natural Language Processing (NLP). Tom Mitchell [135] defined

Machine Learning as the study of computer algorithms that improve automatically through experience. Nonetheless, following Sutton and Barto's view [165], Machine Learning is simultaneously a problem, a class of methods that work well on that problem, and the field that studies this problem and its solutions.

Machine Learning pipelines rely on determining a good way to describe the data. Assume you are given a picture of a number and tasked with describing what you see without naming the number explicitly. Few people would describe the number based on individual pixels. Instead, one would commonly refer to different observed shapes, such as the number of loops, the curvature, or sharp angles of some portions of the number. Such summarized patterns are sometimes described as features; determining them is generally called feature engineering. Feature engineering relies on knowledge about the task, frequently called domain knowledge. Deep learning led to a paradigm shift because it simplified the overall feature engineering process. Deep learning allows for obtaining these descriptions as features or representations with less direct human involvement by learning them while being fed data. Deep Learning models have led to breakthroughs in fields where explaining the process that occurs naturally in humans is difficult[84]. Most individuals can recognize someone's face. Nonetheless, explaining the actual process is complex.

## Different types of Machine Learning

Machine Learning is often divided into three major categories: Supervised, Unsupervised, and Reinforcement Learning.

**Supervised Learning** describes scenarios where the label is available. For instance, we are given pictures of an animal and the type of animal present in the image. For example, a picture of a cat and its associated label 'cat.' Supervised learning aims to train a model to learn how to map the input data and the labels.

**Unsupervised Learning** describes scenarios where the label is unavailable. We are given many medical records without specific information about the outcome. The aim is to leverage the data to learn the underlying structure from the data.

**Reinforcement Learning** describes scenarios where we consider an agent in an environment, such that the agent learns through interaction with the environment. For instance, a robot learns to navigate in its environment.

For further information about these different types we refer the readers to the works of Mitchell [135], Hastie et al., [92] Bishop [14] and Sutton et al., [165].

### 2.1.2 What is Deep Learning?

Deep learning is a subset of machine learning based on Artificial Neural Network (ANN), a method involving multiple neuron layers. Deep learning has gained significant interest in the past decade, leading to multiple breakthroughs in various tasks in Computer Vision (CV) and Natural Language Processing (NLP).

A key concept of deep learning is stochastic gradient descent (SGD), an optimization algorithm widely used in training deep learning models. SGD is a variant of the traditional gradient descent algorithm. Instead of calculating the gradient using the entire dataset, it approximates it using a randomly selected subset (or even a single sample). This stochasticity introduces noise in the optimization process, allowing the algorithm to escape local minima and potentially find better solutions. SGD is particularly well-suited for deep learning due to its computational efficiency and ability to handle large-scale datasets, which are common in deep learning applications. Formally, given a loss function  $L(\theta)$ , the parameter updates for SGD are computed as follows:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t) \quad (2.1)$$

With  $\theta_t$  representing the model parameters at step  $t$ ,  $\eta$  is the learning rate, and  $\nabla_{\theta} L(\theta_t)$  is the loss function gradient of the parameters. In practice, mini-batch SGD is often used, which processes small batches of data at each iteration instead of individual samples [84]. Through this process, the weights on the neural networks are trained based on the errors, allowing the model to adapt its weight to solve the problem better. This is a critical strength of Deep Neural Networks, as they can learn to extract representations that help them solve the task for which they are trained.

### 2.1.3 Further resources

For a deeper understanding of deep learning, we refer the readers to the works of Goodfellow et al. [84] and Zhang et al. [196].

## 2.2 Challenges of Machine Learning

Two aspects are ascribed to the fast development of Machine Learning and Deep Learning: access to more computing resources and more data. Nonetheless, these aspects are also becoming limiting factors.

### 2.2.1 Computational cost

While computing capacities have increased over the past decades thanks to the development of specialized computing equipment such as Graphical Processing Units (GPU) and Tensor Processing Units (TPU), the amount of resources required to train some of the state-of-the-art methods is becoming prohibitive for many actors. For instance, DINOv2 [142] reports an estimated 22,000 GPU hours, thus close to 3 years of GPU computing, and LAMA [168] reports an estimated 1,000,000 GPU hours so more than a century of GPU hours. Nonetheless, as these models are shared with the broader community, the field’s potential for growth and innovation is vast. The increasing cost of training, whether time-wise, finance-wise, or computation-wise, also has implications when potential biases or wrongful behavior are detected, as it is currently challenging to remove such behavior from the models.

### 2.2.2 Data availability

Despite driving the advances of Deep Learning, data availability remains a limiting factor for diverse fields, such as healthcare, where the sensitivity of the data, the need for specialized equipment, and expert labeling make it challenging to obtain publicly available large-scale labeled datasets [38]. Furthermore, despite routinely creating large amounts of clinical data, these data points are often labeled with a specific aim: providing the appropriate information

to help treat the patient. This implies that while an MRI scan exists, it is unlikely to be labeled with tissue segmentation or disease classification tasks if they were irrelevant at collection time.

Nonetheless, one aim of Deep Learning methods is to extract generalizable representations, that is, representations that can be reused and transferred to other tasks and domains.

## 2.3 Towards generalizable representations

A key goal in machine learning is to develop models that can perform well on unseen data [135, 51]. Such models are said to generalize to new data. In contrast, models that only perform well on their training data points are said to overfit [135]. To mitigate overfitting, a set of approaches called regularization methods are commonly used [196]. Regularization is a modification we apply to an algorithm to reduce its generalization error but not its training error [84]. Common approaches are L2-Regularization or Weight Decay for Deep Neural Networks, Dropout [162], and Data Augmentation. Weight Decay consists of restricting the values the parameters of a machine learning model can take [196], Dropout deactivates neurons randomly during the training phase, and data augmentations perturb the input data.

### 2.3.1 Data Augmentations

Data Augmentations generally involve applying a perturbation to the input data to generate new data points that can be used to train the model [196]. In computer vision, such augmentations can comprise rotating the images, mirroring the image according to a given axis, and modifying the colors within the image [157]; in addition to these simple transformations, there exist more complex techniques that merge input points or attempt to generate new data points using generative models.

**Increasing the dataset size, introducing variability, and strengthening decision boundaries.** By generating similar yet distinct versions of the input, we can artificially increase the size of the dataset. Additionally, the modification applied to the data can help introduce variability. For instance, one can make the model recognize objects from different angles that might not have been present in the original dataset. Furthermore, an augmentation

that slightly perturbs the inputs can help strengthen decision boundaries [158]. Nonetheless, data augmentations are often dataset-specific and hand-crafted to incorporate relevant prior information[140, 18]

**Data augmentation often requires careful consideration and domain knowledge** Depending on the task and the data type, one can require drastically different data augmentations. For example, for text data, Shorten et al. [158] distinguish between symbolic augmentations and neural augmentations. That is augmentations that use symbolic rules versus augmentations that leverage additional neural networks. An example of symbolic augmentations is Easy Data Augmentations (EDA) by Wei et al. [181], which consists of synonym replacement, random insertion, random swap, and random deletion. An example of neural augmentation is generative data augmentation via methods such as Conditional BERT (C-BERT) [184], which augments the data by replacing masked tokens from the original input while using class-related information. For image data, Shorten et al. [157] distinguishes between basic image manipulation and deep learning-based augmentations. Among basic image manipulation, there are geometric transformations (mirroring the image along an axis, rotating the image, and random cropping), color space transformations (alteration of brightness, contrast, or hue), and image mixing [104] (fusion of pairs of images). Examples of deep learning-based augmentations are feature space augmentations [54], where the augmentations are done in the feature space of the model instead of the input space, or Neural Style Transfer [76] which attempts to transfer the style of an image onto another while preserving the content of the target image.

### 2.3.2 Self-Supervised Learning

Data augmentations have been applied to train Self-Supervised models bridging the gap between Supervised learning and Unsupervised learning in Computer Vision (CV) [34, 29, 95]. Self-supervised learning regroups methods that can be trained using large amounts of unlabeled data using a pretext task. However, one of the key challenges is the proper design of the pretext tasks [38].

**Leveraging unlabeled sources** One limiting factor of Supervised Learning is requiring a large-scale, manually annotated dataset to train complex models. This is challenging for

domains such as medicine and healthcare and some industrial applications, for which manual labeling can require both equipment and expertise, leaving a large quantity of data unlabeled [38]. The need for manual labeling leaves a large amount of data unused [38]. For example, wearable devices give access to large amounts of unlabeled data [161], which can be beneficial for healthcare settings [160]. Two advantages of such large-scale data collection via wearable devices are that they can provide much more temporal granularity than hospital visits and collect data on a large population.

**Learning from the data directly** A key aspect of self-supervised learning is that the methods leverage information that is readily accessible from the data rather than require manual labeling. Such labels extracted from the data then help design pretext tasks, where the assumption is that information learned through these pretext tasks can be applied to downstream tasks. Such pretext tasks often rely on transformations that are automatable and applicable to new data points without requiring human labeling. For instance, for visual information, one can train the model by solving jigsaw puzzles [141], predicting image rotations [77], colorization [199] and denoising. For audio-visual content, one can train the model to determine whether pairs of audio and visual information come from the same video [3, 4]. For text data, such a task can include masking part of the sentence, training the model to predict the missing content, or predicting the following sentence [53].

### 2.3.3 Foundation models

Over the past years, the field of artificial intelligence has undergone a paradigm shift, as models trained on large-scale datasets using self-supervision have shown great adaptability across various downstream tasks; such models are often referred to as Foundation Models [15]. Foundation models open opportunities in critical areas of society, such as healthcare, law, and education. However, these opportunities come with additional challenges given the sensitivity of such fields, such as interpretability, privacy, and security [15]. These models are often trained on large-scale datasets that require significant computational resources. Once trained, their representation is assumed to be transferrable to other potentially data-scarce tasks. Zhou et al. [203] provide a survey on pre-trained foundation models focusing on Natural Language Processing, Computer Vision, and Graph Learning.

## 2.4 Model comparison

Once we trained models, the question of their performance remains. In the supervised learning regime, one can evaluate the model directly based on the task on which it was trained. For example, if one trains a model to recognize objects in pictures, one can evaluate the model based on its classification accuracy. Nevertheless, more generally, self-supervised foundation models are trained on pretext tasks. While the performance on the pretext tasks might be interesting, what matters most is their performance on what is commonly referred to as downstream tasks, namely, a set of other supervised tasks.

### 2.4.1 Downstream task performance

Self-supervised and Foundation models are often assessed based on the transferability of representations to other tasks and domains. Let the pre-trained model be  $f$ , which can obtain representations  $r$  for an input  $x$  as  $f(x) = r$ . Given this process, one can extract the representations for all the dataset elements. That is, obtaining  $r_i = f(x_i) \forall i \in \{1 \cdots N\}$  where  $N$  is the number of entries in the dataset and  $x_i$  is the  $i$ -th input in the dataset. Then we can use  $\{r_i\}_{i \in \{1, \dots, N\}}$  as input of a downstream classifier  $g$ . Standard methods for the evaluation of self-supervised models are: linear evaluation [34, 30, 185], feature fine-tuning [30] and nearest neighborhood [185, 30].

**Linear evaluation** usually consists of extracting the different representations of the dataset; then, one can use these representations to train a linear model such as Logistic Regression or a linear Support Vector Machine (SVM). One can also generate an augmented version of the input, and use these augmentations as part of the linear model training set [30]. This linear evaluation protocol helps determine whether the representations learned by the self-supervised model can be used to linearly separate classes in the feature space.

**Feature finetuning** usually consists of initializing the model with pre-trained weights obtained through self-supervision and jointly training the classifier and feature extractor. The

assumption is that the pre-trained weight already contains relevant information but could be further improved by learning what is more appropriate for the task.

**Nearest Neighborhood** commonly follows the same steps as Linear Evaluation; the main difference is that the classifier is a weighted K-Nearest Neighbors (K-NN)[185, 30]. As stated by Caron et al. [30], this evaluation protocol requires minimal hyperparameter tuning, namely the number of neighbors, and usually no data augmentation. It can be operated via a single pass over the downstream dataset, simplifying the evaluation protocol.

## 2.4.2 Comparing Representations

In addition to comparing models in terms of their performance on downstream tasks, models can be compared based on their representational similarity [114]. Neural Network representations are often compared based on the activations of a fixed set of inputs and different layers. That is, given two feature extractors  $f_1$  and  $f_2$ , and a set of inputs  $\{x_i\}_{i \in \{1 \dots N\}}$ , one can extract the activation patterns of the inputs for both models. Thus, we can obtain  $R_1 = [f_1(x_1) \cdots f_1(x_i) \cdots f_1(x_N)]$  and  $R_2 = [f_2(x_1) \cdots f_2(x_i) \cdots f_2(x_N)]$ . Often, these representations are not directly comparable; thus, we need mappings that assign a similarity score between  $R_1$  and  $R_2$ .

**Measures based on Canonical Correlation Analysis (CCA) [98]** CCA is a Multivariate Statistics technique that identifies and measures the relationships between two sets of variables or finds patterns representing the variation in the data. It constructs pairs of linear combinations (called canonical variables) from the original variables that are orthogonal within each set and are maximally correlated between sets. For Neural Networks, it compares representations by finding the linear relationships between two sets of activations. Raghu et al. [147] proposed Singular Value CCA (SVCCA) as they considered that representation could be noisy and that CCA would benefit from removing the noise before performing the CCA.

**Representational Similarity Matrices** Centered Kernel Alignment (CKA) [116] can be used to compare the similarity between neural network representations, particularly between

different layers or models. Unlike methods such as CCA, which are based on linear projections, CKA uses kernel methods to capture more complex, nonlinear relationships between the representations. CKA is invariant to orthogonal transformations and isotropic scaling, allowing comparison of different models trained under different initial conditions.

## 2.5 Privacy concerns

Awareness of the privacy risk associated with AI systems has increased over the past decade. Two areas of concern are the capacity of memorization [5, 27, 68] of potentially sensitive information by models [67, 163] and the challenge of removing specific learned data when necessary [154, 119, 66, 70]. These issues can lead to vulnerabilities, allowing adversaries to extract or infer sensitive information from the model.

### 2.5.1 Training set memorization

Training set memorization occurs when machine learning models retain details from training data instead of learning more general patterns. While models are expected to generalize, complex models can memorize exact samples. This poses a privacy risk if the training data includes sensitive or private information, such as personally identifiable information (PII), medical records, or other confidential data.

For instance, large language models trained on text from the Internet can generate examples from their training set, which could include private conversations, unreleased documents, or potentially harmful content. This becomes problematic when sensitive data is leaked or exposed to users querying the model, as it is often difficult to trace which specific inputs contributed to the memorization.

**Model Inversion Attacks** [72, 71, 138, 177] Model inversion attacks aim to reconstruct sensitive input data from a machine learning model’s outputs or internal representations. An adversary exploits access to a trained model, usually by querying it or analyzing its parameters, to infer details about the data on which it was trained. For example, given a model trained on facial images, a successful model inversion attack could recover an approximate version of a

specific individual’s face. This is concerning for models trained on private or sensitive data, such as medical records or personal pictures, because the attacker could extract information that should remain confidential. Model inversion attacks emphasize the need to carefully consider privacy-preserving mechanisms to reduce such risks, such as differential privacy [60, 1].

For example, the study by Fredrikson et al. [72] showed that genetic markers of a patient could be determined based on demographic information about the patient and a model trained for personalized warfarin dosing. Their work highlighted that such machine learning models could carry privacy risks. They showed that with an appropriate privacy budget, they could mitigate the effect of their attacks through Differential Privacy (DP) [60].

**Membership Inference Attacks [156, 26]** Membership inference attacks are a type of privacy attack in which an adversary attempts to determine whether a particular data point was part of the training dataset for a machine learning model. This attack can exploit the fact that models often behave differently on data they have seen during training compared to new, unseen data. An attacker can infer if a given sample was likely included in the training set by observing the model’s outputs, such as confidence scores, prediction, or loss. Membership inference is particularly problematic in scenarios involving sensitive data, like healthcare, finance, or social networks, where knowledge of someone’s inclusion in a dataset could expose them to harm or discrimination. Techniques such as differential privacy, careful regularization, or limiting overfitting in models are often used to reduce the likelihood of leaking information about individual data points. Adversaries can query the model and use its responses to infer the likelihood that the model was trained using a specific individual’s data.

### 2.5.2 Learning to unlearn

Machine Unlearning refers to the process of removing specific knowledge or data from a trained machine learning model [19]. Removing such information can be necessary for complying with data deletion requests, by data regulations such as the General Data Protection Regulation (GDPR) [63], or deleting incorrect or harmful content learned during training. Achieving unlearning is challenging, especially with deep neural networks. It requires removing the influence of specific data or concepts without retraining the entire model from scratch, which can be costly and time-consuming.

**Forgetting concepts** Through unlearning, one could aim to make the models 'unlearn' or forget specific knowledge, concepts, or behaviors once they have been trained. For example, models trained on copyrighted, inappropriate, biased, or outdated information need to remove these sources' information from their parameters. However, forgetting in machine learning is challenging because removing certain concepts or data points can be difficult [65], as altering the model's weights affects its overall performance.

# Chapter 3

## Comparing representations: Feature Impact Balance (FIB) score and Min-Max Relative Change Quadrant (MMRCQ) plot

Given data points, such as images of different objects, we want to compare them to determine if they belong to the same class. To do so, we often require new ways of representing points, allowing us to extract the most relevant information for the task. Once we have these new representations, we need a comparison method. This chapter proposes two tools: the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) plot. These tools provide deeper insights into the distribution of errors across features and how the range of values changes over time. I designed and implemented both the FIB [20] score and the MMRCQ plot, implemented the evaluation pipeline, adapted the codes, ran the experiments, conducted the analyses, and wrote the manuscript.

### 3.1 Introduction

Data representations are critical to model performance in machine learning and data analysis. Whether predicting object properties from images, analyzing medical records, or processing

audio recordings, the representation of these data points influences our ability to predict their properties correctly.

Traditional error metrics, such as Mean Squared Error (MSE) or Mean Absolute Error (MAE), measure how predicted values differ from actual values. Nevertheless, while traditional metrics provide information about the magnitude of the errors, they do not account for the positions of the errors. Knowing whether all the positions in a vector contributed to the errors can matter in domains such as healthcare, where understanding which features contribute most to prediction errors is essential for model trustworthiness and interpretability.

We introduce two novel tools to complement the traditional error metrics: the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) plot. The FIB score quantifies the error distribution across positions in a vector, providing a clearer understanding of which features contribute most to the error. Understanding the distribution of errors allows for a more detailed error analysis than standard metrics like MSE, which only report overall error magnitude without highlighting individual feature contributions. Meanwhile, the MMRCQ plot visualizes changes in extreme values (*e.g.*, minimum and maximum) across settings, such as changes over time or between datasets. Visualizing changes in settings could provide insights into tracking distribution shifts in finance, healthcare, or climate science domains. For instance, in finance, tracking the minimum and maximum stock prices over time using the MMRCQ plot could help visualize a stock’s volatility. A point in the upper-left quadrant of the MMRCQ plot would indicate that the stock’s minimum price has decreased while its maximum price has increased, suggesting increased volatility.

### 3.1.1 Defining data representations

Table 3.1: Example of Tabular Data Representation

| Hospital ID | Patient ID | Age | Gender | Cognitive Score | Diagnosis                 |
|-------------|------------|-----|--------|-----------------|---------------------------|
| 1           | 101        | 65  | Male   | 25              | Healthy                   |
| 1           | 102        | 70  | Female | 17              | Mild Cognitive Impairment |
| 2           | 201        | 68  | Male   | 8               | Cognitively Impaired      |

A representation refers to how one provides data to a computer or machine learning model. For different tasks and data modalities, the appropriate representation can vary significantly. For instance, in Computer Vision (CV), pixel values form image representations, while in Natural

Language Processing (NLP), embeddings or tokens can represent words. Let us illustrate a tabular format with a fictional medical record dataset. In such a format, each row represents a unique observation (*e.g.*, a patient), and each column represents an attribute (*e.g.*, Age, diagnosis) (Table 3.1). For instance, the **Age** column contains the Age of each patient; the **Hospital ID** column contains the identifier of the hospital that admitted the patient, and so on.

### 3.1.2 Why representations matter?

The structure of a representation affects a model’s ability to learn from data. For instance, models trained with unnecessary information, such as the hospital or patient identifiers, could learn spurious correlations. Such models might focus on irrelevant rather than of patient-related information. Meanwhile, representations that capture relationships, such as time series data in healthcare, improve model performance by leveraging relevant patterns to learn from. One can draw a parallel between machine learning representations and data structures, where selecting the appropriate structure (*e.g.*, tree versus graph [49]) impacts algorithm efficiency.

As such, depending on the task, certain representations are more suitable than others. For instance, in table 3.1, if we keep the Hospital ID or the Patient ID, we might end up with a model that uses the Hospital ID to make its predictions.

We further illustrate the impact of representation by considering the problem of distinguishing between two concentric circles, adapted from Goodfellow et al., [84]. This example shows how different representations can drastically change the ability to separate data classes.

### 3.1.3 Example of representation transformation: Distinguishing concentric circles

Consider the problem of distinguishing between two concentric circles, which represent two distinct classes of data points (Figure 3.1). Drawing a linear boundary between these two classes is difficult in Cartesian coordinates, where each point is defined by its  $x$  and  $y$  values. However, the class separation becomes more straightforward if we switch to polar coordinates, where each point is defined by its distance from the origin ( $r$ ) and angle ( $\theta$ ). Moreover, combining the

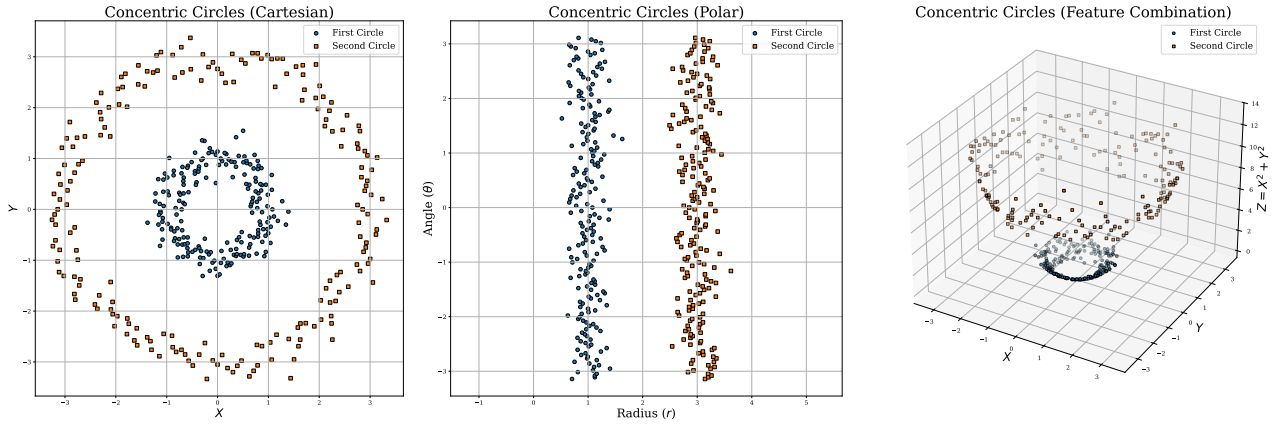


Figure 3.1: Two concentric circles with different radii using different representations: Cartesian Coordinates (left), Polar Coordinates (center), Feature Combination (right). Each of the representation can be derived from the Cartesian coordinates, nevertheless, using the Polar Coordinates (center) or the Feature Combination (right), one could draw a linear separation between the set of points of both circles.

Cartesian coordinates to create a new feature, such as  $z = x^2 + y^2$ , creates a new representation space where we can linearly separate the circles. Such differences highlight how the choice of representation can directly affect a model's ability to distinguish between different classes, thereby influencing overall performance.

The main difference comes from the type of coordinate system used to represent the data points. In mathematics, Cartesian coordinates and polar coordinates are two systems used to specify a point's position in a plane.

**Cartesian Coordinates** : A point  $P$  in a two-dimensional plane is represented by its horizontal and vertical distances from the origin, denoted as  $(x, y)$ .

**Polar Coordinates** : A point  $P$  can also be represented by its distance  $r$  from the origin and the angle  $\theta$  it makes with the positive  $x$ -axis. The equations give the conversion between polar and Cartesian coordinates:

$$x = r \cos(\theta) \quad (3.1)$$

$$y = r \sin(\theta) \quad (3.2)$$

In this representation, the data points are represented in a new feature space, where the distance from the origin and the angle are the new features. We could now draw a line on the distance

$r$  axis in this space to separate the two classes (Figure 3.1).

**Feature Combination** : In this version, we generate a new dimension  $z$  based on the combination of the Cartesian coordinates by adding the square of  $x$  and the square of  $y$ . This is expressed as:

$$z = x^2 + y^2 \tag{3.3}$$

By creating this new feature, we transform the data points into a new feature space where a linear separator can divide the two classes (Figure 3.1). This example highlights the importance of choosing representations when working with complex data. Different representations impact the model’s learning ability and affect feature error distribution. We introduce the Feature Impact Balance (FIB) score to understand better and quantify this distribution.

## 3.2 Assessing balance of contributions: Feature Impact Balance (FIB)

This section covers the Feature Impact Balance (FIB) score, designed to determine whether the error between two vectors is concentrated on a single position or distributed across multiple positions. First, we review standard error metrics used for vector comparisons and their limitations. We then introduce the FIB score and its advantages.

### 3.2.1 Error Metrics

Error metrics provide a quantitative way to compare different models based on their performance (*i.e.*, an error means the deviation of the predicted value from the true value of a quantity of interest). We can use these metrics to objectively evaluate how well a model performs its functions, such as regression tasks. Standard error metrics are Mean Squared Error (MSE), Mean Absolute Error (MAE), which can be used to measure the average of the squares of the differences between predicted and actual values. These metrics are easy to compute and interpret, so they have been widely used in practice. Nonetheless, they do not provide information about the distribution of errors or the specific features contributing to them. Three standard

components determine the characteristics of performance metrics and are used to design them: 1) method for measuring point distance, 2) normalization method, and 3) method for aggregating point distances over a dataset [17]. Determining the point distance between actual and predicted values, normalizing them, and then aggregating them over the entire set of data is necessary as a sequential process. All performance metrics explicitly contain components that determine the distance between points and aggregate them, while the normalization component is optional [17].

While these error metrics, such as MSE, are widely adopted partially due to their simplicity and ease of computation, they have limitations. Such metrics do not inform about the distribution of errors across different positions, providing limited insight into the contribution of individual elements to the overall error. The following section dives into this issue, illustrating cases where commonly used metrics like MSE fall short in distinguishing between errors distributed unevenly across vectors.

### 3.2.2 Lack of insight into the error distribution

The Mean Squared Error (MSE) is a commonly used error metric in regression tasks. It calculates the average squared difference between predicted and true values, summarizing the overall error magnitude. However, we observed that MSE does not distinguish between different patterns of errors or reveal how the discrepancies are distributed across individual elements in the vectors. To illustrate, consider comparing a reference vector  $\mathbf{x}$  to a set of vectors  $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$ . Let us take  $\mathbf{x} = [1.0, 4.0, 1.5, 4.0]^T$  and  $\mathbf{y}_1 = [1.0, 4.0, 1.5, 8.24 \dots]^T$ ,  $\mathbf{y}_2 = [1.0, 0.0, 0.5, 3.0]^T$ , and  $\mathbf{y}_3 = [3.0, 2.0, 3.5, 1.55 \dots]^T$ . When we compute the MSE between  $\mathbf{x}$  and any of the  $\mathbf{y}_i$ , the metric treats all squared differences equally, regardless of how they are distributed across the vector position. Leading to nearly equal MSE values between all pairs  $(\mathbf{x}, \mathbf{y}_i)_{i \in \{1,2,3\}}$ . This limitation is illustrated in Figure 3.2, where the MSE values are equal, even though the error distribution varies significantly.

For instance, for the pair  $(\mathbf{x}, \mathbf{y}_1)$ , the entire difference between the vectors comes from the fourth element, indicating that a single position accounts for all the errors. In contrast, each element contributes to the difference for  $\mathbf{y}_2$ ; that is, the error is spread more evenly across all positions. Despite these distinct patterns, the MSE does not indicate the distribution of differences,

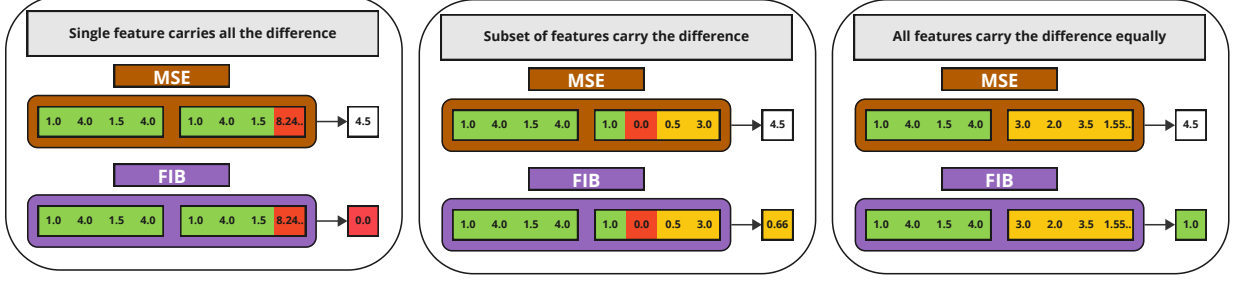


Figure 3.2: Comparison between Mean Squared Error (MSE) and Feature Impact Balance (FIB), illustrating how differences are distributed across vector positions. Given a reference vector and three other vectors, we compute the MSE between the reference vector and each comparison vector. While the MSE remains the same for all three comparisons, the distribution of differences across vector positions varies. These differences are captured by the FIB, which are 0.0, 0.66, and 1.0 from left to right, respectively. The FIB values indicate that: (left) a single position carries all the difference, (center) a subset of positions carries the differences, and (right) the differences are distributed equally across all positions.

treating both cases as equivalent in terms of error magnitude. This lack of granularity can be problematic when understanding the source and spread of error is essential. Meanwhile, if we now consider the difference between  $\mathbf{x}$  and  $\mathbf{y}_3$ , all four elements of the vector contributed to the difference, and therefore, the difference spreads across the entire vectors.

Now, to draw parallels with real-world use cases, consider that each vector represents an aggregated representation of the population of different hospitals. If we were to use the MSE to compare hospitals, they could be considered similar, even though different features and positions in the vectors carry the differences.

To address this limitation, we introduce the Feature Impact Balance (FIB) score, a tool designed to identify whether errors are concentrated on a few key features or distributed more evenly across all features. This granularity allows for a deeper understanding of model performance beyond what traditional error metrics like MSE can offer.

### 3.2.3 Addressing the lack of insight into the error distribution: Feature Impact Balance (FIB) score.

We introduce the Feature Impact Balance (FIB) score to measure error variability. First, we define the different steps to compute the FIB score. The FIB score measures how evenly errors

are spread across positions, ranging from 0 (where one feature accounts for all the errors) to 1 (where errors are equally distributed). This design ensures that FIB provides a normalized measure of error distribution independent of the magnitude of the error. By squaring the differences in feature contributions and summing them, FIB penalizes cases where error distributions differ from a uniform distribution, making it sensitive for uncovering imbalances.

## The Feature Impact Balance (FIB) score

The FIB score is defined as:

$$FIB(\mathbf{c}) = 1 - \frac{M}{M-1} \sum_{k=1}^M (c_k - \frac{1}{M})^2 \in [0, 1] \quad (3.4)$$

With  $\mathbf{c}$  the contribution vector (see Section 3.2.3) and  $M$  the number of features.

## The contribution vector $\mathbf{c}$

The contribution vector  $\mathbf{c}$  is a vector that contains the contributions of each position to the error. This vector has the following properties:  $\sum_{k=1}^M c_k = 1$  and  $\forall k \in \{1, \dots, M\}, c_k \in [0, 1]$ . Namely, we could consider any function  $g(\mathbf{x}, \mathbf{y}) = \mathbf{c}$  that takes two vectors  $\mathbf{x}$  and  $\mathbf{y}$  and returns a vector  $\mathbf{c}$  that satisfies the properties mentioned above. The construction of such vector depends on the use case, a simple choice for the contribution vector is the one that is used in the following experiments,  $g(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}-\mathbf{y}|}{\|\mathbf{x}-\mathbf{y}\|_1}$ . Nonetheless, this choice can drastically change what is being observed. For instance, consider that  $\mathbf{x}, \mathbf{y}$  are RGB images. One could build a contribution vector that only considers the relative importance of each color channel, leading to a vector of length 3. Once we obtain the contribution vector, we can compute the balance error.

## The Balance Error (BE)

To quantify the imbalance of the contributions, we define the Balance Error (BE). When each feature contributes equally, *i.e.* contributes to  $\frac{1}{M}$  of the sum of the errors, the BE equals 0. For instance, with four elements, if each element contributes to 25% of the sum of the errors,

the BE equals 0. Otherwise, it quantifies how far the contributions are from the expected contributions. One could use the MSE to compute the BE. Once we have the BE, we can compute the Feature Impact Imbalance (FII).

## The Feature Impact Imbalance (FII)

The Feature Impact Imbalance (FII) as the Balance Error (BE) between the contribution vector  $\mathbf{c}$  and a balance vector  $\mathbf{b} = \frac{\mathbf{1}_M}{M} \in \mathbb{R}^M$ . As such we define the FII as:

$$FII(\mathbf{c}) = BE(\mathbf{c}, \frac{\mathbf{1}_M}{M}) \quad (3.5)$$

This provides a way to quantify the error between the observed contributions and uniform contributions. Finally, we take the mean of these balance errors to obtain a single value to quantify the imbalance. Nonetheless, the scale of FII depends on the number of features  $M$ . To address this, we normalize the FII and give its maximal value.

**Maximal values** We demonstrate that using the MSE as balance error leads to FII that can reach a maximum via Theorem 3.2.2. We provide its value using Theorem 3.2.3.

Let  $(c_k)_{k \in \{1, \dots, M\}}$  be the values of  $FI$  such that  $FI(\mathbf{x}, \mathbf{y}) = [c_1, \dots, c_M]^T$ . By construction  $\forall k \in \{1, \dots, M\}, 0 \leq c_k \leq 1$  and  $\sum_{k=1}^M c_k = 1$ . The right-hand side of the equation 3.5 when using the MAE can be rewritten as a function of  $M$  variables noted  $\varphi(c_1, \dots, c_M) : \mathbb{R}^M \rightarrow \mathbb{R}$

$$\varphi(c_1, \dots, c_M) = \frac{1}{M} \sum_{k=1}^M |c_k - \frac{1}{M}| \quad (3.6)$$

Similarly, when using the MSE, we now consider  $\psi(c_1, \dots, c_M) : \mathbb{R}^M \rightarrow \mathbb{R}$

$$\psi(c_1, \dots, c_M) = \frac{1}{M} \sum_{k=1}^M (c_k - \frac{1}{M})^2 \quad (3.7)$$

We first give the following definition to compute the maximum of  $\varphi$  and  $\psi$ .

**Definition 3.2.1.** [204] A  $M$ -simplex in  $\mathbb{R}^M$ ,  $M \geq 1$ , is the convex hull of  $M + 1$  affinely independent points.

An  $M$ -simplex is the intersection of all convex sets containing these points; it is also the smallest convex set containing these points. It finally follows from a theorem due to Carathéodory [204] that the  $M$ -simplex  $S$  defined by the points  $a_1, \dots, a_{M+1}$  is characterized as follows: a point  $a$  belongs to  $S$  if and only if there exists  $\lambda_i, i = 1, \dots, M+1, \lambda_i \in [0, 1], i = 1, \dots, M+1, \sum_{i=1}^{M+1} \lambda_i = 1$ , such that

$$a = \sum_{i=1}^{M+1} \lambda_i a_i.$$

This yields that an  $M$ -simplex is closed and bounded, and thus compact, in  $\mathbb{R}^M$ ; it is also, by definition, convex.

**Theorem 3.2.2.** *Let  $f : S \rightarrow \mathbb{R}$  be continuous and convex, where  $S \subset \mathbb{R}^M$  is the  $M$ -simplex defined by the points  $a_1, \dots, a_{M+1}, M \geq 1$ . Then  $f$  reaches its maximum at one of the  $a_i$ 's.*

*Proof.* First note that  $f$  is continuous on the nonempty compact set  $S$ . It thus reaches its maximum.

Let then  $a \in S$ . We can thus write

$$a = \sum_{i=1}^{M+1} \lambda_i a_i,$$

where  $\lambda_i \in [0, 1], i = 1, \dots, M+1, \sum_{i=1}^{M+1} \lambda_i = 1$ . Since  $f$  is convex, we have

$$\begin{aligned} f(a) &= f\left(\sum_{i=1}^{M+1} \lambda_i a_i\right) \leq \sum_{i=1}^{M+1} \lambda_i f(a_i) \\ &\leq \max(f(a_1), \dots, f(a_{M+1})), \end{aligned}$$

which proves the theorem. □

**Theorem 3.2.3.** *Let  $A = \{(c_1, \dots, c_M) \in \mathbb{R}^M, c_i \in [0, 1], i = 1, \dots, M, \sum_{i=1}^M c_i = 1\}$ . We set  $\varphi(c_1, \dots, c_M) = \sum_{i=1}^M |c_i - \frac{1}{M}|$  and  $\psi(c_1, \dots, c_M) = \sum_{i=1}^M (c_i - \frac{1}{M})^2$ . Then,  $\max_A \varphi = 2\frac{M-1}{M}$  and  $\max_A \psi = \frac{M-1}{M}$ .*

*Proof.* For  $M = 1$  we have  $c_1 = 1$ , so that both  $\varphi$  and  $\psi$  vanish and the assertion of the theorem holds true. Let us now assume that  $M \geq 2$ . Note that  $\varphi$  and  $\psi$  are continuous and convex (as sums of continuous and convex functions). Then note that we wish to maximize  $\varphi$  and  $\psi$

on the  $(M - 1)$ -simplex defined by the points (in  $\mathbb{R}^{M+1}$ )  $(1, 0, \dots, 0), \dots, (0, 0, \dots, 0, 1)$  in the affine hyperplane  $\sum_{i=1}^M c_i = 1$ . The result then follows from Theorem 3.2.2.  $\square$

When using the MSE as a Balance Error, we know from Theorem 3.2.2 that:

$$0 \leq FII(\mathbf{c}) \leq \frac{M - 1}{M^2} \quad (3.8)$$

where the value 0 means that the feature impact is balanced, as it is achieved in the case where  $\forall k \in 1, \dots, M, c_k = \frac{1}{M}$ .

## Normalized Feature Impact Imbalance (NFII)

Using the maximal values of the FII, we can now normalize the FII scores and obtain values in  $[0, 1]$ . For a more straightforward interpretation of the FII values, we define the Normalized Feature Impact Imbalance (NFII) as:

$$NFII(\mathbf{c}) = \frac{M^2}{M - 1} FII(\mathbf{c}) \quad (3.9)$$

From equations 3.8 and 3.9 it comes that:

$$0 \leq \frac{M^2}{M - 1} FII(\mathbf{c}) \leq 1 \quad (3.10)$$

Finally, we obtain the Feature Impact Balance (FIB). For conciseness, we omit the mention that it is normalized and define it as:

$$FIB(\mathbf{c}) = 1 - NFII(\mathbf{c}) \in [0, 1] \quad (3.11)$$

A value of 1 indicates that each feature contributed equally to the errors, and 0 indicates that a single feature is responsible for the error.

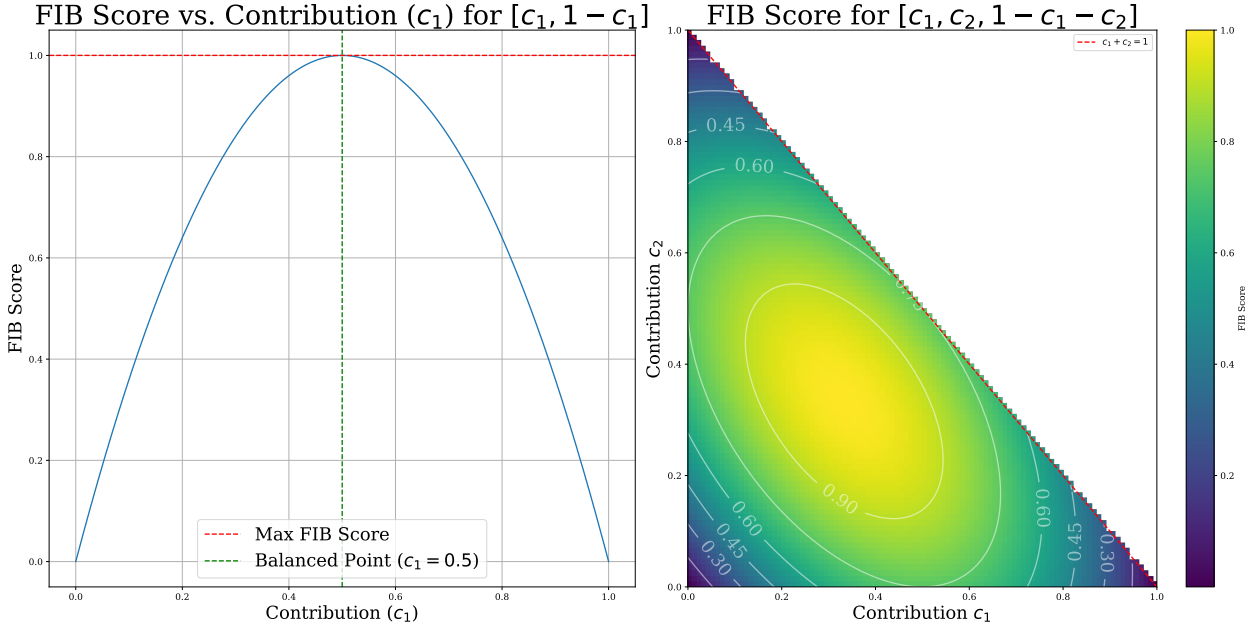


Figure 3.3: Visualization of the Feature Impact Balance (FIB) score as a function of feature contributions. The left plot shows the FIB score as a function of the contribution  $c_1$  for two features,  $c_1$  and  $1 - c_1$ , highlighting the maximum FIB score and the balanced point where  $c_1 = 0.5$ . The right plot extends this to three features,  $c_1$ ,  $c_2$ , and  $1 - c_1 - c_2$ , showing how FIB scores vary as contributions are distributed across multiple features. Higher FIB scores correspond to more balanced feature contributions.

### 3.2.4 Limitations

One of the challenges of the FIB score is the choice of the contribution vector  $\mathbf{c}$ . If the number of features is high, the FIB reaches its maximal value often. Furthermore, since it does not account for the magnitude of the contributions, it should be combined with the MAE or MSE.

### 3.2.5 Perspectives

We can consider different contribution vectors depending on which information we want to focus on. For instance, the FIB could be applied directly to the predictions of regression models to understand whether the model makes errors on a specific data point. Or, to obtain a more fine-grained understanding of the errors, we could group data points based on their labels. We could determine whether the errors are systematic within a specific group, such as low or high values, or based on properties of interest and specific attributes in image recognition.

### 3.3 Visualizing relative changes: Min-Max Relative Change Quadrant (MMRCQ) Plot

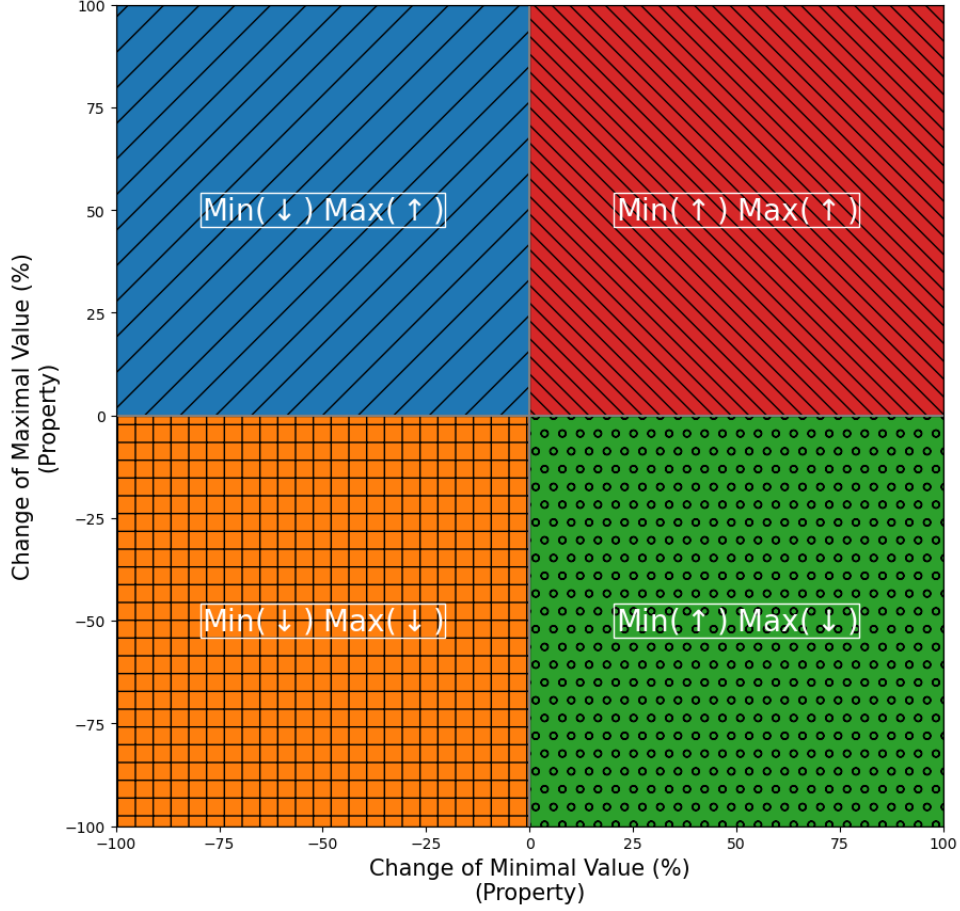


Figure 3.4: Min-Max Relative Change Quadrant (MMRCQ): Based on the localization of a point in the MMRCQ plot we can infer changes in the minimal and maximal values of a property with respect to a baseline. Points located in the upper quadrants indicate an increase in the maximal value while points located in the lower quadrants indicate a decrease in the maximal value. Points located in the left quadrants indicate lower minimal values, and points located in the right quadrants indicate higher minimal values.

In this section, we introduce a novel visualization tool called the Min-Max Relative Change Quadrant (MMRCQ) plot. We designed this tool to capture and compare changes in the range of values to a baseline, providing a visual representation of the evolution of the minimum and maximum values. Unlike plots that focus on mean tendencies, the MMRCQ plot offers insights into the variations of extreme values in data sets.

Consider a property with an initial set of values  $s = \{1, 2, 3, 4\}$ , where the minimum is 1 and

the maximum is 4. Now, suppose a new set of observations  $s' = \{0, 1, 1, 5\}$ , where the minimum has decreased to 0 and the maximum has increased to 5.

The relative change in the minimal value is calculated as:  $\frac{0-1}{1} = -1$  This indicates a *100% decrease* in the minimal value. Similarly, for the maximum value:  $\frac{5-4}{4} = 0.25$  this represents a *25% increase* in the maximum value.

These changes can be plotted as the point  $(-100, 25)$  in the MMRCQ plot, placing it in the upper-left quadrant (shaded with blue diagonal (Figure 3.4)). This quadrant indicates that while the minimal value has decreased by 100%, the maximal value has increased by 25%.

The MMRCQ plot is valuable for visualizing the evolution of the minimum and maximum values compared to a baseline. Each quadrant reveals distinct patterns of change, offering insight into the overall range dynamics:

- **Upper-left quadrant:** Minimum value decreased, maximum value increased — indicating a widening range.
- **Lower-right quadrant:** Minimum value increased, maximum value decreased — indicating a narrowing range.
- **Upper-right quadrant:** Both minimum and maximum values increased — the range has shifted upwards.
- **Lower-left quadrant:** Both minimum and maximum values decreased — the range has shifted downwards.

The MMRCQ plot (Figure 3.4) provides a clear and intuitive visual representation of how the range of a property evolves over time or under different conditions.

### 3.3.1 Limitations

One limitation of the MMRCQ plot is that it may not be as effective for comparing properties with different scales. For example, suppose we have a property with values in  $[0, 1000]$ , and another with values in  $[1, 10]$ . In that case, the changes in the range of the first property will be

much smaller than the changes in the range of the second property, even if the relative changes are the same.

### 3.3.2 Perspectives

One can use the MMRCQ plot to visually inspect relative values changes in domains, such as finance, weather, time series, or healthcare. For instance, one could monitor fluctuations of maximal and minimal values of a treatment effect over time. In sports, the MMRCQ plot could be used to compare the performance of athletes over time or to compare the performance of different athletes.

In the following chapter, we consider the effects of data representations on different classes of data points. We will see one application of the FIB score and MMRCQ to compare the effects of common data augmentations on image data.

# Chapter 4

## Data Augmentation affects classes differently

XC designed and implemented the evaluation pipeline, adapted the codes, ran the experiments, conducted the analyses, and wrote the manuscript.

### 4.1 Introduction

Deep learning methods have remarkably succeeded in Computer Vision (CV) tasks. However, one of the significant challenges in training these models is the availability of high-quality annotated data. Such datasets can be tedious to obtain, notably in domains that require specialized equipment or domain knowledge, such as medical tasks.

Data augmentation has become an essential technique for improving the generalization ability of deep learning models, particularly in fields where acquiring labeled datasets is both time-consuming and expensive, such as medical image processing, image segmentation, and image-guided therapy [149]. In tasks like image segmentation and diagnosis, where high-quality annotated data is often limited, augmentation techniques allow models to learn from smaller datasets by artificially increasing the size of the training set through transformations such as flipping, cropping, and rotating images. Furthermore, augmentation can help address class imbalance by generating examples from classes with few samples.

Augmentation methods are commonly applied uniformly irrespective of class, and their impact can differ depending on each class. For instance, some augmentations can alter the image to the extent of changing its class, such as flipping the number '6' into a '9': we will refer to this as a label-breaking transformation, as opposed to a label-preserving transformation that does not alter the label. This variation in impact highlights the need for a more nuanced approach to data augmentation, particularly in tasks where class characteristics and representation vary significantly.

Given the diverse effects that data augmentation can have across classes, a comprehensive evaluation is necessary to determine its impact. In this chapter, we leverage the two tools introduced in Chapter 3: the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) plot. These tools allow us to quantify and visualize how augmentations alter the relationships between different classes, providing valuable insights into how augmentation affects each class.

## 4.2 Background

Data augmentation is a technique used to improve the performance of deep learning models. It can introduce more variability in the training data and increase its size without requiring additional data collection. By augmenting the variability of the training data, the model can learn to generalize better to new, unseen data.

However, the impact of these augmentations can vary significantly depending on the specific characteristics of each class. This section discusses two main families of data augmentations for Computer Vision (CV) [157]: *geometric image manipulations* and *photometric image manipulations* and the critical distinction between *label-preserving* and *label-breaking* transformations.

### 4.2.1 Adding variability to the training data using Data Augmentation

Data augmentation can take different forms. In Computer Vision (CV), two typical families of data augmentation methods are *geometric image manipulations* and *photometric image*

*manipulations.*

## Modifying structural information using *Geometric transformations*

Geometric augmentations involve manipulating the spatial structure of an image. These transformations effectively reduce positional biases in the training data, such as the object-centric bias in ImageNet [146]. Namely, the object of interest is often alone in the image and generally in the center of the image. The most commonly used geometric augmentations include: *flipping*, *rotating*, and *cropping* transformations [118, 56, 167, 157, 9, 34].

- **Flipping:** creates a mirror image of the original image horizontally, vertically, or both. Horizontal flips are more common, especially in tasks with less critical object orientation, such as animal or landscape recognition. Nonetheless, these augmentations can have adverse effects when applied to classes for which the orientation is crucial, such as text or digits. For instance, by successively flipping horizontally and vertically an image of a 9, we obtain a picture of a 6 (Figure 4.1).
- **Rotating:** rotates an image by a specified angle. Nonetheless, excessive rotation can distort spatial information, leading to misclassification. For instance, by rotating an image of a 9 by 180 degrees, we obtain a picture of a 6 (Figure 4.1).
- **Cropping:** Random cropping, commonly followed by padding to restore image size, helps reduce over-reliance on positional information. Cropping can help the model generalize better, as long as it does not remove critical parts of the object.

These geometric transformations can improve the models' generalization capabilities. Nonetheless, it is important to consider photometric transformations. While geometric transformations modify the spatial characteristics of the image, such as its position or orientation, photometric transformations modify its color and intensity.

## Modifying the visual appearance of the image using *Photometric transformations*

Photometric transformations modify the image’s visual appearance without modifying its spatial structure. These transformations focus on adjusting the color properties of the image and usually include *brightness*, *contrast*, *saturation*, and *hue*.

- **Brightness:** alters the intensity of the light in the image, which can help models generalize better to variations in lighting conditions.
- **Contrast:** contrast adjustments amplify or dull the difference between lighter and darker areas of the image. Contrast alteration can help in tasks where image clarity varies across samples. However, extreme changes in contrast can hide critical features, particularly in fine-grained classification tasks.
- **Saturation:** makes colors more vibrant or shifts the image closer to grayscale. Saturation alteration can be helpful in natural scene datasets, where the richness of color may vary.
- **Hue:** alters the dominant color, which can help train models to be less sensitive to color variations. However, excessive shifts can distort the original colors, making it challenging to recognize objects. For example, some bird species have plumage that appears similar in different lighting conditions, and significant hue shifts can lead to their misclassification.

Although powerful, photometric augmentations, such as strong color shifts, may degrade the model’s ability to differentiate between certain classes. It has been shown that for Convolutional Neural Network (CNN), training using augmented datasets leads to better performance, and geometric augmentation outperforms photometric transformation [167].

## Additional data augmentations: Noise Injection, Random Erasing, and Image Mixing

In addition to Geometric and Photometric augmentations, other augmentations have been proposed: noise injection [137], random erasing [202], or combination of data points [104, 166].

- Noise Injection [137]: This technique involves adding noise to the image. Usually, one draws such noise from a Gaussian distribution.
- Random Erasing: this technique randomly erases a part of the image. Enforcing the model to learn representations less reliant on a specific pattern. Parallel can be drawn to Dropout [162], where the model learns more robust features as neurons are randomly dropped during training.
- Image Mixing: This technique combines two images in a structured manner. For instance, Inoue et al. [104] introduced Sample Pairing, which focuses on averaging pairs of training samples to create new training samples. Additionally, Zhang et al. [198] proposed MixUp, a method based on linear interpolation of training samples and labels to create new training samples.

#### 4.2.2 Data augmentation in the feature space rather than the input space

Rather than applying data augmentations in the input space as commonly done, Devries and Taylor [54] proposed to apply data augmentation in the feature space. A significant observation was that extrapolation within the feature space helped generate new synthetic samples, which are more valuable than interpolated examples. Nonetheless, the authors noted that extrapolated examples were more valuable when class boundaries were complex, while interpolated were more valuable when classes were well separated.

#### 4.2.3 The distinction between label-preserving and label-breaking transformations

A key aspect of data augmentation is whether the transformation application preserves the sample's label. The preservation of the label matters and leads to the distinction between *label-preserving* and *label-breaking* transformations. Label-preserving transformations do not change the sample's label while label-breaking transformations do.

Label-preserving transformations ensure that the image still represents the same class after augmentation, a common assumption in most augmentation techniques. However, some transformations can lead to label-breaking outcomes, where the image no longer belongs to the same class after augmentation. For example, flipping a digit '9' horizontally and vertically can turn it into a '6', introducing noise into the dataset (figure 4.1). Such transformation defeats our purpose in identifying the number 9; this augmentation would make the dataset harder to learn from instead of easier. As such, label-breaking transformations can confuse the model during training, leading to degraded performance. Conversely, horizontally flipping an image of a cat does not change its classification, making it a label-preserving transformation (figure 4.2).

This distinction becomes particularly important in tasks where class labels are sensitive to spatial orientation or visual features. For instance, slight photometric changes in medical images may distort critical diagnostic features, leading to misclassification.

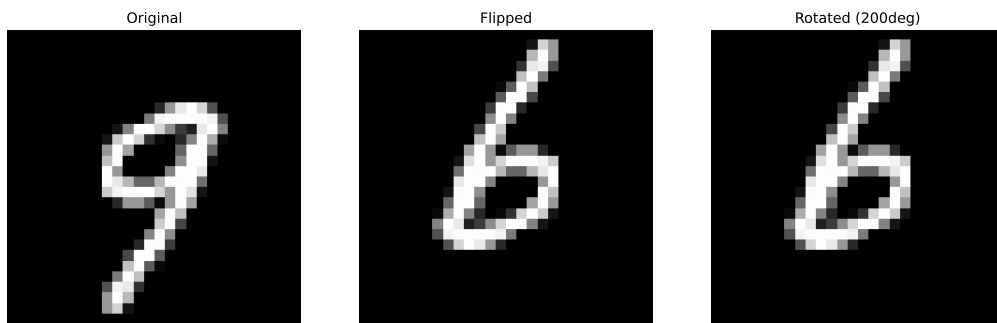


Figure 4.1: Label-breaking transformation: Starting from a 9, if we flip both horizontally and vertically, we obtain 6. This highlights that data augmentations requires careful consideration depending on the data. (Images from MNIST[123])



Figure 4.2: Label-preserving transformation: A cat remains a cat after the different transformations. (Images from Oxford IIIT Pet[143])

More formally, in the context of an image classification task. Let  $\mathbf{x} \in \mathcal{I}$  be an image, with  $\mathcal{I}$

the image space. Let  $\phi : \mathcal{I} \rightarrow \mathcal{T}$  be a transformation, with  $\mathcal{T}$  the transformed image space. Let  $c$  be an oracle that takes an image and returns its label. We say that  $\phi$  is **label-preserving** for  $\mathbf{x}$  if  $c(\mathbf{x}) = c(\phi(\mathbf{x}))$  otherwise it is label-breaking.

To illustrate a label-preserving transformation, consider an image of a cat: horizontally flipping the image does not change the label of the image; it still represents a cat (Figure 4.2).

#### 4.2.4 Adversarial Training

Rotation and translation perturbation in the training set can lead to performance decrease in Convolutional Neural Network (CNN) [61, 145]. Adversarial attacks leveraging color perturbations highlight shortcomings when training deep neural networks. Semantic adversarial examples [97, 13, 99] attempt to perturb the model while semantically representing the original object by introducing variations in the hue and saturation components of images.

#### 4.2.5 Inter-Class Similarity

In classification tasks, one can observe similarities between classes at different levels. For example, from a semantic standpoint, there exists a stronger similarity between "cats" and "dogs" than between "cats" and "airplanes" based on WordNet[69]. While it is common practice to penalize miss-classification irrespective of the similarity between classes, Gare et al. [75] proposes to penalize classification differently based on class similarities. Fujii et al. [74] suggests selectively adjusting the data augmentation strategy based on inter-class distances.

#### 4.2.6 Contextual Bias in Image Classification

In image classification, it is common to have both the object one aims to identify and some background. Research investigating the effect of background in image classification made the standard model misclassify 87.5% of images as the class of an adversarial background [187]. Furthermore, researchers have become more interested in the impact of contextual biases in computer vision in recent years. Singh et al. [159] showed that standard classifiers perform worse when presented with biased classes without their usual context. Luccioni et al. [128] noted

that, in ImageNet-1k [52], some species of the wild animal classes 61.6% of their occurrences were represented alongside humans.

#### 4.2.7 Motivation for Novel Evaluation Metrics

Common deep-learning pipelines apply data augmentations with limited consideration of the impact of the classification on the classes. As such, the impact of data augmentation at the class level within a dataset has yet to be systematically investigated. We, therefore, set out to fill this gap by proposing a method to evaluate the effects of data augmentation methods (positive or negative) by measuring their impact at the input level on different classes and how these changes affect their proximity to one another. To meet our objective, we used the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) visualization method based on changes in minimal and maximal values. The FIB score quantifies the balance of changes across class features, providing insight into whether augmentations affect the majority of a class’s features or only a few. The MMRCQ plot helps understand the extent of change in class relationships due to augmentations by comparing the minimal and maximal feature distances between classes.

### 4.3 Methodology

This section aims to evaluate the impact of data augmentation on class representations using aggregated metrics. We analyze how different data augmentation techniques affect the aggregated representations of classes, which summarize key features of each class. Our analysis utilizes metrics such as Mean Squared Error (MSE), Structural Similarity Index (SSIM), and the Feature Impact Balance (FIB) score to quantify and compare the effects of augmentations.

#### 4.3.1 Class aggregated representations

We compute an aggregated representation for each class by averaging the transformed images in the class. Aggregating class representations helps us capture key features while eliminating noise from individual images. Formally, the aggregated representation for class  $c$  after applying

a data augmentation transformation  $\phi$  is given by:

$$A_c^\phi = \frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} \phi(x) \quad (4.1)$$

Here,  $\mathcal{X}_c$  represents the set of images belonging to class  $c$ , and  $|\mathcal{X}_c|$  denotes the number of images in the class. When  $\phi = \text{id}$ , i.e., no augmentation is applied, we obtain the baseline aggregated representation of the class  $A_c$ . By comparing  $A_c^\phi$  with  $A_c$ , we can evaluate how augmentations change the representation of each class.

### 4.3.2 Computing differences between aggregated representations

We use the Mean Squared Error (MSE) to quantify the difference between aggregated representations, which captures pixel-wise differences between two images. The MSE between two images  $x$  and  $y$  is defined as:

$$\text{MSE}(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (4.2)$$

where  $N$  is the number of pixels, and  $x_i$  and  $y_i$  are the pixel values at position  $i$ . MSE provides a direct measure of how much the application of a data augmentation transformation changes the overall pixel values of a class's aggregated representation.

### 4.3.3 Computing similarities between aggregated representations

In addition to MSE, we use the Structural Similarity Index (SSIM) [179] to measure the structural similarity between images. SSIM is designed to account for perceptual differences between images, focusing on luminance, contrast, and structure. The SSIM is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.3)$$

Where  $\mu_x$  and  $\mu_y$  are the means of images  $x$  and  $y$ ,  $\sigma_x^2$  and  $\sigma_y^2$  are the variances of  $x$  and  $y$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ ,  $C_1$  and  $C_2$  are constants to avoid division by zero.

While MSE captures pixel-wise differences, SSIM provides insight into structural changes caused by augmentations, making it a complementary metric for evaluating image transformations.

#### 4.3.4 Computing the error balance between aggregated representations

To quantify the balance of feature changes across classes, we leverage the Feature Impact Balance (FIB) score [20]. The FIB score measures how balanced the changes in features are across the dataset after applying augmentations. It is defined as:

$$\text{FIB}(c) = 1 - \frac{1}{M-1} \sum_{k=1}^M \left( c_k - \frac{1}{M} \right)^2 \quad (4.4)$$

Where  $M$  is the total number of features in class  $c$ ,  $c_k$  represents the contribution of position  $k$  to the errors.

The FIB score ranges from 0 to 1, where one indicates that the changes are evenly distributed across features. In contrast, lower scores suggest that only a subset of features has been significantly altered.

### Grouping differences

Given the dimensionality of the aggregated representations, we group the features into  $k$  groups. To do so, we compute the differences first, then sort them by intensity, then split the sorted differences into  $k$  groups of equal size. Therefore, FIB- $k$  indicates that we computed the FIB score after grouping the differences into  $k$  groups. With this grouping mechanism, we can understand whether a subset of features carries the differences. For instance, with a score of 0, the differences are carried by  $\frac{1}{k}$  of the positions in the original representations. For example, if we consider 4 groups, a score of 0 means that the differences are carried by 25% of the positions in the original representations.

In summary, our methodology involves computing aggregated class representations and comparing them across various data augmentations using MSE, SSIM, and the FIB score. This

approach provides a comprehensive view of how data augmentation impacts class representations, allowing us to analyze both pixel-level and structural changes and feature-level imbalances introduced by different augmentation techniques.

## 4.4 Problem setup

We consider a dataset  $\mathcal{D}$ , composed of pairs  $(\mathbf{x}, y)$ , where  $\mathbf{x} \in \mathcal{I}$  is an image, with  $\mathcal{I}$  the image space, and  $y \in \mathcal{Y}$  is the label. Such that  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  with  $N$  the number of samples in the dataset. We consider  $\mathcal{D}_c = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_c} \subseteq \mathcal{D}$  the subset of images in  $\mathcal{D}$  that belong to the class  $c \in \mathcal{Y}$ . Let  $\phi$  be a data augmentation transformation, such that  $\phi : \mathcal{I} \rightarrow \mathcal{T}$ , with  $\mathcal{T}$  the transformed image space, which might not be the same as the image space. We aim to evaluate the impact of the data augmentation  $\phi$  on the datasets  $\mathcal{D}_c$ . To do so, we consider the original dataset  $\mathcal{D}_c$  and the transformed dataset  $\mathcal{D}_c^\phi$ . We compute the aggregated representations  $A_c$  and  $A_c^\phi$  of the original and transformed dataset. Namely,  $A_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{x}_i$  and  $A_c^\phi = \frac{1}{N_c} \sum_{i=1}^{N_c} \phi(\mathbf{x}_i)$ . We then aim at understanding changes between the aggregated representations  $A_c$  and  $A_c^\phi$ , and across the datasets  $\mathcal{D}_{c_i}$  and  $\mathcal{D}_{c_j}$  and  $\mathcal{D}_{c_i}^\phi$  and  $\mathcal{D}_{c_j}^\phi$ .

## 4.5 Experimental Setup

To evaluate the effects of various data augmentation techniques on class-wise performance, we conducted experiments using three datasets: CIFAR-10, [117], ImageNet-10 [118], and CIFAR-100 [117]. We applied a set of geometric and photometric transformations and compared the aggregated class representations before and after augmentation, using metrics such as MSE, SSIM, and FIB scores.

### 4.5.1 Datasets

In this section, we describe the datasets used to evaluate the impact of different data augmentation techniques on class representations. We utilize three widely used datasets: CIFAR-10, CIFAR-100, and ImageNet-10. We selected these datasets to provide a range of classification

challenges, from simple tasks with few classes to more complex tasks with high inter-class variability.

**CIFAR-10** [117] consists of 60,000 RGB images, each of size  $32 \times 32$ , across 10 classes with a balanced number of samples per class. Due to its simplicity and relatively low number of classes, CIFAR-10 is a benchmark dataset for Computer Vision (CV) tasks. The balanced nature of the dataset makes it ideal for testing the general impact of augmentation methods without significant bias from class imbalance.

**CIFAR-100** [117] also consists of 60,000 RGB images of the same size ( $32 \times 32$ ), but it covers 100 classes. Each class has an equal number of samples, making it more challenging than CIFAR-10. The larger number of classes and the fewer images per class provide a more complex scenario to evaluate how augmentations affect inter-class similarities and model performance.

**ImageNet-10** [118] is a subset of the ImageNet dataset (ILSVRC 2012) [118]. The original dataset contains 1,281,167 training images, 50,000 validation images, and 100,000 test images spanning 1,000 object classes. We randomly sampled 10 classes and resized the images to  $224 \times 224$  pixels for our experiments. This subset allows us to test augmentation strategies on a larger-scale, high-resolution dataset, mimicking real-world challenges while maintaining a manageable computational cost.

## 4.5.2 Data augmentations

This experiment considers widely used geometric and photometric augmentation. These augmentations are simple to implement and provide insights into how models respond to variations in spatial and visual features. Although more advanced augmentation methods, such as noise injection, random erasing, and image mixing, have been proposed in recent literature, We excluded them from our experimental setup as they are more complex and challenging to interpret. We aim to evaluate the impact of straightforward augmentations that are simpler to analyze and widely applicable.

## Geometric augmentations

We consider four geometric transformations: Horizontal Flipping, Vertical Flipping, Random Cropping with Padding, and Random Rotation.

**Horizontal Flipping:** generates a mirrored version of the image by reversing it along the vertical axis. In our experiments, we apply horizontal flipping with a probability of  $p = 0.5$ , meaning each image has a 50% chance of being flipped. Horizontal flipping helps to reduce positional biases in datasets, mainly when the orientation of the objects in the images is arbitrary. This transformation improves the model’s ability to recognize features regardless of orientation. However, in cases where object orientation is critical (*e.g.*, certain digits, medical images), horizontal flipping can generate images with altered labels.

**Vertical Flipping:** Vertical flipping mirrors the image along the horizontal axis. Like horizontal flipping, we flip the image with a probability  $p = 0.5$ . Nevertheless, this transformation may not be suitable for datasets where vertical orientation is essential (*e.g.* numbers like '9' and '6' or images where the top and bottom are clearly defined - Figure 4.1).

**Random Cropping with Padding:** reduces the visible area of the image by cropping part of it, then padding it to maintain the original dimensions. The cropping is random but constrained to ensure that essential features of the object remain within the frame. This augmentation helps reduce over-reliance on object location and positional information. Exposing the model to different parts of an object teaches it to generalize better to varying viewpoints or partial occlusions. However, Random cropping might remove critical parts of the image, especially in datasets where the object occupies a small area.

**Random Rotation:** we rotate the images randomly between  $-30^\circ$  and  $+30^\circ$ . The rotation helps the model become rotation-invariant and improves its ability to handle rotated objects in real-world scenarios. This can matter in domains such as medical imaging, where an object’s orientation may vary across scans. Nonetheless, excessive rotation can distort objects, leading to misclassification, especially for asymmetrical objects (*e.g.*, rotating the number '9' can turn it into '6' - Figure 4.1).

Using these data augmentations, we obtain four new versions of the data  $\mathcal{D}$ .

In addition to these geometric transformations, we consider photometric augmentations.

## Photometric augmentations

We consider four photometric transformations: Brightness, Contrast, Saturation, and Hue. For each of these transformations, we consider three levels of variation.

**Brightness:** alters the intensity of the light without changing the colors significantly. We adjust the brightness by a factor  $f_b \sim \mathcal{U}_{[1-b, 1+b]}$  with  $b \in \{0.1, 0.3, 0.5\}$ . Varying brightness helps models generalize to different lighting conditions, which is essential in many real-world applications, such as outdoor object detection and medical imaging, where lighting can differ significantly. Extreme brightness shifts may wash out or darken critical features, making object recognition more challenging, especially in fine-grained classification tasks.

**Contrast:** amplifies or dulls the lighter / darker areas of the image. We adjust the contrast by a factor  $f_c \sim \mathcal{U}_{[1-c, 1+c]}$  with  $c \in \{0.1, 0.3, 0.5\}$ . This helps models handle variations in image clarity, crucial in environments where contrast can vary, such as low-light photography or medical scans with different contrast levels. Nonetheless, significant contrast alternation can obscure fine details, mainly when applied to highly detailed or noisy images.

**Saturation:** makes color more vibrant or shifts it toward the grayscale. We adjust using a factor  $f_s \sim \mathcal{U}_{[0, 1-s]}$  with  $s \in \{0.1, 0.3, 0.5\}$ . This transformation allows models to learn from images with different color intensities, making them more robust to color variations. Over-saturation can distort colors and reduce the model’s ability to recognize objects based on their natural color cues.

**Hue:** refers to the dominant wavelength of colors; by changing it, we shift the color spectrum of the image. We adjust the hue by a factor  $f_h \sim \mathcal{U}_{[-h, +h]}$  with  $h \in \{0.1, 0.3, 0.5\}$ . These variations make models more resilient to different lighting conditions and color shifts, particularly useful in natural images where the color balance can vary depending on the time of day,

weather, or camera settings. However, excessive hue shifts may drastically change the image’s appearance, causing it to lose critical color-related features crucial for object identification. Accounting for the transformations and their levels of variation, we obtain 12 new versions of the data.

By combining the geometric and photometric transformations, we obtain 16 new versions of the data (4 geometric and 12 photometric).

## Flowchart

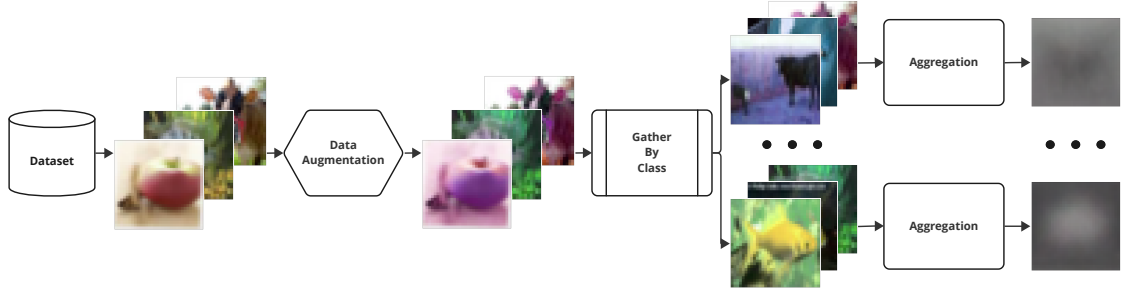


Figure 4.3: Flowchart of the representation aggregation pipeline on CIFAR-100 using the strongest HUE perturbation. Given a dataset, we consider a single pass through the data augmentation method. Therefore, we obtain a single version of an image  $x$ , its transformed versions  $\phi(x)$ . Then, we regroup all images that belong to the same class. Finally, we aggregate the transformed representations for each class.

### 4.5.3 Aggregated representations

We note  $A^\phi$  as the vector of aggregated representations for a given dataset  $\mathcal{D}$  and data augmentation  $\phi$ . Namely,  $A^\phi = [A_1^\phi, \dots, A_C^\phi]$ , where  $A_c^\phi$  is the aggregated representation for class  $c$  and data augmentation  $\phi$ .

### Original dataset’s aggregated representations: $A^{\text{id}}$

First, we compute the original dataset’s aggregated representation  $A^{\text{id}}$ . Namely, we group all the images belonging to a given class and take the mean of their pixel values. In addition to

these aggregated representations, we compute the aggregated representations of the augmented dataset.

### **Augmented dataset's aggregated representations: $A^\phi$**

Then, we compute the aggregated representations of the augmented dataset:  $A^\phi$ . We apply the data augmentation  $\phi$  for each image and compute the aggregated representation of the transformed images based on their classes.

## **4.6 Results**

We evaluate the impact of the data augmentations on the aggregated representations. First, we provide examples of aggregated representations (section 4.6.1). Second, we show how differences between aggregated representations can be informative (section 4.6.1). Then, we study how applying data augmentations modifies the similarities between aggregated representations (section 4.6.2). Finally, we show that data augmentations impact classes differently. We conduct the experiments using CIFAR-10, ImageNet-10, and CIFAR-100. Nonetheless, we focus on visual examples from CIFAR-10 and CIFAR-100. We compute each version of the dataset's aggregated representations and derive the scores (SSIM, MSE, FIB-10). We grouped FIB scores into 10 groups to better summarize the differences between aggregated representations. Finally, we report results by considering the average scores obtained across all three versions of the augmented datasets. For instance, we generated 3 versions of CIFAR-10, for which we applied random vertical flipping. We then aggregated MSE scores obtained across all 3 versions.

### **4.6.1 Aggregated representations of the original dataset**

This section analyzes the aggregated representations derived from the original datasets. These aggregated representations show us that the average image provides information about the class's general shapes, backgrounds, and colors. (Figures 4.4 and 4.5)

Figure 4.4 shows that in CIFAR-10, color and patterns are specific to certain classes in the aggregated representations. For instance, the "airplane" and "ship" classes usually contain

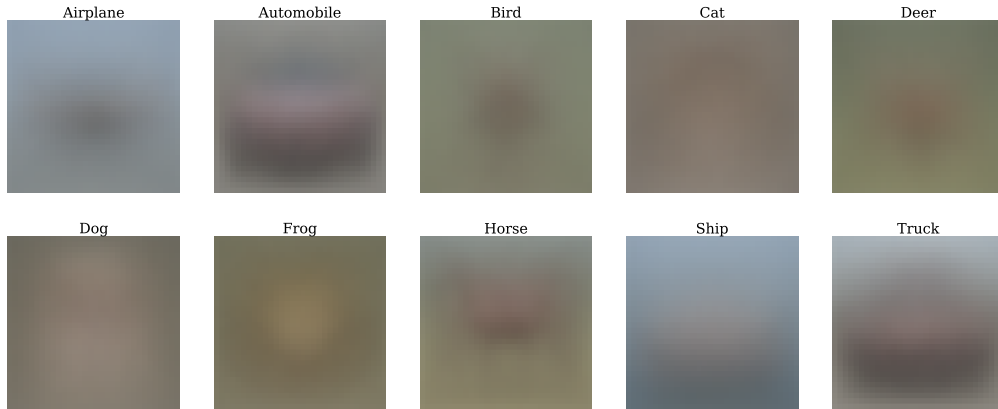


Figure 4.4: Aggregated Representation over CIFAR-10’s training set. Each square represents the average representation for a given class. For instance, the top-left element represents the average representation of the "airplane" classes. We can observe that, even based on the average representation, both airplane and boat classes exhibit similar color schemes with more nuances of blue compared to other classes.

mostly blue portions surrounding the center, with the "ship" having darker shades of blue in the lower half of the image. One can expect the presence of such blue shades given the contextual bias in which the pictures are taken: aerial and maritime elements are more likely to be represented next to the sky and the sea than terrestrial animals, and objects would be [128]. Similarly, we can observe that the "bird," "deer," and "horse" are often surrounded by green colors and that, on average, the central portions are associated with shades of brown.

In figure 4.5, we consider the aggregated representation from the 100 classes from CIFAR-100. Similarly to CIFAR-10 and ImageNet-10, we observe colors and locations characteristic of certain classes. For instance, "Dolphin" and "Shark" classes tend to have darker shades of blue surrounding them than "Whale" and "Ray" classes. It also exhibits the difference in sizes for the different classes. For example, the average representation of the "Apple" class in CIFAR-100 takes up a similar dimension as the different classes representing trees such as "Oaktree" or "Maple tree." Furthermore, we could expect that most samples associated with the "Apple" class were red, indicating that green apples might be less often represented than red ones. We confirmed this by visually examining the images associated with the "apple" class. The location bias in observed ImageNet-10 concurs with the observation that ImageNet has an Object-Centric bias[146], namely, most images contain a single object centered. Visually, the



Figure 4.5: Aggregated Representation over CIFAR-100’s training set. Each square represents the average representation for a given class. For instance, the top-left element represents the average representation of the "airplane" classes. We can observe that, even based on the average representation, both airplane and boat classes exhibit similar color schemes with more nuances of blue compared to other classes.

aggregated representations of CIFAR-10 and CIFAR-100 hint towards a similar bias.

## Aggregated representations of the augmented dataset

From these aggregated representations, we can observe that in some classes, such as "Cat" (third row) and "Dog" (fourth row), the aggregated representations appear visually similar across augmentations (Figure 4.6). Meanwhile, for classes such as "Airplane" (first row) and "Horse" (fifth row), vertical flipping leads to more visible changes in the aggregated representations. Furthermore, augmentations such as random cropping with padding and random rotation lead to aggregated representations with dark borders, which increases common visual appearances across classes.

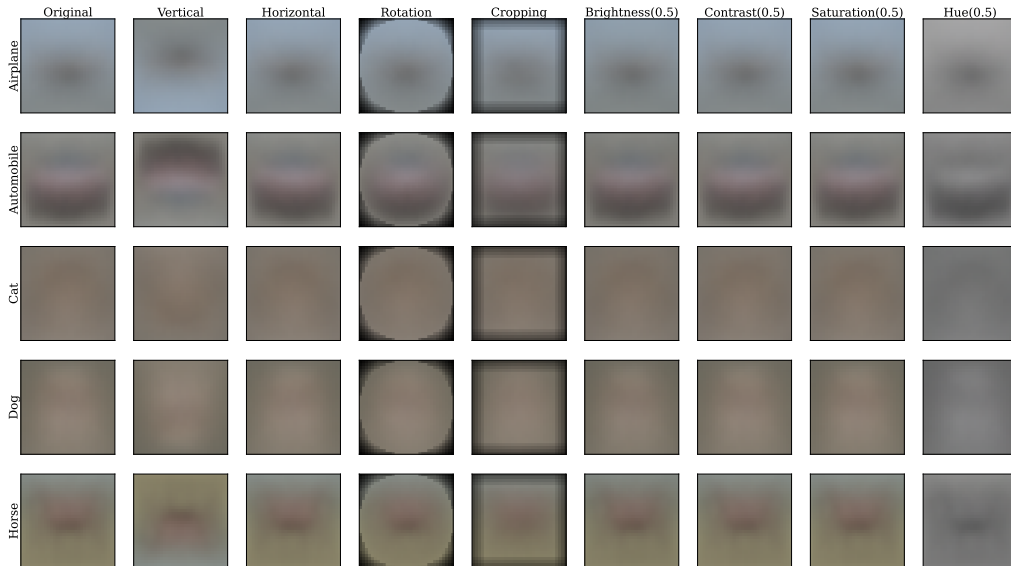


Figure 4.6: Examples of data augmentations applied to CIFAR-10

## How are aggregated representations different?

Here, we focus on the differences between the aggregated representations by computing the squared differences between the aggregated representations. The differences between the aggregated representation are minor in the CIFAR-10 dataset, ranging from 0.000 to 0.136, and displaying these differences would lead to primarily dark patches. Instead, we apply a Min Max Normalization over each difference between aggregated representations. The resulting re-scaled

differences are shown in figure 4.7. The re-scaling allows for identifying colors and locations that are most different between aggregated representations. As such, figure 4.7 further confirms that the difference between the average representation of the "airplane" class and the other classes mostly came from the blue channel, except for the "ship" class. Moreover, the difference between the "airplane" and "ship" classes is primarily due to the lower half of the image.

### 4.6.2 How do data augmentations impact the similarities of aggregated representations?

We analyze the similarity between aggregated representations using Structural Similarity Index (SSIM). Figure 4.8 shows the densities of the SSIM values computed between the aggregated representations for each data augmentation on CIFAR-100. The distributions belong to similar families and are left-skewed. Aggregated representations are mostly similar, and applying the different augmentations does not change this dynamic. Nevertheless, three augmentations set themselves apart: vertical flipping, random cropping with padding, and rotations. These augmentations exhibit more similarities between the aggregated representations. One can ascribe these similarities to the stable elements introduced by these augmentations in the aggregated representation. For instance, applying rotations on individual images leads to carving circles around the aggregated representations. These circles can be found for all aggregated representations, contributing to increased class similarities.

We observed similar patterns in CIFAR-10 and ImageNet-10, except for random cropping with padding on ImageNet-10. The amount of padding pixels can explain such patterns, representing a much smaller proportion of the images. We consider ImageNet-10 images with sizes of  $224 \times 224$ , while both CIFAR-10 and CIFAR-100 images have a size of  $32 \times 32$ .

### 4.6.3 Do data augmentations impact all classes the same way?

Using ten groups, we computed different pairwise metrics between the aggregated representations: MSE and FIB scores. We considered the original dataset as the baseline to determine the evolution of these different metrics across data augmentation. We focus on the range the

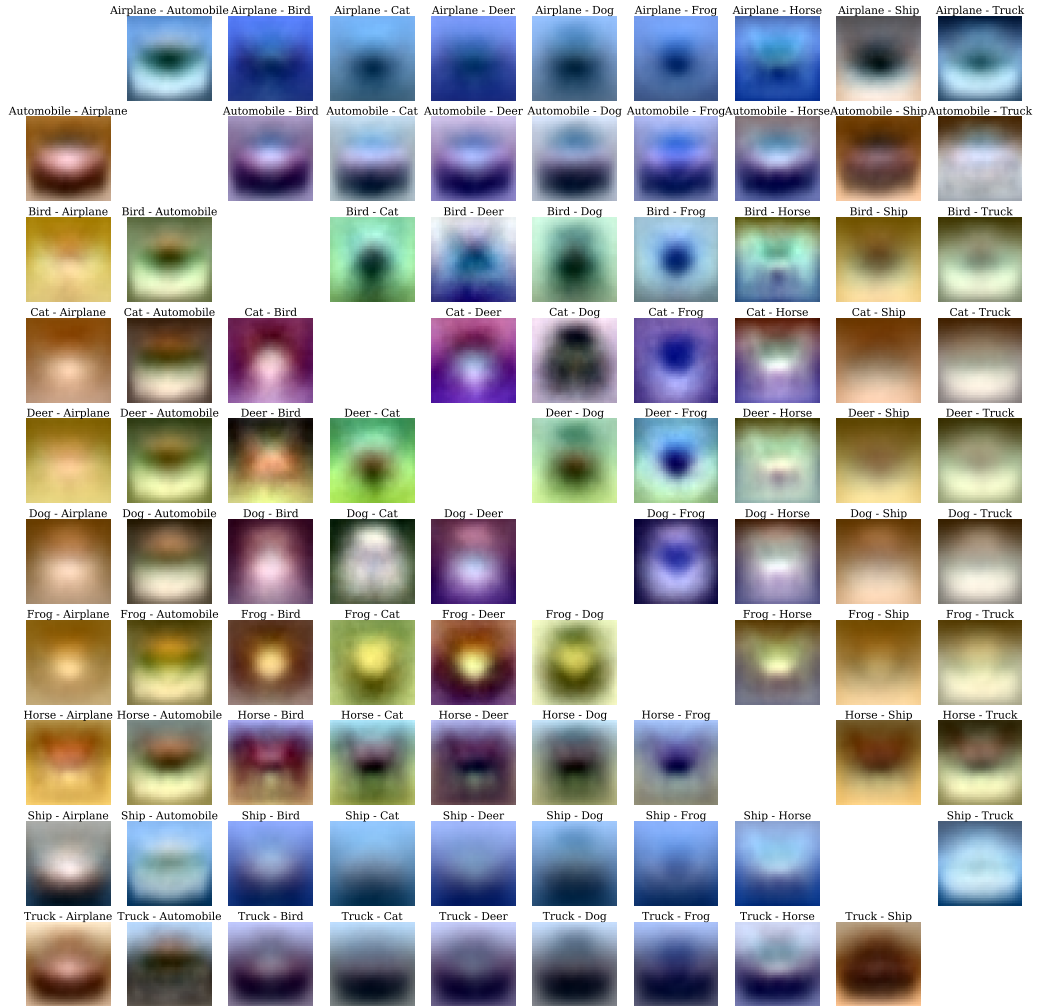


Figure 4.7: Normalized difference between aggregated representations of classes of CIFAR-10. We observe that "Airplane" and "Ship" classes have differences with other classes that are mostly from the blue channel. This is due to the context in which the different pictures are taken, as these classes more likely to be against the sky or water.

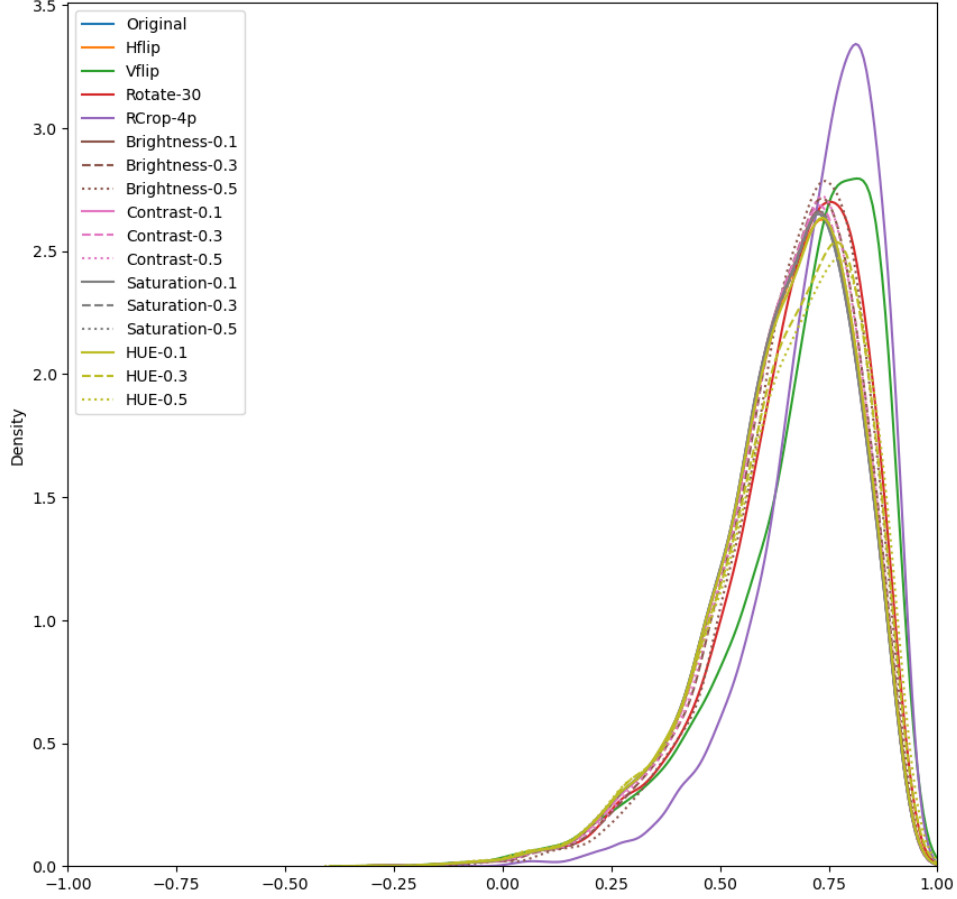


Figure 4.8: Density of Structural Similarity Index (SSIM) values between aggregate representations on CIFAR-100. Most data augmentations do not change the similarity between aggregated representations. The distributions are left-skewed with peaks within  $[0.6, 0.8]$

different metrics cover rather than the average values. By focusing on the minimal and maximal values from pairwise comparisons, we can determine how globally the augmentation impacts specific classes.

Let us consider a class  $i$  and the MSE. We obtain all the MSE values between the aggregated representation of class  $i$  on the original dataset  $A_i$  and the other aggregated representations  $A_j$  with  $j \neq i$ . In the case of CIFAR-100, we obtain 99 values for class  $i$ . We obtain the minimal and maximal value noted  $\min_i$  and  $\max_i$  respectively. This process is repeated on aggregated representations from an augmented version of the data using augmentation method  $\phi$ , *i.e.*  $A_i^\phi$ . We obtain their minimal and maximal values  $\min_{i,\phi}$  and  $\max_{i,\phi}$  respectively. We note the relative changes between the minimal and maximal values as  $R_{\min,i}$  and  $R_{\max,i}$ . Thus, each class  $i$  can be represented as a point  $(R_{\min,i}, R_{\max,i})$  in a MMRCQ plot (presented in Section

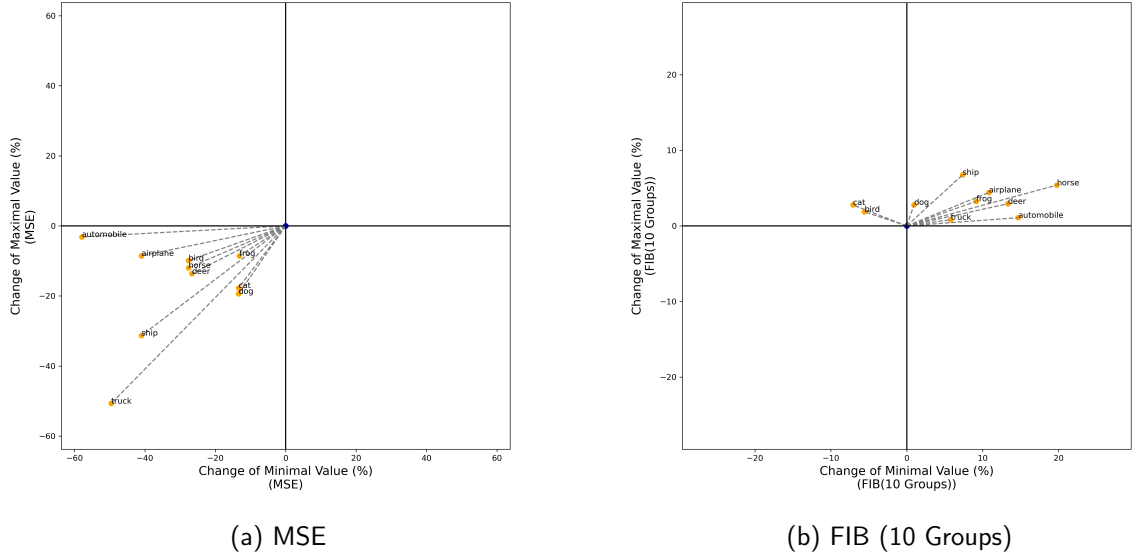


Figure 4.9: CIFAR-10 MMRCQ plot between the original data and the data after applying vertical flipping. When considering the MSE scores, the "truck" class has the strongest relative changes in its minimal and maximal distances from other classes.

### 3.3)

These minimal and maximal values allow for determining the percentage of classes in each quadrant and the minimal and maximal changes observed for any augmentations compared with the original dataset. (Table 4.1).

In addition to the number of classes per quadrant, we report Minimal relative change and maximum relative change as the Euclidean Distance to the origin for the closest and furthest classes. These numbers provide information about the range of changes induced by the different augmentations on MSE or FIB in 10 groups.

## CIFAR-10

Figure 4.9 focuses on how vertical flipping affects MSE and FIB scores between aggregated representations on CIFAR-10. The "truck" class is the most impacted when using MSE scores. Both the minimal and maximal values of the "truck" class have decreased, *e.g.* it is in the lower left quadrant. Vertical flipping leads to aggregated representations from other classes closer to the "truck" class than in the original space. Figure 4.9b uses the FIB scores with ten groups

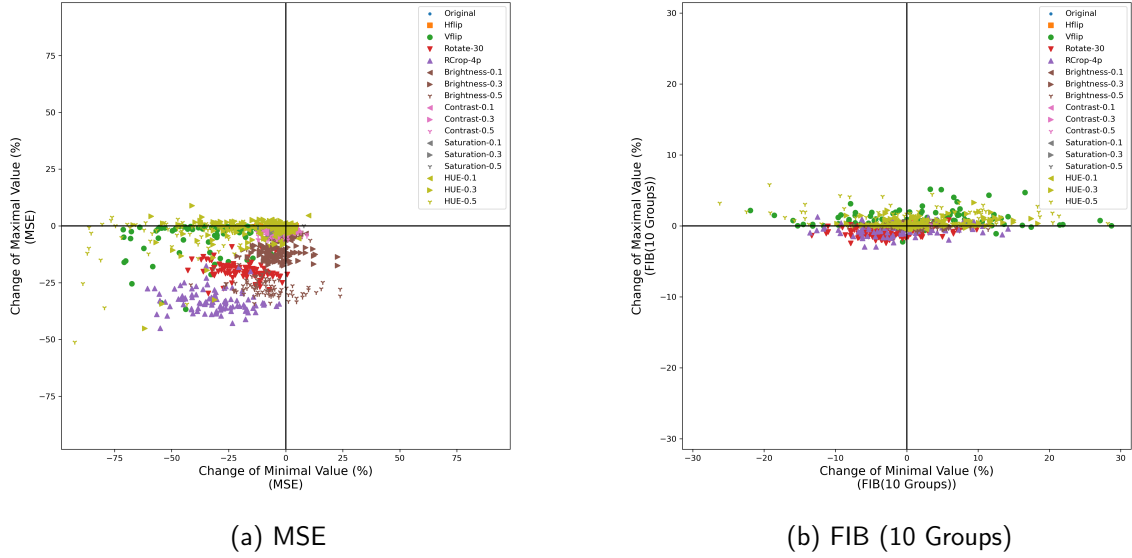


Figure 4.10: CIFAR-100 Relative Change Quadrant plot between the original and the training sets post vertical flip augmentation. When considering the MSE scores, the truck class has the strongest relative impact on its differences with the aggregated representation of other classes.

instead of the MSE. We observe that 8 out of 10 classes now have errors that are better spread across the image overall, *e.g.* they are located in the upper right quadrant.

## CIFAR-100

We follow a similar procedure on CIFAR-100 as the one applied on CIFAR-10. Most data augmentation strategies reduced the minimal and maximal MSE values for all classes, 63% at the lowest for horizontal flipping and above 99% for vertical flipping, random rotations, and random cropping with padding. Table 4.1 summarizes the locations of the classes per quadrants for geometric and photometric augmentations, respectively. We observe that changes in brightness and contrast, regardless of the intensity of the parameters, mostly lead to decreased minimal and maximal values of MSE. More than 69% of the classes follow that pattern. Most classes see their maximal values increase for saturation-based augmentations, with a balanced mix of minimal values increasing or decreasing. Such an increase indicates that saturation-based augmentations pushed most classes further apart with respect to the intensity of differences. Most classes have seen their minimal values decrease for hue-based augmentations, associated with a balanced mix of increasing or decreasing maximal values.

We conducted a similar analysis using the Feature Impact Balance score. From figure 4.10b, we observe that most changes are associated with relative changes in the minimal values, as the variations along the y-axis are much more minor than those along the x-axis. Furthermore, while the MSE changes were mainly concentrated in the lower left quadrant, changes in the FIB scores are more spread across quadrants.

From table 4.1, we observe that among geometric augmentations, vertical flipping made most classes have more balanced differences with other classes as 56% of the classes increase both their minimal and maximal balance scores. The horizontal flipping exhibit changes across all four quadrants, 52% of the classes saw a decrease of their minimal balance. Random crop favored decreases in minimal and maximal values with 43% of the classes following such pattern and an accumulated 70% of classes seeing a decline in their maximal values. Random rotation within  $30^\circ$  favored a decrease of the balance scores, with 75% of the classes decreasing their minimal and maximal balance score, and 95% decreased their maximal values.

This can be explained by the effect of these augmentations on the aggregated representations for both random crop and random rotations. The effect of random rotation carves a circle around the aggregated representation, while the random cropping with padding carves a square around it. These carved portions can be observed across all aggregated representations. Therefore, pixels from these carved portions barely contribute to the difference between the aggregated representations, inducing an imbalance in contribution.

From table 4.1, we observe the effect of the different photometric augmentations. Brightness-based augmentations spread across all four quadrants with a strong effect on lowering the maximal balance. Contrast-based augmentations favored an increase of maximal values with at least 70% of classes increasing their maximal FIB score. Saturation-based augmentations spread across all four quadrants with a stronger effect on increasing classes' minimal and maximal FIB scores. Finally, hue-based augmentations favored an increase in maximal FIB scores, with at least 70% classes increasing their maximal FIB scores. The effect of hue-based augmentation on the aggregated representation tends to bring them closer to gray scales images, making the difference less dependent on the image colors. This indicates that while most data augmentations reduce the difference between the aggregated representations of the classes, as seen using the MSE scores, certain classes have changes in the imbalance of their differences with other classes.

Table 4.1: Quadrant Attribution for Data Augmentation effects of MSE and FIB score on CIFAR-100

| Geometric Augmentation   | Min(↑) MSE | Max(↑) FIB | Min(↓) MSE | Max(↑) FIB | Min(↓) MSE | Max(↓) FIB | Min(↑) MSE | Max(↓) FIB | Min RC MSE | RC FIB | Max RC MSE | RC FIB |
|--------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|--------|------------|--------|
| Hflip                    | 7          | 22         | 4          | 37         | 63         | 15         | 26         | 26         | 0.07       | 0.00   | 14.45      | 1.92   |
| Vflip                    | 0          | 57         | 0          | 39         | 100        | 1          | 0          | 3          | 1.12       | 0.21   | 72.62      | 28.71  |
| Rotate-30                | 0          | 1          | 0          | 4          | 99         | 75         | 1          | 3          | 11.65      | 0.21   | 47.16      | 13.41  |
| RCrop-4p                 | 0          | 19         | 0          | 11         | 100        | 43         | 0          | 27         | 25.23      | 0.38   | 71.13      | 14.2   |
| Photometric Augmentation | Min(↑) MSE | Max(↑) FIB | Min(↓) MSE | Max(↑) FIB | Min(↓) MSE | Max(↓) FIB | Min(↑) MSE | Max(↓) FIB | Min RC MSE | RC FIB | Max RC MSE | RC FIB |
| Brightness-0.1           | 0          | 27         | 0          | 18         | 69         | 20         | 31         | 35         | 0.31       | 0.02   | 9.42       | 3.65   |
| Brightness-0.3           | 0          | 25         | 0          | 27         | 74         | 12         | 26         | 36         | 4.57       | 0.06   | 28.21      | 7.29   |
| Brightness-0.5           | 0          | 23         | 0          | 11         | 81         | 20         | 19         | 46         | 11.18      | 0.08   | 49.04      | 11.41  |
| Contrast-0.1             | 0          | 48         | 0          | 22         | 70         | 12         | 30         | 18         | 0.2        | 0.01   | 2.12       | 0.49   |
| Contrast-0.3             | 0          | 57         | 0          | 25         | 73         | 5          | 27         | 13         | 0.5        | 0.0    | 8.21       | 1.86   |
| Contrast-0.5             | 0          | 68         | 0          | 16         | 82         | 7          | 18         | 9          | 2.47       | 0.03   | 13.72      | 3.51   |
| Saturation-0.1           | 43         | 32         | 45         | 24         | 4          | 13         | 8          | 31         | 0.15       | 0.0    | 1.74       | 0.36   |
| Saturation-0.3           | 31         | 36         | 49         | 23         | 14         | 22         | 6          | 19         | 0.12       | 0.0    | 6.75       | 0.82   |
| Saturation-0.5           | 32         | 45         | 44         | 18         | 18         | 13         | 6          | 24         | 0.12       | 0.0    | 12.13      | 1.45   |
| hue-0.1                  | 15         | 53         | 43         | 22         | 32         | 6          | 10         | 19         | 0.22       | 0.02   | 23.77      | 5.23   |
| hue-0.3                  | 0          | 60         | 47         | 34         | 50         | 2          | 3          | 4          | 2.33       | 0.45   | 76.48      | 18.38  |
| hue-0.5                  | 0          | 48         | 38         | 50         | 61         | 2          | 1          | 0          | 1.38       | 0.15   | 105.72     | 28.18  |

## 4.7 Discussion

### 4.7.1 Conclusion

In this work, we explored commonly used image augmentation techniques such as geometric transformations (*e.g.*, flipping, cropping, rotation, translation) and photometric transformations (*e.g.*, changes in brightness, hue, and saturation) across different image classes. These augmentations are widely applied and demonstrated to improve classification performance [133], showing varying effects on class-specific representations in the transformed input space.

We leveraged the Feature Impact Balance (FIB) (Chapter 3) score to quantify the balance of contributions from different parts of the feature space. The FIB score ranges from 0 to 1 and helps identify whether the differences between two feature vectors are concentrated in specific regions or distributed more evenly. The FIB provides a novel metric for evaluating the localized impact of augmentations on class-specific representations.

Additionally, we visualized the impact of augmentations using Min-Max Relative Change Quadrant (MMRCQ) plots, revealing that while most augmentations reduced the intensity of differences between aggregated representations (as measured by MSE), certain transformations like vertical flipping and random cropping with padding induced larger shifts. These augmentations caused a significant reorganization of the relationships between classes in the input space, particularly in datasets like CIFAR-100, where some classes experienced up to a 72% change

in their minimal and maximal distances from other classes.

Although our analysis relied on class-aggregated representations, which provide a coarse-grained overview of class-level feature distributions, it demonstrated that some augmentations disproportionately affect certain classes. Future research could refine this by focusing on subgroup-level analysis within each class.

Moreover, while the FIB score helps quantify balance, it does not measure the intensity of feature errors. Integrating this score with additional metrics could further enhance our understanding of how data augmentations impact different classes. This analysis opens new opportunities to leverage class-specific augmentation effects in building more robust models, especially within the Out-of-Distribution (OOD) paradigm.

### 4.7.2 Limitations

Our study was limited to commonly used, easily implemented augmentation techniques that require minimal computational resources. However, recent advancements in GAN-based augmentation methods allow for the generation of synthetic data that better captures the complexity of real-world distributions [201, 170, 194, 164]. These methods could provide more nuanced control over the type of augmentation and lead to more targeted feature learning.

Another limitation comes from using class-aggregated representations, which provide a global perspective of each class. While this approach highlights overall trends, it may miss subtle intra-class variations or details at the subgroup level, which could further inform model performance and augmentation strategies.

### 4.7.3 Perspectives

In future work, we will examine how augmentations affect the feature spaces learned by Deep Neural Network (DNN) models rather than focusing solely on the input and transformed image spaces. Exploring the behavior of augmentations in the deeper layers of neural networks could provide new insights into their influence on learned representations.

One could also investigate augmentation strategies targeting class-level differences to enhance

model robustness in both In-Distribution and Out-of-Distribution settings. Combining classical and GAN-based augmentation techniques may offer a hybrid approach that capitalizes on the strengths of both methods to produce more resilient models across a broader range of tasks.

# Chapter 5

## Leveraging foundation models trained on non-medical data to conduct multiple classification tasks for automated dysarthria assessments.

In this chapter, we aimed at leveraging foundation models trained on non-medical speech data to extract useful representations for a medical application: Automatic Dysarthric Assessment [21]. XC designed and implemented the evaluation pipeline, adapted the codes, ran the experiments, conducted the analyses, and wrote the manuscript. RA and XC jointly conducted the feature extraction for the audio samples.

### 5.1 Introduction

#### 5.1.1 Challenges of medical applications

Training machine learning models for medical applications presents challenges, primarily due to the difficulties in acquiring and labeling data. Medical data is often sensitive and ruled by strong privacy regulations, making it difficult to share across institutions or researchers. Furthermore, medical data can require specialized equipment, such as MRI or PET scans,

which are costly and time-consuming. Additionally, trained professionals are often required to label medical data. This resource-intensive process can be a bottleneck for providing machine learning solutions. Nonetheless, models trained on large-scale non-medical datasets exist that can be used as foundation models for non-medical tasks.

### 5.1.2 Foundation models and downstream tasks

Foundation models, which are large-scale models pre-trained on large datasets, offer potential solutions to the challenges of medical data scarcity. These models learn robust representations from large datasets, often using self-supervised learning techniques. By learning from the data without relying on labels, foundation models can better generalize to new tasks with limited labeled data.

These models have been successfully applied in domains such as Computer Vision (CV) and Natural Language Processing (NLP). However, most foundation models are trained on non-medical data, prompting the question of whether their learned representations can be effectively transferred to medical applications.

This chapter explores the transferability of foundation models, focusing on speech data and their use in automated dysarthria assessments.

### 5.1.3 Our contributions

This chapter investigates whether foundation models trained on non-medical speech data can be leveraged to assess dysarthric speech. Our main contributions are as follows:

- We develop a visualization tool for dysarthria assessment results that provide insights into the speech samples contributing to classification performance.
- We use the proposed tool to empirically evaluate different representations (e.g., acoustics and self-supervised methods) and classifiers (e.g., Logistic Regression (LR) and Multi-Layer Perceptron (MLP)), on binary and multi-class tasks (e.g., detection, word, and intelligibility classification).

- Our tool can provide a better understanding of types of impairments leading to empirical features, usable for developing programs that facilitate the identification of disorders and their characteristics.

Our analysis is conducted under various experimental conditions: default, noisy, and enhanced to simulate real-world scenarios. Notably, our tool reveals that using HuBERT[100] features significantly outperforms acoustic algorithms in severity classification tasks under all three conditions. By providing insights into the specific speech impairments contributing to classification performance, our tool offers valuable guidance for identifying speech disorders and developing more tailored therapeutic programs. By addressing the challenge of limited labeled data in medical applications, we aim to demonstrate that foundation models, even those trained on non-medical data, can provide valuable insights and improve the accuracy of dysarthria assessments.

## 5.2 Background

### 5.2.1 What is Dysarthria?

Dysarthria is a motor speech disorder involving a lack of articulatory control and muscle weakness [192, 112, 59]. This condition affects speech rate, dynamic amplitudes and pitches, and how the spoken word is pronounced, making communication difficult for those affected. Dysarthria can result from physical injuries that damage the neural cells in the brain, such as strokes [134]. All of these contribute to unintelligible speech and difficulty understanding due to the inaccurate articulation of phonemes and abnormal speech patterns [131].

The classification of dysarthria can help diagnose the disorder, determine the appropriate treatment, and conduct appropriate speech therapy sessions [195]. Huang et al. have [102] shown that machine learning-based assessments achieve high accuracy. However, the methodology used to evaluate the assessment models could overestimate their generalizability. Some models were trained and evaluated on the same speakers (using different audio recordings) or a single unseen speaker per intelligibility class [102]. However, the scarcity of available dysarthric speech data makes it challenging to train models that generalize well to new patients.

### 5.2.2 Towards automated dysarthria assessment

Trained clinicians usually perform dysarthric intelligibility assessments based on how many words they can understand. Nonetheless, these assessments can be subjective and time-consuming.

#### Dysarthria Automatic Assessments

Dysarthria intelligibility assessment is typically carried out in two stages [90]: **Training Stage** and **Assessment Stage**.

1. **Training stage:** one trains a computational model using speech samples from patients labeled with their respective intelligibility levels. The model learns to associate specific acoustic features with these intelligibility classes.
2. **Assessment stage:** once trained, the model can classify the intelligibility of new speakers by comparing their acoustic features to those seen during training.

There are two main approaches to automatic intelligibility assessment: **reference-free** and **reference-based methods**.

1. **Reference-free** methods focus on developing classification models that do not rely on comparisons to healthy speech. Instead, they extract acoustic features directly from dysarthric speech, identifying patterns that are highly correlated with intelligibility [64]. These methods operate solely on the assumption that certain acoustic properties, such as pitch and formant structure, indicate how understandable the speech is without needing a benchmark from healthy speech.
2. **Reference-based** methods compare dysarthric speech to a standard derived from healthy speakers. These methods use healthy speech data to establish a baseline for what constitutes intelligible speech. For example, Automatic speech recognition (ASR)-based approaches analyze the difference between how healthy and dysarthric speech is processed. These models exploit the fact that ASR systems trained on healthy speech tend to perform

poorly on dysarthric speech, and the accuracy of the ASR system typically declines as the severity of the speech impairment increases [192, 173]. By measuring this performance drop, the system can estimate the level of speech intelligibility in dysarthric speakers.

Automated assessments offer a way for objective evaluation as they can provide quantitative information on speech impairment. This quantitative evaluation can open the way to more precise monitoring over time, early detection, and remote assessments.

### 5.2.3 Learning general-purpose representations via Self-Supervised Learning

Self-Supervised Learning (SSL) is a machine learning paradigm that involves training models on data without explicit labels. The model is trained to learn the underlying structure of the data by using pretext tasks.

### 5.2.4 Transferring self-supervised representations to dysarthria assessment

Transfer Learning is a machine learning technique that uses a pre-trained model on a first task onto a second task. The assumption is that the knowledge gained from the first task is relevant to the second task. Transfer learning is a common practice in Deep Learning, as it can help improve performance on tasks with limited training data.

Two common approaches for transfer learning are feature extraction and fine-tuning:

1. **Feature extraction:** The pre-trained model is used as a fixed feature extractor. The output of the pre-trained model is used as input to a new model for the task. This approach assumes that the new representation learned by the pre-trained model contains relevant information for the new task without further training.
2. **Fine-tuning:** The pre-trained model is used as a starting point for the new task. The model is then further trained on the new task. The assumption is that the original model's weights will serve as good starting points for the new task.

In the following, we focus on the feature extraction approach as the amount of labeled data is quite limited and coarse. For intelligibility assessment, the labels are given at a patient level, which is far fewer than audio recordings.

## 5.3 Methodology

Because the quality of recordings, the representations used, and the classification algorithm can influence the effectiveness of dysarthria assessments, we aim to develop an interpretable tool that facilitates understanding of their outputs. We provide an overview of the proposed tool in Figure 5.1, which can be easily adapted to extract various features, followed by multiple classification algorithms. Then, our tool aggregates the results per patient to verify the assessment results' reliability. Aggregation outputs could be interpreted as intelligibility classes, such as low, mid, and high levels, and could provide clinicians with an interpretable classification of the speaker's intelligibility.

### 5.3.1 Problem Setup

We consider a dataset  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^N$  where  $\mathbf{x}_i \in \mathbb{R}^d$  is a feature vector and  $y_i \in \mathbb{R}^C$  is a class label. Let  $\mathcal{D}_{\text{Detection}}$ ,  $\mathcal{D}_{\text{Words}}$ , and  $\mathcal{D}_{\text{Intelligibility}}$  be the datasets for the detection, words, and intelligibility tasks, respectively. Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  be a feature extractor, and  $g : \mathbb{R}^k \rightarrow \mathbb{R}^C$  be a classifier. Let  $r_i = f(\mathbf{x}_i)$  be the feature vector for the  $i$ -th recording as extracted by  $f$ . We evaluate the performance of the feature extractor  $f$  and the classifier  $g$  by considering the accuracy of the model  $g(f(\mathbf{x}))$  on the dataset  $\mathcal{D}$ . Let  $\mathcal{D}_{\text{Detection}}^{\text{Default}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Default}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Default}}$  be the datasets of the original audio recordings for the detection, words, and intelligibility tasks. We generate  $\mathcal{D}_{\text{Detection}}^{\text{Enhanced}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Enhanced}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Enhanced}}$  by applying the noise reduction transformation to  $\mathcal{D}_{\text{Detection}}^{\text{Default}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Default}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Default}}$ , respectively. We generate  $\mathcal{D}_{\text{Detection}}^{\text{Noisy}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Noisy}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Noisy}}$  by applying the noise addition transformation to  $\mathcal{D}_{\text{Detection}}^{\text{Default}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Default}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Default}}$ , respectively.

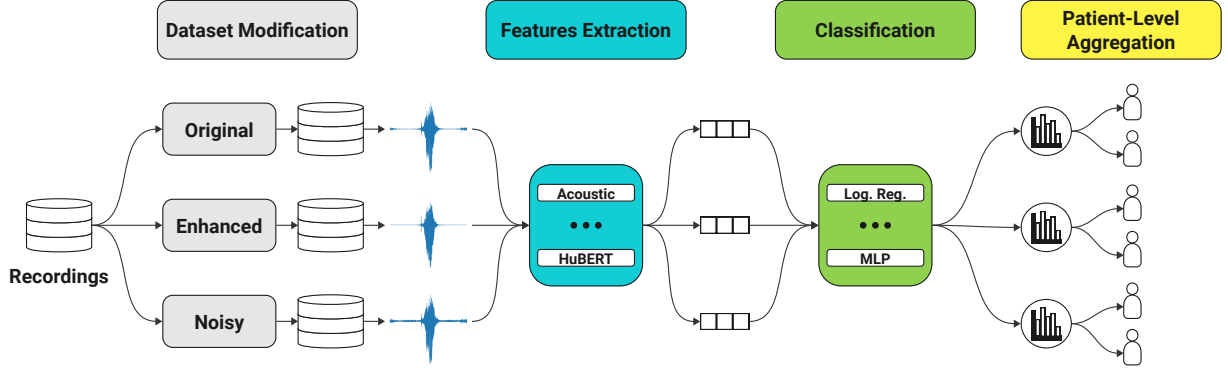


Figure 5.1: Our evaluation pipeline. We start by generating the different versions of the dataset then use different feature extractors to obtain representations of the audio recordings. Using these representations we train diverse classification models and predict different properties. Finally, we perform patient level aggregation, for instance for the intelligibility level we consider the intelligibility class that is the most associated with the given patient.

### 5.3.2 Overview

Several factors, such as the quality of speech recordings, feature representations, and classification algorithms, can significantly impact the accuracy and reliability of dysarthria assessments. To address this, we propose the development of an interpretable tool that provides insights into these assessments and can help clinicians better understand the results.

Figure 5.1 shows an overview of the proposed tool. The tool is designed to be flexible, allowing for extracting various speech features and applying multiple classification algorithms. The proposed tool aggregates results per patient, providing a comprehensive view of their speech intelligibility. These aggregated outputs can be categorized into intelligibility classes (*i.e.*, low, mid, and high), offering an interpretable assessment of the speaker’s intelligibility levels.

### 5.3.3 Experimental Settings

We conduct our experiments using multiple feature representations, including hand-crafted acoustic and self-supervised features. We evaluate these features across different classification tasks related to dysarthria.

## Dataset.

We use the UA-Speech dataset [111], which contains speech recordings from both control and dysarthric speakers. The dataset includes recordings from 13 healthy speakers and 15 speakers with dysarthria, covering a vocabulary of 455 distinct words. These words include digits, radio alphabets, computer commands, common words, and uncommon words. Speakers are divided into four severity levels: high (H), mid (M), low (L), and very low (VL).

## Extracted Features

**Hand-Crafted Features.** Regarding hand-crafted features, we extracted acoustic articulation, voice, and prosody measures using PRAAT [109]. Among these features, we include the mean harmonic-to-noise ratio (HNR), the fraction of locally unvoiced frames, the number of voice breaks, the degree of voice breaks, the mean and standard deviation of pitch, jitter, shimmer, and cepstral peak prominence (CPP) [169].

**Self-Supervised Features.** For the Self-Supervised features, we use the pre-trained models implemented by SUPERB[190] benchmark (<https://github.com/s3prl/s3prl>). We selected methods such as wav2vec2 [6], Modified CPC [150], and HuBERT[100] which have been successfully used for a variety of speech classification tasks [175]. To extract the features, we start from the raw .wav files, from which we obtain the waveform and sample rate. If necessary, we perform a resampling. Then, each waveform is fed into a pre-trained model, which is readily available under their respective algorithms, that is, Wav2vec2, HuBERT, and Modified CPC, through the SUPERB pipeline. We obtain the last hidden state representations and aggregate them. Thus, we get a representation for each audio recording in the dataset.

## Classification Tasks

**Dysarthria Detection.** We consider dysarthria detection as a binary classification task; an individual is either in the control group, class 0, or the dysarthria group, class 1.

**Word Classification.** Dysarthric patients often find it easier to produce isolated words than continuous sentences [193]. In this context, isolated word recognition refers to converting a single spoken word into its corresponding text format [73]. Another related task, keyword spotting, focuses on detecting specific words or phrases within longer spoken sentences or utterances [31]. However, this study focuses on isolated word recognition as a multi-class classification problem. Specifically, we classify each spoken word into one of 155 distinct words from the dataset.

**Intelligibility Classification.** Dysarthria intelligibility classification is a multi-class classification task. The goal is to classify the severity level of dysarthria. The severity of dysarthria determines the different degrees of intelligibility. We consider 5 classes, 4 provided by the UA-Speech dataset, and an additional one for control speakers.

## Classification Models

To perform the classifications, we consider different classifiers  $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^C$ , where  $C$  is the number of classes. These classifiers take the Hand-Crafted or Self-Supervised features from a feature extractor  $f(\cdot) : \mathbb{R}^T \rightarrow \mathbb{R}^d$ . We compared the performance of Logistic Regression (LR) and Multi-Layer Perceptron (MLP) classifiers, which are common in the literature.

## Evaluation Metric

To evaluate the performance of the combinations of feature extractors and classifiers, we report results in terms of accuracy using a Leave-One-Speaker-Out (LOSO) cross-validation. Leave-One-Speaker-Out consists of training with  $n - 1$  speakers out of  $n$  and testing with the speaker not included in the training phase; this is akin to Leave-One-Out (LOO) cross-validation. We obtained the classifications per recording and aggregated the results per speaker when needed. For instance, an audio sample may be classified as dysarthric or not, but the final classification is made based on the most common class for a given speaker.

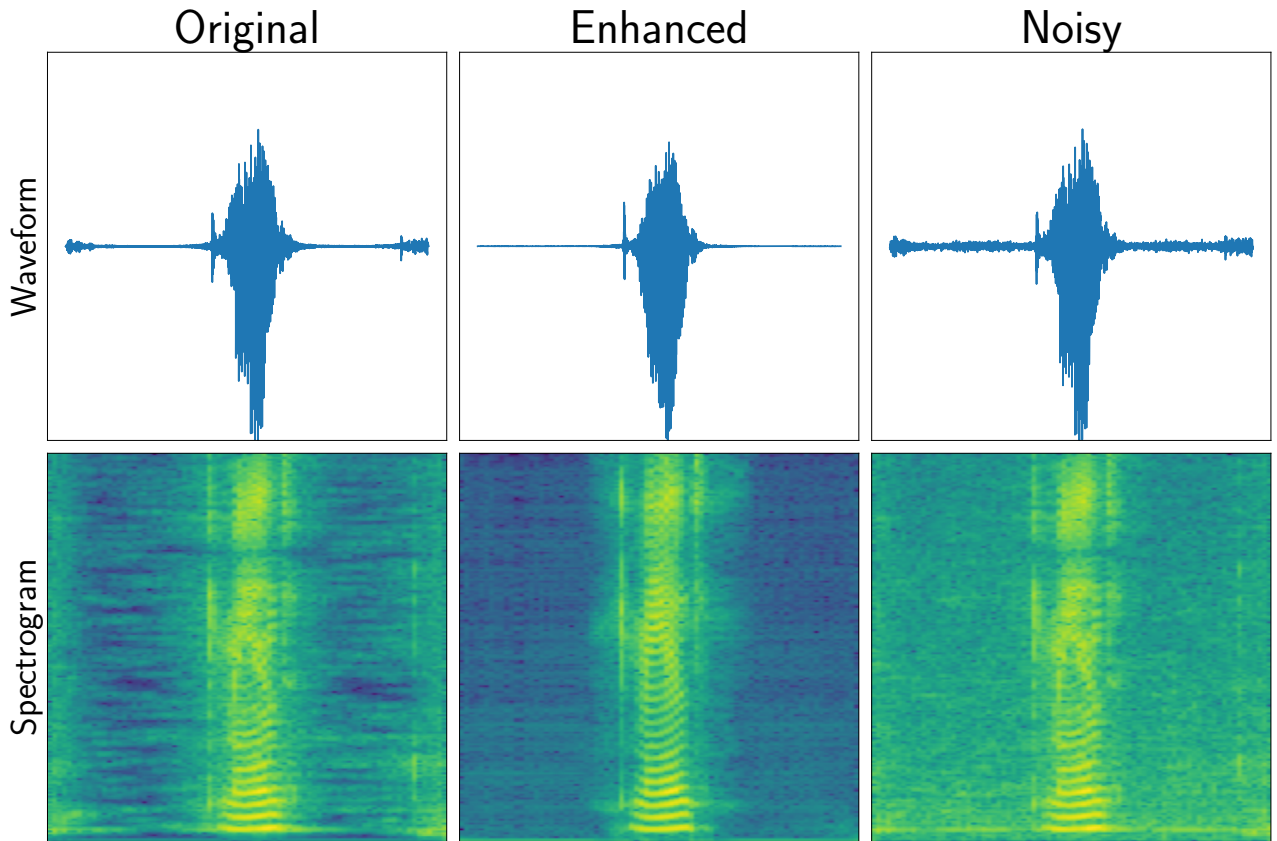


Figure 5.2: Audio Settings: Example of the same recording under different audio settings.

## 5.4 Experiments

To better understand the reliability of the assessment at the patient level, we developed a pipeline (Section 5.3.2) and visualize per-patient aggregated results for the Intelligibility classification (Figure 5.3), which allows for a detailed analysis of the predictions. This tool can be adapted to other feature extractors and classification models. Therefore, this pipeline could provide a more interpretable assessment for individual patients, leading to more personalized treatments.

### 5.4.1 Original: *How reliable are classifiers when confronted with unknown speakers?*

**Motivation:** Automated dysarthria assessment can help clinicians better understand and manage speech difficulties associated with dysarthria. Such assessment could lead to appro-

Table 5.1: Task Performance: results are reported using Accuracy, using the Leave-One-Speaker-Out cross-validation. For the Disease and Words classification problems, results are based on individual recordings. For the Intelligibility task, results are calculated performance at the speaker level.

| TASK            | EXTRACTOR    | DEFAULT            |                                           | ENHANCED           |                                           | NOISY              |                                           |
|-----------------|--------------|--------------------|-------------------------------------------|--------------------|-------------------------------------------|--------------------|-------------------------------------------|
|                 |              | ACCURACY %<br>LR   | ( $\mu_{(\pm\sigma)}$ $\uparrow$ )<br>MLP | ACCURACY %<br>LR   | ( $\mu_{(\pm\sigma)}$ $\uparrow$ )<br>MLP | ACCURACY %<br>LR   | ( $\mu_{(\pm\sigma)}$ $\uparrow$ )<br>MLP |
| DETECTION       | ACOUSTIC     | 69.3( $\pm 22.6$ ) | 65.8( $\pm 24.6$ )                        | 59.4( $\pm 23.8$ ) | 54.8( $\pm 33.7$ )                        | 79.4( $\pm 17.9$ ) | 76.6( $\pm 21.0$ )                        |
|                 | WAV2VEC2     | 94.2( $\pm 8.0$ )  | 94.1( $\pm 8.5$ )                         | 82.8( $\pm 16.4$ ) | 81.8( $\pm 17.9$ )                        | 99.8( $\pm 0.2$ )  | 99.7( $\pm 0.4$ )                         |
|                 | HUBERT       | 94.0( $\pm 8.0$ )  | 93.4( $\pm 9.8$ )                         | 84.8( $\pm 15.9$ ) | 85.5( $\pm 15.1$ )                        | 99.7( $\pm 0.5$ )  | 99.4( $\pm 1.3$ )                         |
|                 | MODIFIED CPC | 94.9( $\pm 7.3$ )  | 95.8( $\pm 6.7$ )                         | 82.0( $\pm 19.4$ ) | 84.1( $\pm 15.6$ )                        | 99.7( $\pm 0.5$ )  | 99.6( $\pm 0.9$ )                         |
| WORDS           | ACOUSTIC     | 9.6( $\pm 5.3$ )   | 9.6( $\pm 5.7$ )                          | 8.9( $\pm 5.2$ )   | 11.3( $\pm 6.4$ )                         | 9.2( $\pm 5.0$ )   | 9.6( $\pm 5.7$ )                          |
|                 | WAV2VEC2     | 56.5( $\pm 27.4$ ) | 56.5( $\pm 27.2$ )                        | 43.9( $\pm 23.8$ ) | 44.4( $\pm 24.0$ )                        | 54.0( $\pm 26.5$ ) | 54.5( $\pm 26.3$ )                        |
|                 | HUBERT       | 70.6( $\pm 29.1$ ) | 69.3( $\pm 29.0$ )                        | 56.6( $\pm 27.7$ ) | 55.8( $\pm 27.4$ )                        | 70.2( $\pm 29.0$ ) | 69.1( $\pm 29.1$ )                        |
|                 | MODIFIED CPC | 53.8( $\pm 29.4$ ) | 57.3( $\pm 29.1$ )                        | 46.1( $\pm 25.8$ ) | 48.1( $\pm 26.5$ )                        | 54.7( $\pm 30.2$ ) | 57.1( $\pm 29.5$ )                        |
| INTELLIGIBILITY | ACOUSTIC     | 45.5( $\pm 34.8$ ) | 45.1( $\pm 32.7$ )                        | 46.3( $\pm 41.8$ ) | 39.3( $\pm 36.8$ )                        | 53.5( $\pm 36.4$ ) | 53.2( $\pm 34.9$ )                        |
|                 | WAV2VEC2     | 63.4( $\pm 37.5$ ) | 63.2( $\pm 37.6$ )                        | 54.3( $\pm 35.5$ ) | 54.6( $\pm 36.0$ )                        | 67.6( $\pm 38.5$ ) | 67.4( $\pm 38.1$ )                        |
|                 | HUBERT       | 61.6( $\pm 39.1$ ) | 62.6( $\pm 37.9$ )                        | 55.4( $\pm 36.2$ ) | 55.5( $\pm 35.9$ )                        | 66.2( $\pm 40.2$ ) | 66.9( $\pm 39.5$ )                        |
|                 | MODIFIED CPC | 59.6( $\pm 41.1$ ) | 62.4( $\pm 39.9$ )                        | 52.3( $\pm 35.0$ ) | 54.1( $\pm 34.6$ )                        | 62.9( $\pm 41.9$ ) | 64.9( $\pm 41.0$ )                        |

priate personalized treatment plans to improve communication. While several studies have demonstrated promising results in automated dysarthria assessment with high accuracy, concerns about the generalizability of these models remain due to their evaluation methodology [90]. In particular, many models are trained and tested on the same speakers or evaluated with only one unseen speaker per class [2]. This limited evaluation approach can lead to overfitting, where models inadvertently learn speaker-specific traits rather than dysarthria-related features, thereby reducing their reliability in real-world clinical applications.

**Configuration:** We group recordings based on the patient ID, ensuring no overlap between the training and testing sets. Therefore, all the recordings from a speaker are either in the training or test set. We then perform a Leave-One-Speaker-Out (LOSO) cross-validation and assess the classifiers’ performance against speakers absent during training. We first extract features from the recordings using the feature extractors, hand-crafted features, and features from self-supervised models. Then, we evaluate the usability of these extracted features for the three tasks, *i.e.* detection, words, and intelligibility.

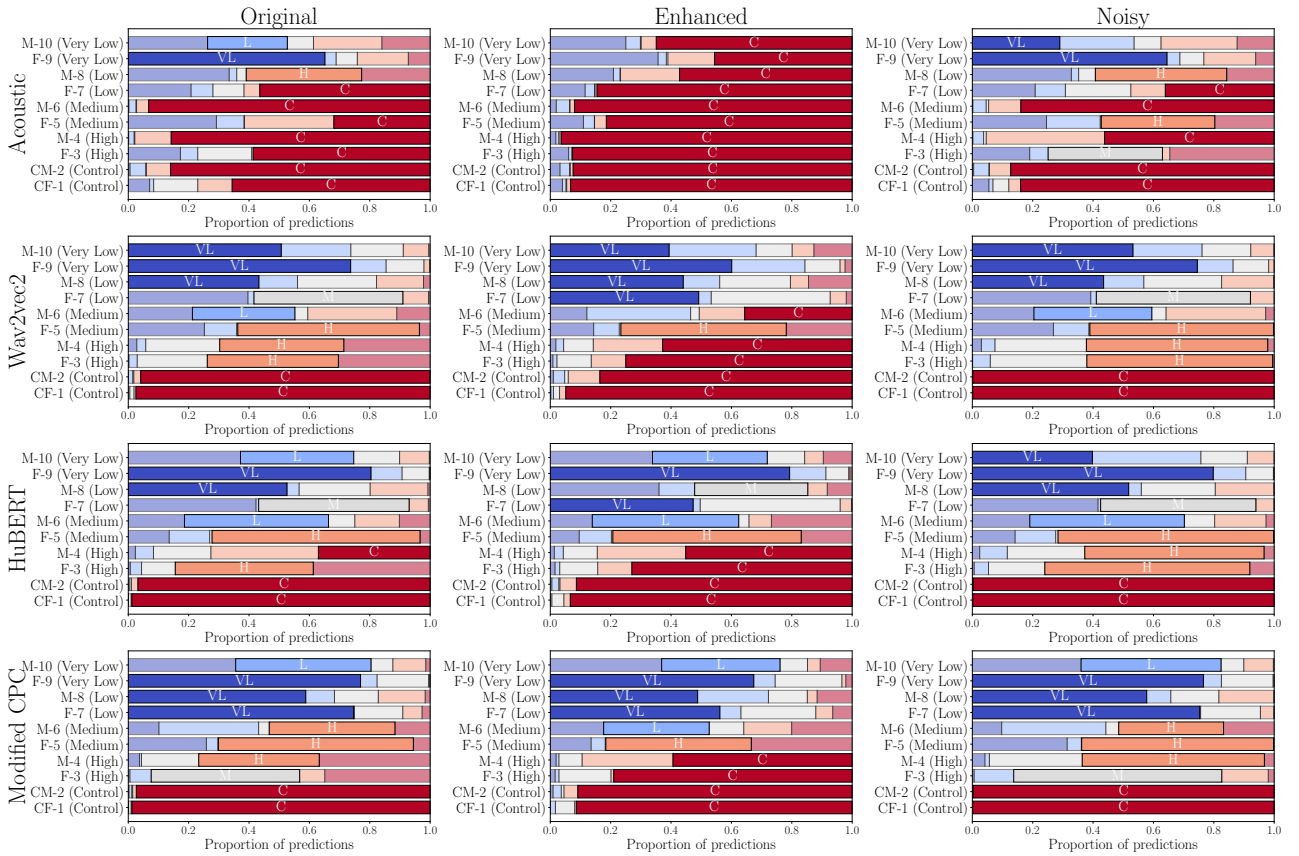


Figure 5.3: Speaker Intelligibility across datasets (columns) and features (rows). All predictions are made using logistic regression-based classifiers. While the performance based on HuBERT-extracted features is higher than acoustic, for a given speaker, there are major mis-classifications at the recording level.

**Self-Supervised Features outperform Hand-Crafted Features without model fine-tuning.** Huang et al., [102] highlights that most studies on automated dysarthria assessment have achieved high accuracy (ranging from 75.1% to 93.97%). Nonetheless, many of these studies have only tested their models on the same speakers used for training, as such results might not be generalizable to unseen speakers. Furthermore, there are often significant imbalances in the classes of speech tasks being assessed. Self-supervised features outperform hand-crafted features for all tasks (Table 5.1). Furthermore, the results are obtained without fine-tuning the self-supervised feature extractors, making them a promising direction for automated dysarthria assessment.

Now that we have evaluated the performance of the feature extractors on the original dataset, we aim to understand how robust they are when presented with corrupted input data.

### 5.4.2 Noisy Control: *Are models trained on the Original dataset biased by patterns related to the noise in recordings?*

**Motivation:** The original dataset has some recordings with different noise levels and clicking sounds. Thus, one could argue that the good performance observed on the detection task is due to patterns associated with external factors from the recording rather than speech-related information. To confirm the feature extractors’ and models’ sensitivity to such patterns, we exacerbate such effects by mixing audio samples.

**Configuration:** We generate  $\mathcal{D}_{\text{Detection}}^{\text{Noisy}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Noisy}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Noisy}}$  by mixing the original recordings from control individuals with noise patterns from WHAM dataset [182]. First, we obtained a single background noise sample from the WHAM dataset. Then, we mixed every audio recording from control patients with that noise pattern. In this scenario, feature extractors and classifiers that can extract these noise patterns should achieve higher performance, especially on the detection task.

**All feature extractors can capture noise patterns in control recordings that could lead to biased results.** All combinations of feature extractors and classifiers achieved higher accuracy on the detection tasks. (Table 5.1) Therefore, all feature extractors and models could leverage noise patterns repeated across control patients.

We now consider the impact of enhancing the recordings with VoiceFixer [127], which aims at addressing multiple distortions simultaneously.

### 5.4.3 Enhanced: *What impact does enhancing the recordings have over the different tasks?*

**Motivation:** Given the observation from the Noisy setting, we attempt to enhance the original recording using speech restoration methods. The speech restoration process involves restoring degraded speech signals to their original quality [79]. For instance, speech is typically surrounded by background noise, blurred by room reverberation, or recorded with low-quality

equipment. Ambient noise from clinical clicks or other artifacts may be present in Dysarthric recordings.

**Configuration:** We aimed to enhance the recordings by applying one of the speech enhancement approaches and evaluate the models’ performance in such scenarios. We generate  $\mathcal{D}_{\text{Detection}}^{\text{Enhanced}}$ ,  $\mathcal{D}_{\text{Words}}^{\text{Enhanced}}$ , and  $\mathcal{D}_{\text{Intelligibility}}^{\text{Enhanced}}$  by applying ‘VoiceFixer’ [127], a method that attempts to remove multiple distortions simultaneously. The pre-trained ‘VoiceFixer’ models produce outputs with 44.1 kHz sample rate. Therefore, we resample input to 16 kHz before extracting the representations. This ensures that the sampling rates match the input signal and the feature extractor models pre-trained on data with 16 kHz sample rate.

**The application of enhancement tools needs to be carefully considered.** The performance decreased compared to the default setting across all the tasks, feature extractors, and classifiers tested (Table 5.1). This suggests that the enhancement tools eliminate patterns previously present in the default settings. We investigate patient-level variations more deeply using our proposed visualization tool, which displays the model’s predictions for the speaker intelligibility task. After enhancement, the hand-crafted acoustic features lead to classification models being unable to predict the intelligibility levels, as they tend to predict every speaker with control-level intelligibility (Figure 5.3). Furthermore, the application of the enhancement tool led to information loss in some recordings, such as partial speech segment removal. This indicates that the systematic use of such enhancement tools requires additional care if they are used as a preprocessing step in automated assessment pipelines.

## 5.5 Discussion

### 5.5.1 Conclusion

#### Inspecting prediction at Patient-Level

We proposed a novel way to visualize predictions at a patient level. One can observe the different predictions made by the model for a given patient instead of focusing solely on a

summary statistic (Figure 5.3). We propose a tool to inspect the predictions at a patient level. Given a reliable classifier, one can inspect the different predictions for a given patient and determine whether the predictions indicate a mix of classes or if a class is overwhelmingly represented.

### 5.5.2 Limitations

We conducted our analyses on a single dataset, which may limit the generalizability of our results. We used readily available pre-trained model and evaluated their performance without modifying the features extractor.

#### **Classes are imbalanced.**

There is a major class imbalance in the number of recordings, *i.e.*, 8.3% (VL), 14% (L), 13.9% (M), 63.8% (H + C) or 19.6% (H) and 44.2% (C). Such imbalance can lead to the classifiers requiring further refining to distinguish the VL, L, and M classes.

#### **The labels are coarse with abrupt boundaries.**

The labels given to the intelligibility levels are coarse. For instance, patients with 26% intelligibility score are grouped with those with a 49% intelligibility score. Such coarse labels are challenging for the model to learn class boundaries.

#### **Intelligibility is a patient-level label.**

In addition to being coarse, the labels are given to the speaker. Nonetheless, each speaker might have different difficulties with all the words. As we associate each audio's intelligibility score with its speaker's intelligibility score, the decision boundaries might need to be better defined.

We showed that foundation models trained on healthy speakers can be transferred to dysarthric speakers with high accuracy. These feature extractors outperform handcrafted features.

### 5.5.3 Perspectives

Analyses on other datasets, such as the Nemours[132] dataset, could provide insights into the generalizability of the results. Furthermore, this work focused on the transferability of pre-trained extractors trained on healthy speakers without leveraging any patient-specific information. Patient-specific information could be used to further improve the performance of the models on the same dataset.

#### **Using pre-trained-extractors to determine per-sample intelligibility.**

Since foundation models have been trained on a large-scale dataset of speakers without dysarthria, one could use the audio recordings of the speakers without dysarthria in datasets such as UASpeech to define the reference intelligibility performance on any given word. Then, one could determine the per-audio word recognition, which in turn could provide a finer-grained intelligibility label of the audio recordings.

#### **Data Leakage on Medical Recordings.**

Depending on the dataset split operated, data leakage can bias results. Models trained and evaluated on patients with recordings in the training and evaluation set might perform well. Still, they could be less reliable as they have been partly trained on speaker-specific properties. While such a scenario can be useful in terms of personalized usage, for instance, for word recognition on a personal device, these splits can lead to an overconfident model that cannot generalize to unseen patients. Ensuring dataset splits such that patients are exclusively in one set or another allows for investigating performance in the future and, therefore, unknown at train time patients.

Now that we have shown that it is possible to transfer knowledge from healthy speakers to dysarthric speakers, we aim to understand these feature extractors' robustness when presented with corrupted input data.

# Chapter 6

## Assessing the Robustness of Self-Supervised Learning Representations to Incomplete Data

This chapter aims to understand how robust are Self-Supervised Learning (SSL) representations. We focus on the robustness of representations extracted by Self-Supervised feature extractors to corrupted data inputs. Regarding corruption, we consider masking parts of the input images. This work [22] is part of a joint effort. XC designed and implemented the evaluation pipeline, adapted the codes, ran the experiments, conducted the analyses, and wrote the manuscript.

### 6.1 Introduction

Supervised Learning has led to significant advances in the field of Computer Vision (CV) and Natural Language Processing (NLP). This is partially ascribed to the model's ability to learn relevant features from the data to solve the task for which they are trained. One limiting factor of Supervised Learning is the need for high-quality, labeled data. Such datasets can be challenging to construct due to collection and annotation costs and potential annotators' disagreement. Furthermore, labels are task-specific, while the input data can contain information that serves many purposes. For instance, while a picture of a cat can be labeled as a cat, it can also be

labeled as an animal, feline, or mammal. It can also separate background and foreground, detect objects, classify textures (hair, fur, and grass), and more. In recent years, Self-Supervised Learning (SSL) has gained popularity as it allows learning general-purpose representations by learning from signal from the data itself by exploiting its underlying structure [108, 35, 95, 8]. Furthermore, despite being trained on unlabeled data, SSL representations are competitive with Supervised Learning representations in many downstream [34, 11, 94, 142] and can achieve superior performance in many downstream vision tasks, including image classification, object detection, and semantic segmentation. SSL methods rely on the training via suitable pretext task [36], which by design are not targeted toward a specific supervised task, and such can lead to more generalized and usable representations.

In recent years, the robustness of representations in Supervised Learning has gained interest. In their work, Xiao et al., [187] have shown that by adversarially selecting the background of images, one could make the model misclassify with high accuracy. Following this observation, we asked *"How robust are SSL representations against information removal?"*.

## 6.2 Background

Supervised learning requires access to high-quality datasets with good labels. We assume that models trained in supervised learning are directly relevant to the task they are supervised on. Nonetheless, models trained in a supervised manner are sensitive to adversarial background changes [187], highlighting a reliance on the background. Self-Supervised Learning (SSL) has emerged as a key method for training general-purpose feature extractors, closing the gap with Supervised Learning methods [34, 95, 87]. Since the feature extractor is trained on unlabeled data, the representations it extracts are expected to be more general and transferable than models trained in a supervised manner.

### 6.2.1 Self-Supervised Learning (SSL): learning by leveraging signal from unlabeled data

SSL is a learning paradigm in which the model is trained on many unlabeled data using pretext tasks. As such, they can be trained without needing large amounts of labeled data. The

pretext tasks are commonly designed to leverage readily available information in the data, such as context and temporal information for textual or speech data, or predict the masked part of the input. Balestrieri et al. in [8] classify SSL methods into four broad categories, each emphasizing different principles of learning:

- Deep Metric Learning (*e.g.*, SimCLR [34]): methods that encourage similar input to be closer in the feature space.
- Self-Distillation (*e.g.*, DINOv2 [142]): methods that use the outputs of a model as pseudo-labels for training.
- Canonical Correlation Analysis (CCA) (*e.g.*, SwAV [28] and VICReg [11]): methods that aim at maximizing correlations between different views of the data.
- Masked Image Modeling (MIM) (*e.g.*, MAE [94]): methods that involve masking a part of the input and predicting the masked parts.

In the following, we draw inspiration from masked input modeling approaches, namely training the models to predict missing parts of the input, which has been extensively studied in Natural Language Processing (NLP).

### 6.2.2 Learning representation by masking parts of the input

In text or audio data, predicting purposely masked parts of the input has led to breakthroughs in the field of Natural Language Processing (NLP). In Computer Vision (CV), image inpainting and masked image modeling are two tasks that attempt to learn using masked parts of the input.

**Image Inpainting** Image inpainting aims at filling in missing or corrupted parts of an image. The models are commonly trained to generate content that blends with the surrounding areas.

While inpainting focuses on restoring images by filling gaps with appropriate content, Masked Image Modeling (MIM) focuses on learning representation from masked inputs.

**Masked Image Modeling (MIM)** MIM became a popular self-supervised visual pretraining method due to its impressive fine-tuning performance on a variety of downstream computer vision tasks [57, 32, 172]. The iGPT[32], ViT[58], and BEiT[10] research teams have recently re-explored this approach about modern vision transformers [30], and demonstrate a strong potential for representation learning by introducing specific designs for some components. These include clustering pixels, predicting mean color, and tokenizing using a block-wise masking strategy in addition to dVAE networks [188].

Since SSL methods are trained with pretext tasks, we need ways to evaluate their performance.

### 6.2.3 Evaluating SSL on downstream tasks

To evaluate the performance of SSL methods, linear probing is commonly used. Linear probing consists of training a linear classifier that takes advantage of the representations obtained by the feature extractor. In addition, it is possible to proceed to the visual evaluation [16]. Bordes et al. [16] suggest, for example, training a conditional generative diffusion model using a SSL representation. Thus, one can gain some insights into the representation by analyzing which information remains constant across different generated samples and which information does not remain constant.

In addition to these approaches, we are inspired by literature [120, 116] that uses representational similarity analysis to compare the robustness of supervised models. This work focuses on the robustness of the SSL models to corrupted data input, where masking parts of the input images corrupt data inputs.

### 6.2.4 Evaluating the similarity of representations

Recent research has attempted to explain the behavior of neural networks by comparing representations between layers and different trained models. We build on previous studies investigating the similarity of neural network representations [120] to evaluate the robustness of different self-supervised learning methods when learning from masked or missing data. In particular, we compare the learning performance of various SSL methods that belong to different

families using Canonical Correlation Analysis (CCA) [91, 136] and Centered Kernel Alignment (CKA) [116].

## Cosine Similarity (CS)

Cosine similarity is a metric used to measure the similarity between two vectors, commonly used in comparing neural network representations. Cosine similarity calculates the cosine of the angle between the vectors, with values ranging from -1 to 1. For neural networks, vectors can represent the activations or embeddings generated by the model.

Mathematically, the cosine similarity between two vectors  $\mathbf{x}$  and  $\mathbf{y}$  is defined as:

$$\text{cosine\_similarity}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

Where  $\mathbf{x} \cdot \mathbf{y}$  represents the dot product of the vectors, and  $\|\mathbf{x}\|$  and  $\|\mathbf{y}\|$  are the magnitudes of vectors  $\mathbf{x}$  and  $\mathbf{y}$ .

## Canonical Correlation Analysis (CCA) and Singular Value Canonical Correlation Analysis (SVCCA)

CCA is a Multivariate Statistics technique that identifies and measures the relationships between two sets of variables or finds patterns that represent the variation in the data. It constructs pairs of linear combinations, called canonical variables, from the original variables that are orthogonal within each set and are maximally correlated between sets. For Neural Networks, it compares representations by finding the linear relationships between two sets of activations. CCA projects the two sets of vectors into a shared subspace with a maximum correlation between them. This makes it particularly useful for understanding how different layers or networks encode information. SVCCA [147] is an extension of CCA that reduces dimensionality by using singular value decomposition (SVD) before applying CCA, improving the stability and interpretability of the comparison. Mathematically, CCA finds projections  $W_X$  and  $W_Y$  such that the correlation between  $W_X^T X$  and  $W_Y^T Y$  is maximized, where  $X$  and  $Y$  are the sets of activations from two networks:

$$\max_{W_X, W_Y} \text{corr}(W_X^T X, W_Y^T Y) \quad (6.1)$$

Here,  $W_X$  and  $W_Y$  are the projection matrices that maximize the correlation between the transformed representations of  $X$  and  $Y$ .

Given two sets of activations,  $X \in \mathbb{R}^{d_1 \times n}$  and  $Y \in \mathbb{R}^{d_2 \times n}$ , where  $d_1$  and  $d_2$  are the dimensions of the activations, and  $n$  is the number of samples, CCA aims to find projection matrices  $W_X \in \mathbb{R}^{d_1 \times k}$  and  $W_Y \in \mathbb{R}^{d_2 \times k}$  that project  $X$  and  $Y$  into a common subspace such that the correlation between the projections is maximized:

$$\max_{W_X, W_Y} \text{corr}(W_X^T X, W_Y^T Y) \quad (6.2)$$

This is achieved by solving the following optimization problem:

$$\max_{W_X, W_Y} \frac{\mathbb{E}[(W_X^T X)(W_Y^T Y)]}{\sqrt{\mathbb{E}[(W_X^T X)^2]\mathbb{E}[(W_Y^T Y)^2]}} \quad (6.3)$$

Singular Vector Canonical Correlation Analysis (SVCCA) builds upon CCA by first performing Singular Value Decomposition (SVD) on the input activations to reduce their dimensionality. This step helps to remove noise and focuses on the most significant directions of variance before applying CCA. Given the SVD of  $X$  and  $Y$ :

$$X = U_X \Sigma_X V_X^T, \quad Y = U_Y \Sigma_Y V_Y^T \quad (6.4)$$

SVCCA retains only the top singular vectors (those corresponding to the largest singular values), reducing  $X$  and  $Y$  to lower-dimensional representations. The CCA is then applied on these reduced matrices  $U_X$  and  $U_Y$ , leading to more stable and interpretable results.

## Centered Kernel Alignment (CKA)

Centered Kernel Alignment (CKA) can be used to compare the similarity between neural network representations, particularly across different layers or models. Unlike methods such

as CCA, which are based on linear projections, CKA uses kernel methods to capture more complex, non-linear relationships between the representations. CKA is invariant to orthogonal transformations and isotropic scaling, allowing comparison of different models trained under different initial conditions.

Given two sets of activations  $X \in \mathbb{R}^{n \times d_1}$  and  $Y \in \mathbb{R}^{n \times d_2}$ , where  $n$  is the number of samples and  $d_1, d_2$  are the dimensionalities of the activations, CKA uses the Gram matrices  $K_X$  and  $K_Y$  for  $X$  and  $Y$ , respectively. The Gram matrix for a set of activations  $X$  is computed as:

$$K_X = XX^T \quad (6.5)$$

To account for centering, the centered Gram matrix  $\tilde{K}_X$  is defined as:

$$\tilde{K}_X = K_X - \frac{1}{n}\mathbf{1}_n K_X - K_X \frac{1}{n}\mathbf{1}_n + \frac{1}{n^2}\mathbf{1}_n K_X \mathbf{1}_n \quad (6.6)$$

Where  $\mathbf{1}_n$  is an  $n \times n$  matrix of ones, ensuring that the resulting Gram matrix is centered.

The CKA similarity between  $X$  and  $Y$  is then defined as the normalized Hilbert-Schmidt Independence Criterion (HSIC)[\[86\]](#):

$$\text{CKA}(X, Y) = \frac{\text{HSIC}(X, Y)}{\sqrt{\text{HSIC}(X, X)\text{HSIC}(Y, Y)}} \quad (6.7)$$

where HSIC is the Hilbert-Schmidt Independence Criterion, measuring the similarity between two kernel matrices and defined as:

$$\text{HSIC}(X, Y) = \frac{1}{(n-1)^2} \text{Tr}(\tilde{K}_X \tilde{K}_Y) \quad (6.8)$$

Where  $\text{Tr}(\cdot)$  denotes the trace of the matrix, and the Gram matrices  $\tilde{K}_X$  and  $\tilde{K}_Y$  are the centered kernel matrices for  $X$  and  $Y$ .

## 6.3 Methodology

### 6.3.1 Problem Setup

Let  $\mathcal{D} = (\mathbf{x}_i, y_i)_{i \in [1, \dots, N]}$ , where  $N$  is the number of samples,  $\mathbf{x}_i \in \mathcal{I}$  are the input vectors, and  $y_i \in \mathcal{Y}$  are the labels. We denote by  $f : \mathcal{I} \rightarrow \mathcal{R}$  the feature extractor and by  $g : \mathcal{R} \rightarrow \mathcal{Y}$  the classifier. We denote the feature space with  $\mathcal{R}$ . We denote by  $m : \mathcal{I} \rightarrow \mathcal{I}$  the corruption function that we apply to the input data.

We aim to evaluate the robustness of the feature extractor  $f$  concerning the corruption function  $m$ . We denote  $\mathcal{D}_m$  the dataset obtained by applying the corruption function  $m$  to the dataset  $\mathcal{D}$ . We use the performance obtained using the classifier  $g$  on the modified version of the dataset  $\mathcal{D}_m$ .

We aim to evaluate the robustness of self-supervised representations when confronted with incomplete or corrupted data. We assume that the representations should adapt and perform satisfactorily in real-world scenarios where the data may be imperfect. To assess this, we measure the similarities between pairs of self-supervised representation vectors using Canonical Correlation Analysis (CCA) and Centered Kernel Alignment (CKA) across different dataset variations.

Let two representation vectors be denoted as  $\mathbf{x}, \mathbf{y}$  such that  $\mathbf{x} = [x_1, x_2, \dots, x_n]$  and  $\mathbf{y} = [y_1, y_2, \dots, y_n]$ , with  $n$  representing their dimensionality. We compute similarity metrics to quantify these vectors' alignment and linear relationships. This approach provides valuable insights into how well self-supervised representations align under varying conditions.

### 6.3.2 Removing information from images

#### Removing foreground and background parts of images

To simulate information removal, we applied segmentation masks to the images, selectively removing parts of them. This helps us evaluate self-supervised representations' robustness when key information is missing. Segmentation masks were generated using the Segment Anything

Model (SAM) [113]. For each image, we segmented objects of interest based on class labels. For instance, given an image labeled 'Apple,' we extracted the segmentation mask corresponding to the apple. This process yields two variations of the image: the **Foreground** (containing only the apple) and the **Background** (containing only the background). Notably, the **Foreground** and **Background** images do not share any pixels.

## Removing the center and border of images

We also experimented with a different form of information removal using circular masks. These masks were centered on the image and designed to cover an equal distance in all directions from the center. From these circular masks, we generated two additional image variations: the **Center** (which retains only the central part of the image) and the **Border** (which retains the outer parts). As with the segmentation masks, the **Center** and **Border** images do not share any pixels. We show examples of the different image variations in Figure 6.1.

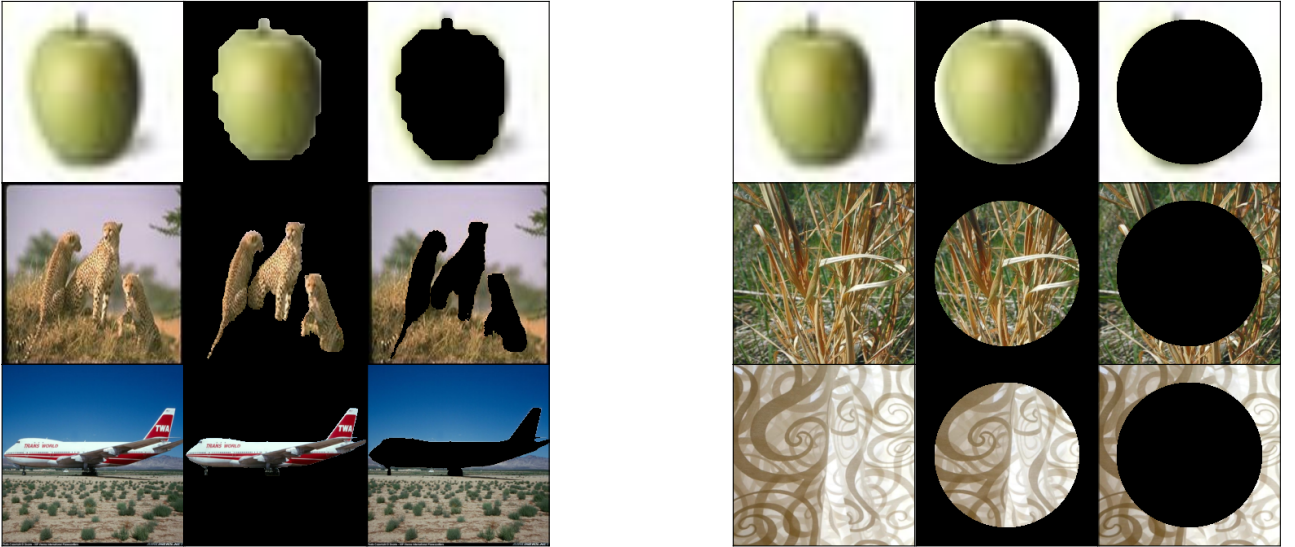


Figure 6.1: Examples of the 5 variations explored. On the left we show the **Original**, **Foreground**, and **Background** from top to bottom, on CIFAR-100, Caltech101, FGVC Aircraft. On the right we show the **Original**, **Center**, **Border** from top to bottom, CIFAR-100, DTD and DTD

### 6.3.3 Variations of the dataset

Given the information removal methods we consider, we obtain five versions of the dataset:

- **Original:** The original image without removal of information. This is typically the image used to train the models.
- **Foreground:** The images with only the foreground. That is, given a picture of an apple, we keep only the apple. This version of the image contains only noncontextual information about the object to classify.
- **Background:** The image with only the background. That is, given a picture of an apple, we keep only the background. This version of the image might contain contextual information about the object to classify but not directly about the specific object.
- **Center:** The images with only the center. That is, we keep only a circular region of the image around its center. This version of the image contains information only about the center of the image, which commonly contains the object to classify for a dataset that is object-centric, such as ImageNet.
- **Border:** The images with only the border. That is, we keep only a circular region of the image around its border. This version of the image contains information only about the border, which commonly includes information on the background and the object’s context.

#### 6.3.4 Linear Evaluation: How well can we classify using the extracted representations?

A common evaluation strategy for self-supervised feature extractors is to freeze the extractor and then extract its representations. Then, one can use a linear model for classification using these extracted representations. Based on the linear model’s performance across multiple datasets, we can assess whether the representations are generalized and suitable for downstream tasks. Namely, assume we have a feature extractor  $f$  and a linear model  $g$ . Given pairs  $(\mathbf{x}_i, y_i)_{i \in [1, \dots, N]}$ , we extract the representations  $\mathbf{z}_i = f(\mathbf{x}_i)$ . Then, we use the extracted representations to train a linear model  $g$  using labels  $y_i$ . We can then assess the performance of the linear model  $g$  on a separate dataset  $\mathcal{D}_{\text{test}}$  to determine the generalizability of the feature extractor  $f$ .

### 6.3.5 Representation Similarity: Are variations of the same image leading to similar representations?

To evaluate the similarity of the representations, we consider three metrics: cosine similarity, Canonical Correlation Analysis (CCA), and Centered Kernel Alignment (CKA). We obtain the extracted representations of the different variations of the dataset and then compute the similarity metrics between a point and its variations. Let  $\mathbf{z}_i$  be the representation of the original image and  $\mathbf{z}_{i,v}$  be the representation of the variation  $v$  of the image. Since we use the same feature extractor for all the data set variations, we can compare the representations directly, as these representations should all lie in the same feature space. Thus, we can make direct comparisons between  $\mathbf{z}_i$  and  $\mathbf{z}_{i,v}$  for a given variation  $v$  and the data point  $i$ .

### 6.3.6 Representation Neighborhood: Are samples in the representations neighboring samples from the same variants?

Given all the representations extracted from the different variations of the dataset, we can use the feature extractor to obtain the neighbors for a given representation. Let  $\mathcal{X}_v$  be the set of representations for a given variation  $v$ . For example,  $\mathcal{X}_{\text{Original}}$  is the set of representations for the original images. Let  $\mathcal{X}_{\text{All}}$  be the set of all representations.  $\mathcal{X}_{\text{All}} = \bigcup_{v \in \mathcal{V}} \mathcal{X}_v$ , where  $\mathcal{V} = \{\text{Original, Foreground, Background, Center, Border}\}$  is the set of all variations. We use K-Nearest Neighbors (K-NN) to find the  $k$ -Nearest Neighbors for a given representation. This allows us to observe whether the image representations are close to the image representations of the same variation.

## 6.4 Experiments

We use linear probing to evaluate the different representations produced by the models (Section 6.4.1). We performed these analyses over multiple datasets and their variants. We compared the representations using Canonical Correlation Analysis (CCA) and Centered Kernel Alignment (CKA) (Section 6.4.2 and 6.4.3). Finally, we discuss the proximity of representations based on their nearest neighbors (Section 6.4.4).

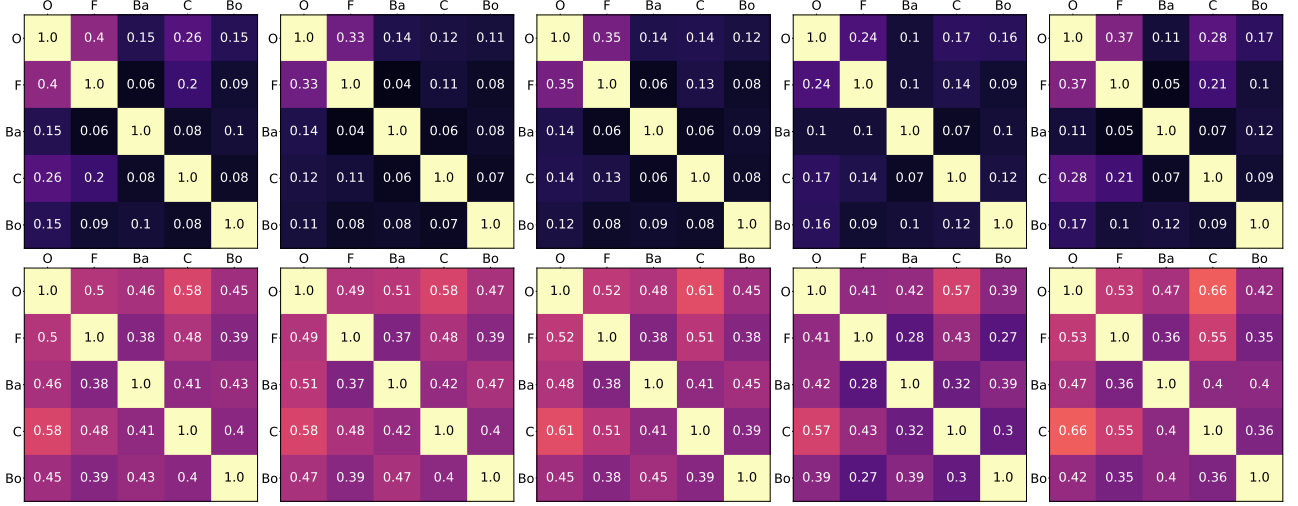


Figure 6.2: Mean Squared CCA correlation  $\mathbb{R}_{CCA}^2$ . The first row is based on CIFAR-100, the second row on Caltech101. From left to right, the matrices are associated with the representation extracted by the Supervised SwAV, VICReg, MAE, and DINOv2 models. **O**riginal, **F**oreground, **Ba**ckground, **C**enter, **Bo**rders.

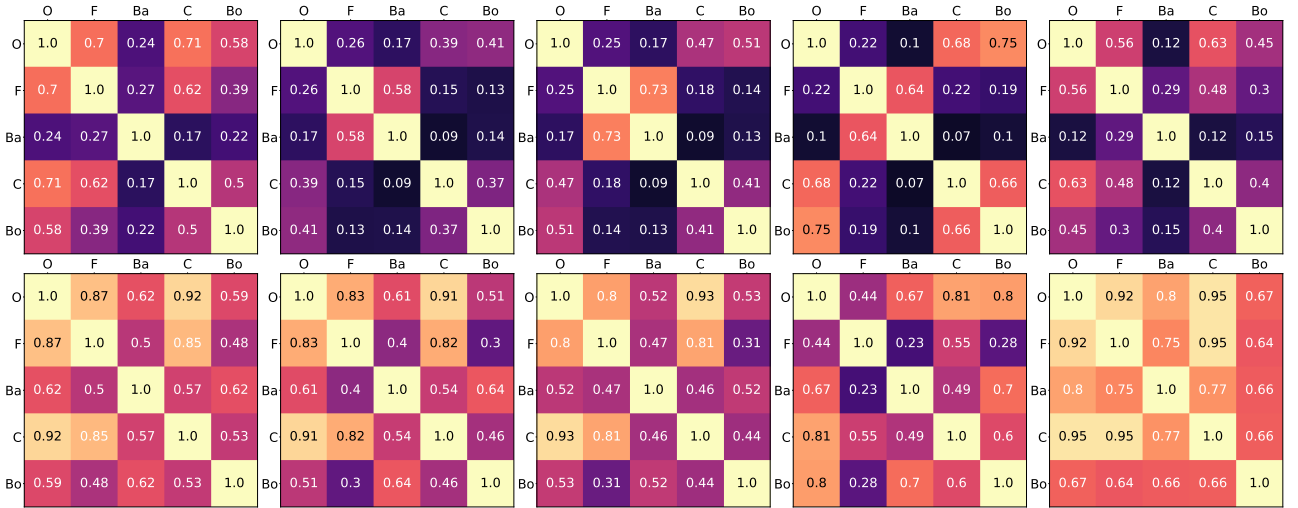


Figure 6.3: CKA scores. The first row is based on CIFAR-100, the second row on Caltech101. The matrices are associated with Supervised, SwAVm, VICReg, MAE, and DINOv2 from left to right. **O**riginal, **F**oreground, **Ba**ckground, **C**enter, **Bo**rders.

#### 6.4.1 The representations contain information about the context.

We performed linear probing on CIFAR-100, Caltech-101, FGVC-Aircraft, and DTD datasets using their five variations: Original, Foreground, Background, Center, and Border (Table 6.1). All the models have higher classification accuracy when using the Original representations

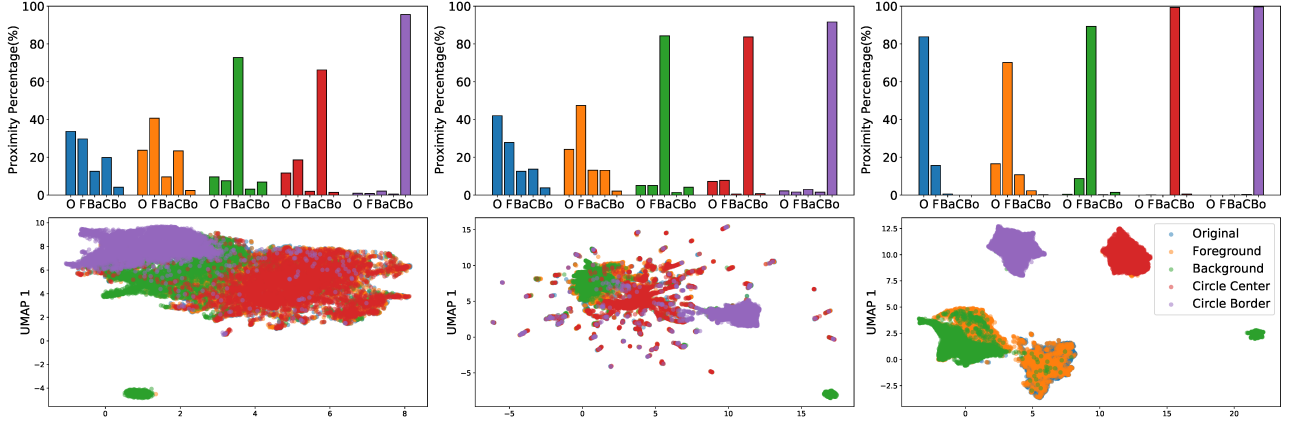


Figure 6.4: The first row indicates how each variant neighboring points are related to other variants. These observations are based on the representations extracted on CIFAR-100. We consider the Supervised, DINOv2, and MAE models from left to right. The second row depicts the UMAP obtained on 10% of the representations.

rather than the Foreground ones on CIFAR-100 and Caltech-101. This indicates that the feature extractor captures contextual information that the linear classifiers use. DINOv2 performs better when using the Center variant than the Foreground variant. For the Foreground variant, the entirety of the object is not guaranteed to be located in the center of the image, even though most common datasets locate the object in the center of the picture. For example, in the Caltech-101 dataset, the images with human faces as targets are not necessarily centered, and some are outside the Center variant. Nonetheless, all the models except MAE reach higher balanced accuracy on the Original image than on the Foreground, and the Center variant leads to higher performance than the Foreground variant. That could indicate that the Self-Supervised models tend to learn additional contextual information from the background. When evaluated on only the Background variations, all models except the MAE model achieve balanced accuracy above 70% on Caltech-101, which could indicate a lack of diversity in terms of background.

### **6.4.2 Representations between Original and Center are more similar in Caltech-101 than Original and Foreground based on Canonical Correlation Analysis (CCA).**

To further analyze the representations, we use the Canonical Correlation Analysis (CCA) to compare the similarity between the different variants. We present the CCA matrices in figure 6.2 for CIFAR-100 and Caltech-101. CIFAR-100 shows that across all models, the representations of the Original images closely align with the Foreground variants, with similarity scores ranging from 0.24 for the MAE model to 0.4 for the Supervised model. This observation aligns with results from the Linear Evaluations. Moreover, for both the Supervised and DINOv2 models, the Original and Center dataset variants exhibit the highest similarity, followed by the Center to Foreground representations. In the Caltech101 dataset, the representations of the Original and Center images are most alike across all models. This may be attributed to the relative size of the objects of interest in these datasets compared to the overall image dimensions. The high similarity levels between Center and Foreground and Border and Background corroborate this assertion.

We note that DINOv2 exhibits learning patterns similar to those of the Supervised model when contrasted with other Self-supervised models. However, the balanced accuracies derived from DINOv2 representations surpass those from the Supervised Model.

Our results underscore the need for additional research to determine whether the emphasis should be on devising new pretext tasks or innovating preprocessing steps to curate datasets.

### **6.4.3 Foreground and Background are more similar than Original and Foreground for SwAV, VICReg, and MAE based on Centered Kernel Alignment (CKA).**

We further compare the representations using the Centered Kernel Alignment (CKA) (Figure 6.3). The Supervised and DINOv2 models have higher similarity scores between the Original and Center variants, followed by the Original and Foreground. On CIFAR-100, all the models have lower similarity between the Center and Background representations, which one can expect

since the objects are located in the center of the images. Nonetheless, for SWaV, VICReg, and MAE, the Foreground and Background representations are more similar than the Original and Foreground representations. Furthermore, the MAE model also shows low similarity between the Background and Original and the Border and Background. On the Caltech101 dataset, the DINOv2 representations are primarily similar across all variants, with scores above 0.64. Furthermore, while the CCA indicated that the Center and Background representations were not that similar, the CKA indicated that for DINOv2, the Background and Center representations are highly similar, with a score of 0.77. These corroborate the linear evaluation performance observed, with a high balance using DINOv2’s representations only in background images.

#### **6.4.4 Representations from different variants are close to each other.**

We further visualize the representation spaces using UMAP and neighborhood proximity analysis using K-Nearest Neighbors (Figure 6.4). Through the K-Nearest Neighbors analysis, we observe that the Supervised and DINOv2 Models lead to Original representations that are close to a mix of others. Indeed, only 40% of the Original and Foreground samples have variant-matching neighbors. Meanwhile, in the MAE representation space, the different variants mostly match their neighbors. Indeed, all variants have more than 80% of their closest neighbors belonging to the similar variant. Nonetheless, MAE separates the variants from the Center and Border representations for the other variants, while the Original, Foreground, and Background overlap slightly.

## **6.5 Discussion**

### **6.5.1 Conclusion**

Understanding the robustness of self-supervised representations is crucial to assessing their practicality and reliability in real-world applications. By investigating the performance of these representations under varying degrees of data corruption, we can identify potential limitations and design strategies to enhance their robustness. We studied the effect of Foreground / Background removal and input masking on the representations extracted by Self-Supervised models.

We analyzed how masked input impacted Self-Supervised models compared to supervised models and showed that the DINOv2 model leads to representation similarity that matches those of the reference supervised model. Despite these similarities, the learned representation leads to higher balanced accuracy scores.

### 6.5.2 Limitations

In this work, we only considered segmentation and masking as means of missing information in visual data. While the Foreground and Background masks are the most natural choice, they do not account for potential size differences between the classes.

### 6.5.3 Perspectives

While we focus on visual information in natural images, one could extend this work to other modalities such as audio, text, and time series data. Furthermore, such analysis could be beneficial for foundation models for medical imaging. Additionally, one could explore one of the following directions:

- Analyzing the distribution of portions of the foreground concerning the total image size and its effect on the classification performance.
- Developing a framework to automatically identify biases in the data and across classes within the dataset to improve the robustness of the models trained on these datasets.

Table 6.1: Linear Probing: results are 5-fold cross-validated Balanced Accuracy. DINOv2 is consistently the best performing model. Except of Caltech-101, the supervised performs worse than the self-supervised extractors. On every dataset, the **Background** representations can be used to train classifiers with much higher than random performance. FGVC-Aircraft is the only dataset where the **Foreground** performance is much higher than the **Original** images.

| DATASET    | CIFAR-100                   |                             |                             |                             |                             |
|------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| VARIATION  | ORIGINAL                    | FOREGROUND                  | BACKGROUND                  | CENTER                      | BORDER                      |
| SUPERVISED | 58.59( $\pm 0.48$ )         | 54.27( $\pm 0.57$ )         | 27.56( $\pm 0.81$ )         | 51.74( $\pm 0.39$ )         | 33.22( $\pm 0.15$ )         |
| SWAV       | 65.02( $\pm 0.43$ )         | 56.79( $\pm 0.47$ )         | 29.53( $\pm 0.45$ )         | 53.94( $\pm 0.49$ )         | 33.46( $\pm 0.54$ )         |
| VICREG     | 62.01( $\pm 0.54$ )         | 57.05( $\pm 0.50$ )         | 27.83( $\pm 0.50$ )         | 54.05( $\pm 0.44$ )         | 32.97( $\pm 0.29$ )         |
| MAE        | 45.55( $\pm 0.80$ )         | 36.08( $\pm 0.81$ )         | 18.85( $\pm 0.32$ )         | 32.46( $\pm 1.46$ )         | 21.98( $\pm 0.49$ )         |
| DINOv2     | <b>85.50</b> ( $\pm 0.31$ ) | <b>74.19</b> ( $\pm 0.35$ ) | <b>30.82</b> ( $\pm 0.67$ ) | <b>77.17</b> ( $\pm 0.19$ ) | <b>43.83</b> ( $\pm 0.57$ ) |
| DATASET    | CALTECH-101                 |                             |                             |                             |                             |
| VARIATION  | ORIGINAL                    | FOREGROUND                  | BACKGROUND                  | CENTER                      | BORDER                      |
| SUPERVISED | 93.28( $\pm 0.63$ )         | 88.87( $\pm 1.21$ )         | 74.83( $\pm 0.62$ )         | 89.78( $\pm 0.42$ )         | 62.32( $\pm 1.19$ )         |
| SWAV       | 90.56( $\pm 0.66$ )         | 86.98( $\pm 1.65$ )         | 72.35( $\pm 1.10$ )         | 87.68( $\pm 0.72$ )         | 55.18( $\pm 1.58$ )         |
| VICREG     | 91.92( $\pm 0.67$ )         | 87.95( $\pm 0.99$ )         | 73.06( $\pm 1.46$ )         | 88.84( $\pm 0.86$ )         | 55.52( $\pm 1.94$ )         |
| MAE        | 69.03( $\pm 3.24$ )         | 71.49( $\pm 1.73$ )         | 37.24( $\pm 2.24$ )         | 69.35( $\pm 1.89$ )         | 25.10( $\pm 2.47$ )         |
| DINOv2     | <b>97.34</b> ( $\pm 0.57$ ) | <b>95.86</b> ( $\pm 0.77$ ) | <b>89.30</b> ( $\pm 1.45$ ) | <b>97.52</b> ( $\pm 0.41$ ) | <b>82.82</b> ( $\pm 0.85$ ) |
| DATASET    | FGVC-AIRCRAFT               |                             |                             |                             |                             |
| VARIATION  | ORIGINAL                    | FOREGROUND                  | BACKGROUND                  | CENTER                      | BORDER                      |
| SUPERVISED | 42.47( $\pm 1.64$ )         | 48.79( $\pm 2.21$ )         | 36.01( $\pm 2.10$ )         | 38.25( $\pm 2.12$ )         | 27.37( $\pm 0.96$ )         |
| SWAV       | 50.51( $\pm 1.53$ )         | 64.98( $\pm 1.88$ )         | 39.98( $\pm 0.77$ )         | 49.39( $\pm 1.64$ )         | 23.56( $\pm 1.22$ )         |
| VICREG     | 58.70( $\pm 1.02$ )         | 68.69( $\pm 1.62$ )         | 46.03( $\pm 0.31$ )         | 57.32( $\pm 1.58$ )         | 25.14( $\pm 0.78$ )         |
| MAE        | 29.98( $\pm 1.46$ )         | 42.28( $\pm 1.57$ )         | 15.01( $\pm 1.56$ )         | 31.99( $\pm 1.10$ )         | 10.04( $\pm 0.98$ )         |
| DINOv2     | <b>84.07</b> ( $\pm 0.70$ ) | <b>82.03</b> ( $\pm 0.83$ ) | <b>75.72</b> ( $\pm 1.46$ ) | <b>83.10</b> ( $\pm 1.01$ ) | <b>58.75</b> ( $\pm 1.14$ ) |
| DATASET    | DTD                         |                             |                             |                             |                             |
| VARIATION  | ORIGINAL                    | FOREGROUND                  | BACKGROUND                  | CENTER                      | BORDER                      |
| SUPERVISED | 71.30( $\pm 2.26$ )         | —                           | —                           | 68.48( $\pm 1.76$ )         | 67.23( $\pm 1.21$ )         |
| SWAV       | 75.85( $\pm 1.65$ )         | —                           | —                           | 72.39( $\pm 2.02$ )         | 70.43( $\pm 1.80$ )         |
| VICREG     | 73.86( $\pm 1.42$ )         | —                           | —                           | 70.53( $\pm 1.40$ )         | 66.94( $\pm 2.01$ )         |
| MAE        | 68.83( $\pm 1.86$ )         | —                           | —                           | 63.22( $\pm 0.85$ )         | 61.20( $\pm 1.92$ )         |
| DINOv2     | <b>82.13</b> ( $\pm 1.10$ ) | —                           | —                           | <b>82.47</b> ( $\pm 1.11$ ) | <b>81.25</b> ( $\pm 0.59$ ) |

# Chapter 7

## Learning to Unlearn

The following chapter is the result of the joint work of [23]. XC designed and implemented the evaluation pipeline, adapted the codes, ran the experiments, conducted the analyses, and wrote the manuscript.

### 7.1 Introduction

Machine unlearning (MU) aims to remove the influence of a specified subset of training data points from trained models [24]. This process is crucial for enhancing privacy preservation, model safety, and overall model quality. MU helps ensure compliance with the right to be forgotten [78], removes erroneous data points that negatively impact model performance [155], and eliminates biases introduced by parts of the training data [33]. Deep neural networks present significant challenges for MU due to their computationally intensive training requirements, highly non-convex loss landscapes, and tendency to memorize substantial portions of their training data [71, 27, 68]. The key open challenges in MU include:

- A degradation in model accuracy often accompanies unlearning;
- Some MU methods require the model to be trained in specific ways, such as saving checkpoints, tracking accumulated gradients, or training with differential privacy, limiting their applicability to already deployed models;

- Assurance of information removal is difficult as there are no reliable metrics to measure it accurately;
- There is no consensus on which methods are the most effective.

A branch of MU known as *exact unlearning* aims to guarantee that the specified *forget* data have been completely removed from the model. The most reliable exact MU method is to *retrain* the model from scratch while excluding the forget data. Another exact MU that offers data removal guarantees is SISA (Sharded, Isolated, Sliced, Aggregated) [19]. However, exact MU is computationally prohibitive, emphasizing the need for more efficient methods. The alternative branch is *approximate unlearning* that aims to approximate data deletion and is often less precise but more computationally efficient. Approximate MU lacks theoretical guarantees, necessitating empirical evaluations to determine its effectiveness, reliability, and computational efficiency across various datasets. Thus, the community needs to properly benchmark and evaluate state-of-the-art approximate MU methods.

In this paper, we benchmark 18 state-of-the-art approximate MU methods across 5 datasets and 2 DNN architectures commonly used in computer vision: ResNet18 [96] and TinyViT [183]. The 18 methods consist of 3 classical baseline methods Fine-tuning (FT), Gradient Ascent (GA), Successive Random Labels (SRL); 7 high-ranking methods in the NeurIPS’2023 Machine Unlearning competition [171], we name them Forget-Contrast-Strengthen (FCS), Masked-Small-Gradients (MSG), Confuse-Finetune-Weaken (CFW), Prune-Reinitialize-Match-Quantize (PRMQ), Convolution-Transpose (CT), Knowledge-Distillation-Entropy (KDE) and Repeated-Noise-Injection (RNI); and 8 recently published methods: Saliency Unlearning (SalUN) [66], Catastrophic Forgetting-K (CF-k) [80], Exact Unlearning-K (EU-k) [80], Scalable Remembering and Unlearning unBound (SCRUB) [119], Bad Teacher (BT) [39], Fisher Forgetting (FF) [82], Influence Unlearning (IU) [115, 105] and Negative Gradient Plus (NG+) [119]. We evaluate the different unlearning methods in terms of four major aspects: privacy evaluation, performance retention, computational efficiency, and unlearning reliability.

The **contributions** of our study is to address the following research questions:

*Q1. Are the commonly used MU baselines reliable?* Most of the recently introduced MU methods are compared only with three baselines: Fine-tuning (FT), Gradient Ascent (FA), and Successive Random Labels (SRL), and not across recently introduced approaches. We

show that with proper hyperparameter selection, FT is a reliable baseline. In contrast, GA consistently performs poorly and should be replaced by the more recent Negative Gradient Plus (NG+) [119] that simultaneously reduces the performance in the forget set while maintaining the performance on the rest of the data points.

*Q2. How reliable are MU methods across datasets, models, and initialization?* Our findings show that, unlike most recent MU methods, Masked-Small-Gradients is consistently among the best-performing methods across various metrics. In contrast, methods such as SRL and NG+ do not consistently outperform others across different datasets.

*Q3. Among existing methods, which are the most reliable and accurate?* Among 18 methods, our evaluation shows that specific methods, such as Masked-Small-Gradients (MSG), Convolution-Transpose (CT), and Fine-tuning (FT), exhibit desirable properties. Specifically, MSG and CT show resilience against U-LiRA, a strong per-sample membership inference attack. Furthermore, all three methods show consistency across datasets, and these methods could serve as reliable baselines for future studies.

## 7.2 Background

### 7.2.1 Machine Unlearning

Machine Unlearning is often first associated with the work of Cao et al. [25], followed by Bourtoule et al. [19], which proposes SISA (Sharded, Isolated, Sliced, Aggregated) as an exact unlearning method. For a recent overview of MU, we refer to the survey by Xu et al. [189], which provides a taxonomy of standard unlearning methods. Furthermore, Zhang et al. [197] review MU through privacy-preserving and security lenses. The authors cover the Confidentiality, Integrity, and Availability security triad and the need for Data Lineage, which relates to following the movement of data in a machine learning pipeline and understanding from where it originates, where it is stored, and how it percolates in the system through transformation. Some might have information on common MU verification methods, privacy evaluation metrics, and datasets; we refer to the works of Nguyen et al. [139] and Shaik et al. [155].

In the following, we focus on the MU taxonomy of Xu et al. [189], which considers Data

Reorganization and Model Manipulation:

(1) **Data Reorganization** methods focus on directly modifying the data to perform unlearning. It is divided into Data Obfuscation, Data Pruning, and Data Replacement.

*Data Obfuscation* refers to modifying the dataset to obscure the influence of the data to be unlearned: random relabeling and retraining [85], SRL (Successive Random Labels) and Saliency Unlearning (SalUN).

*Data Pruning* usually relies on dividing the dataset into multiple sub-datasets and training sub-models on these subsets. This is the category to which SISA [19] relates. Our work does not consider methods associated with this setting, as they make assumptions about the training process.

*Data Replacement* attempts to unlearn by replacing the original dataset with transformed data that simplifies unlearning specific samples. For example, Cao et al. [25] replace training data with summations of efficiently computable transformations. Like data pruning, these methods tend to make strong assumptions about the training process.

(2) **Model Manipulation** methods directly adjust the model parameters to remove the influence of specific data points. Model manipulation is divided into Model Shifting, Model Replacement, and Model Pruning.

*Model Shifting* directly updates the model parameters to offset the influence of the unlearned samples, such as using a single step of Newton’s method on model parameters [89] or decremental updates [152]. In our benchmark Fisher Forgetting (FF), Influence Unlearning (IU), and SalUN would represent these approaches.

*Model Replacement* uses precalculated parameters that do not reflect the data to forget to replace parts of the trained model. For instance, when using decision trees, one can replace nodes affected by the forget set by precalculated node [153]. These methods often make strong assumptions about the training process and the overall model.

*Model Pruning* prunes specific parameters from the trained models to remove the influence of certain samples [106], or Prune-Reinitialize-Match-Quantize (PRMQ) [44] which prunes the model via L1 pruning, reinitializes parts of the model then train the model on  $\mathcal{D}_R$ .

### 7.2.2 Machine Unlearning for Deep Neural Networks

Initially, MU research focused primarily on linear models such as linear regression and logistic models. Such models allow for the design of methods that assume the convexity of the loss function, rendering them less practical for DNN-based approaches. Since DNNs can memorize parts of their training data, they are particularly relevant targets for MU, even more so when trained on large amounts of potentially personal data. For Deep Learning models, unlearning raises additional challenges: 1) the non-convexity of the loss function of Deep Neural Networks [37], 2) the size of the models inducing high computational costs, 3) the randomness coming from the model’s training process, such as the initialization seed, randomness in the minibatch generation process, and 4) the fact that any model update impacts subsequent versions of the models, namely the weights at epoch  $n + 1$  directly depend on the weights at update  $n$ .

When considering MU for DNN, Xu et al. [189] notes that a standard scheme for DNN is to focus only on the final layer, as this layer is expected to be the most relevant for the downstream task and stems from the early works MU. However, Goel et al. [80] showed that simply modifying the final layer is often insufficient to remove information related to  $\mathcal{D}_f$ . However, other approaches, such as those of Golatkar et al. [82, 83, 81], attempt to unlearn the full model via methods derived from Information Theory. For example, weight scrubbing on trained models can be done by approximating the Fisher information matrix.

### 7.2.3 Post-Hoc Machine Unlearning

While proactively designing deep-learning pipelines with built-in unlearning methods such as SISA can greatly simplify the unlearning process, many contemporary services relying on DNN were not implemented with unlearning in mind. This motivates searching for methods that can unlearn from models already trained without making assumptions about the training process.

Thus, we focus on *posthoc MU*, a scenario where we assume that the unlearning method is agnostic to the original model training process. In such a scenario, there are differences in data availability at the time of unlearning. For example, whether one has access to the original training data  $\mathcal{D}$ , the retain set  $\mathcal{D}_R$ , the forget set  $\mathcal{D}_F$ , or even some external sets such as the validation set  $\mathcal{D}_V$ . Therefore, careful consideration should be given to the data requirement

associated with an unlearning method. Some might require having access to both  $\mathcal{D}_R$  and  $\mathcal{D}_F$  at the unlearning time, while others assume that  $\mathcal{D}_R$  is no longer available [40] making them more practical in real-world scenarios. Throughout our benchmark, we make the same assumption as the NeurIPS2023 Unlearning Challenge [171], where the unlearning methods had access to  $f_O, \mathcal{D}_R, \mathcal{D}_F, \mathcal{D}_V$ .

#### 7.2.4 Machine Unlearning and Differential Privacy

We based our Unlearning definition on Sekhari et al. [154] and refer to their work on the distinction between Differential Privacy and the objective of Machine Unlearning. In a high-level picture, differential privacy is a method for publicly sharing aggregated information about a population by describing the patterns discovered among the groups within the dataset while withholding specific information about individual data points. A randomized algorithm  $\mathcal{A}$  is  $(\varepsilon, \delta)$ -differentially private if for all datasets  $D_1$  and  $D_2$  that differ on a single data point, and all  $S \subseteq \text{Range}(\mathcal{A})$ ,

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^\varepsilon \cdot \Pr[\mathcal{A}(D_2) \in S] + \delta. \quad (7.1)$$

In this definition,  $\varepsilon$  (epsilon) is a non-negative parameter that measures the loss of privacy, with smaller values indicating stronger privacy. The parameter  $\delta$  represents the probability of breaking differential privacy, ideally close to or equal to 0.

Despite enabling provable error guarantees for Unlearning methods, Differential Privacy requires strong model and algorithmic assumptions, making MU, derived from it, potentially less effective against practical adversaries [106].

### 7.3 Methodology

**Setting.** Starting with a **training set**  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$  and a trained model  $f_O$ , referred to as the **original model**, the objective of a MU method  $U$  is to *remove* the influence of a particular subset of training set  $\mathcal{D}_F = \{(\mathbf{x}_i, y_i)\}_{i=1}^K \subset \mathcal{D}$  referred to as the **forget set** where

$K \ll N$ . The rest of the training set  $\mathcal{D}_R = \mathcal{D} \setminus \mathcal{D}_F$  is called the **retain set**. The forget and retain sets are distinct and complementary subsets of the training set. The outcome of MU is an **unlearned** model  $f_U$ , the aim for which is to *perform* on par with a model *retrained from scratch* on  $\mathcal{D}_R$ ; this latter model  $f_R$  is referred to as the **retrained** model. We denote the weights of the original model and retrained models' weights as  $\theta_O$  and  $\theta_R$ , respectively. For evaluations, we consider two held-out sets: the **validation set**  $\mathcal{D}_V$  and **test set**  $\mathcal{D}_T$ , both drawn from the same distribution as  $\mathcal{D}$ . We consider the accuracy of the retrained model to be the optimal accuracy. One critical assumption we make is that the MU method has access to  $\theta_O$ ,  $\mathcal{D}_R$ ,  $\mathcal{D}_F$ , and  $\mathcal{D}_V$ .

**(1) Unlearning Evaluation.** To evaluate the success of unlearning, one approach is to check whether data points in  $\mathcal{D}_F$  still influence the predictions made by the unlearned model [26, 119, 93]. This is commonly done via *influence functions* [115, 12, 88], and membership inference attacks (MIA) [156]. MIA has become one of the most common approaches for evaluating MU methods. It aims to determine whether specific data points were part of the original training dataset based on the unlearned model. We compute the population-based U-MIA, denoted with  $\text{MIA}(\mathcal{D}, f_U)$  evaluated on data  $\mathcal{D}^1$ . Using this we define the **discernibility** metric as  $\text{Disc}(\mathcal{D}_V, f_U) = |2 \times \text{MIA}(\mathcal{D}, f_U) - 1| \in [0, 1]$ , and similarly, **indiscernibility** is given by  $\text{Indisc}(\mathcal{D}, f_U) = 1 - \text{Disc}(\mathcal{D}, f_U) \in [0, 1]$ . We set  $\mathcal{D}$  to either the test set  $\mathcal{D}_T$  or validation set  $\mathcal{D}_V$ .

A more recent MIA variation is the unlearning likelihood ratio attack (U-LiRA) [119]. As highlighted by [93], U-LiRA is a more robust evaluation approach for approximate MU. Nonetheless, U-LiRA is much more computationally demanding than U-MIA. We evaluate the methods that defeat the weaker, less expensive U-MIA attack, additionally against the U-LiRA attack.

**(2) Accuracy.** The classification accuracy of the unlearned model on the retained set should be as close as possible to that of the original model. First, we consider three metrics derived directly from the model's classification accuracy on different sets: **retain accuracy** (RA), **forget accuracy** (FA), and **test accuracy** (TA). RA is defined as

$$\text{RA}(\mathcal{D}_R, f_U) = \frac{1}{|\mathcal{D}_R|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_R} \mathbf{1}_{y_i = f_U(\mathbf{x}_i)} \in [0, 1]. \quad (7.2)$$

---

<sup>1</sup>The methodology for this follows the classic MIA: we compute the losses on  $\mathcal{D}_F$  and  $\mathcal{D}_V$ , we shuffle and trim them so that they are of equal size. We then train logistic regression models in a 10-fold cross-validation and compute the average accuracy across the folds.

The metrics for FA and TA can be derived by replacing  $\mathcal{D}_R$  with  $\mathcal{D}_F$  and  $\mathcal{D}_T$ , respectively. Second, while RA, FA, and TA give us insight into the overall accuracy of the unlearned model, they do not capture how well it performs compared to a retrained model  $f_R$ . Considering the  $f_R$  model as a gold standard, we derive three more metrics: **retain retention** (RR), **forget retention** (FR), **test retention** (TR), where RR is given by,

$$\text{RR}(f_U, f_R) = \frac{\text{RA}(\mathcal{D}_R, f_U)}{\text{RA}(\mathcal{D}_R, f_R)} \in [0, +\infty), \quad (7.3)$$

and formulas for FR and TR can be derived using FA and TA, respectively. An unlearned model with a score of 1 indicates that its accuracy perfectly matches the accuracy of the reference retrain model. A score below 1 indicates that the model under-performs, and a score above 1 indicates that the model over-performs. We further define the **retention deviation** (RetDev) as:

$$\text{RetDev} = |\text{RR}(f_U, f_R) - 1| + |\text{FR}(f_U, f_R) - 1| + |\text{TR}(f_U, f_R) - 1| \in [0, +\infty), \quad (7.4)$$

which provides information on the cumulative divergence of the unlearned model regarding retention score. The closer to 0, the better, as 0 indicates that the model perfectly matches the performance of the retrained model.

**(3) Efficiency.** run-time efficiency (RTE) of a MU method should ideally be higher than the naive approach of just retraining the model from scratch on the retained set. As the high computational cost of the retraining algorithm motivated the development of approximate MU methods, we evaluate how much faster each MU method is compared to the retraining algorithm. We define RT of  $U$  as the number of seconds it takes to complete, denoted as  $\text{RT}(U)$  (we use the same machine and resources for all the experiments). To indicate the relative speedup compared to the retrained model, we define the RTE of an unlearn method  $U$  as:

$$\frac{\text{RT}(U_R)}{\text{RT}(U)} \in [0, +\infty), \quad (7.5)$$

where  $U_R$  denotes the retrain method. The RTE of retraining from scratch would thus be 1; any method with an RTE less than 1 is slower than retraining from scratch, and vice versa.

### 7.3.1 Machine Unlearning Methods

We briefly discuss the main unlearning methods considered.

#### Classical Baselines

**FineTune (FT)** finetunes the original model  $f_O$  on only the retain set  $\mathcal{D}_R$  for several epochs.

**Successive Random Labels (SRL)** the model is trained on both the forget set  $\mathcal{D}_F$  and  $\mathcal{D}_R$  where the labels of  $\mathcal{D}_F$  are randomly assigned at each epoch.

**Gradient Ascent (GA)** trains the model using gradient *ascent* steps on the  $\mathcal{D}_F$ .

#### State-of-the-art MU methods

Expanding upon the classical baselines, we additionally evaluate 15 recent MU methods. We first discuss the seven top-performing methods from the Machine Unlearning Competition 2023 on Kaggle.

**Forget-Contrast-Strengthen (FCS)** [41] minimizes the Kullback-Leibler Divergence (KLD) between the model’s output on  $\mathcal{D}_F$  and uniform distribution over the output classes, then alternatively optimizes a contrastive loss between the model’s outputs on  $\mathcal{D}_R$  and  $\mathcal{D}_F$ , and minimizes the cross-entropy loss on  $\mathcal{D}_R$ .

**Masked-Small-Gradients (MSG)** [42] accumulates gradients via gradient *descent* on the  $\mathcal{D}_R$  and gradient *ascent* on the  $\mathcal{D}_F$ , then reinitializes weights with the smallest absolute gradients while dampening subsequent weights updates on the  $\mathcal{D}_R$  for the other weights.

**Confuse-Finetune-Weaken (CFW)** [43] injects noise into the convolutional layers and then trains the model using a class-weighted cross-entropy on  $\mathcal{D}_R$ , then injects noise again toward the final epochs.

**Prune-Reinitialize-Match-Quantize (PRMQ)** [44] first prunes the model via L1 pruning, reinitializes parts of the model, optimizes it using a combination of cross-entropy and a mean-

squared-error on the entropy between the outputs of  $f_O$  and  $f_U$  on  $\mathcal{D}_R$  and finally converts  $f_U$ 's weights to half-precision floats.

**Convolution-Transpose** [45] simply transposes the weights in the convolutional layers and trains on  $\mathcal{D}_R$ .

**Knowledge-Distillation-Entropy (KDE)** [46] uses a teacher-student setup. Both student and teacher start as copies of the original model, and then the student's first and last layers are reinitialized. The student  $f_U$  minimizes its Kullback-Leibler Divergence (KLD) with the  $f_O$  over  $\mathcal{D}_V$ , then minimizes a combination of losses: a soft cross-entropy loss between  $f_U$  and  $f_O$ , a cross-entropy loss on outputs of  $\mathcal{D}_R$  from  $f_U$ , and the KLD between  $f_U$  and  $f_O$  on  $\mathcal{D}_R$ .

**Repeated-Noise-Injection (RNI)** [47] first reinitializes the final layer of the model, then repeatedly injects noise in a different layer of the model while training on the  $\mathcal{D}_R$ .

We further consider eight state-of-the-art methods introduced in the literature.

**Fisher Forgetting (FF)** [82, 66] adds noise to  $f_O$  with zero mean and covariance determined by the 4th root of Fisher Information matrix with respect to  $\theta_O$  on  $\mathcal{D}_R$ .

**Influence Unlearning (IU)** [105, 180, 106] uses Influence Functions [48] to determine the change in  $\theta_O$  if a training point is removed from the training loss. IU estimates the change in model parameters from  $\theta_O$  to the model trained without a given data point. We use the first-order WoodFisher-based approximation from [106].

**Catastrophic Forgetting - K (CF-K)** [80] freezes the first layers then trains the last  $k$  layers of the model on  $\mathcal{D}_R$ .

**Exact Unlearning - K (EU-K)** [80] freezes the first layers and then restores the weights of the last  $k$  layers to their initialization state. We randomly reinitialize the weights instead so that the method no longer requires knowledge about the training process of  $f_O$ .

**SCRUB** [119] leverages a student-teacher setup where the model is optimized for three objectives: matching the teacher's output distribution on  $\mathcal{D}_R$ , correctly predicting the  $\mathcal{D}_R$  set and ensuring the output distributions of the teacher and student diverge on the  $\mathcal{D}_F$

**Saliency Unlearning (SaLUN)** [66] determines via gradient *ascent* which weights of  $\theta_O$  are the most relevant to  $\mathcal{D}_F$ , then trains the model simultaneously on  $\mathcal{D}_R$  and  $\mathcal{D}_F$  with random

labels on  $\mathcal{D}_F$ , while dampening the gradient propagation based on the selected weights.

**Negative Gradient Plus (NG+)** [119] is an extension of the Gradient Ascent approach where additionally a gradient descent step is taken over the  $\mathcal{D}_R$ .

**Bad Teacher (BT)** [39] uses a teacher-student approach with two teachers: the original model and a randomly initialized model - the bad teacher-, the student starts as a copy of  $f_U$  then learns to mimic the  $f_O$  on  $\mathcal{D}_R$  and the bad teacher on the  $\mathcal{D}_F$ .

### 7.3.2 Datasets

**CIFAR-10** CIFAR-10 is a widely used dataset in computer vision and machine learning. It comprises 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 testing images. CIFAR-10 represents a diverse range of everyday objects, such as airplanes, automobiles, birds, and cats, making image classification challenging. The simplicity of the images and variety of categories makes CIFAR-10 a suitable dataset to test the efficacy of machine unlearning algorithms in effectively unlearning information without compromising the model’s performance on the remaining data.

*Data Augmentations:* random cropping to 32x32 with 4-pixel padding, 50% random horizontal flipping, and per-channel normalization with a mean of [0.4919, 0.4822, 0.4465] and standard deviation of [0.2023, 0.1994, 0.2010]. At test time, we resize to 32x32 and normalize.

**CIFAR-100** CIFAR-100 is a more complex extension of CIFAR-10, containing 100 classes with 600 images per class, split into 500 and 100 training and testing images per class. Each class is labeled with a "fine" label and grouped into 20 "coarse" labels, adding another layer of classification difficulty. The increased number of classes and finer granularity make CIFAR-100 an interesting dataset for machine unlearning benchmarks. It is more challenging for models to forget specific classes or groups while retaining knowledge of others, allowing for testing the unlearning method’s precision and effectiveness in handling complex datasets.

*Data Augmentations:* random cropping to 32x32 with 4-pixel padding, 50% random horizontal flipping, and per-channel normalization with a mean of [0.5071, 0.4865, 0.4409] and standard deviation of [0.2673, 0.2564, 0.2762]. At test time, we resize to 32x32 and normalize.

**MNIST** The MNIST dataset is a well-known benchmark in handwritten digit recognition. It comprises 70,000 grayscale images of handwritten digits (0-9), 60,000 of which were used for training and 10,000 for testing. Each image is 28x28 pixels in size. We consider MNIST due to its simplicity and extensive research and development history. The simplicity of MNIST allows researchers to focus on the fundamental aspects of unlearning techniques without the complexity introduced by color or high resolution, providing a clear assessment of unlearning algorithms in a controlled setting.

*Data Augmentations:* conversion to 3 channels, resizing to 32x32 such that both ResNet18 and TinyViT use the same input resolution, 50% random horizontal flipping, and per-channel normalization with a mean of  $[0.1307, 0.1307, 0.1307]$  and standard deviation of  $[0.3081, 0.3081, 0.3081]$ . We convert to 3 channels at test time, resize to 32x32, and normalize.

**Fashion MNIST** Fashion MNIST is a more challenging replacement for MNIST. It contains 70,000 grayscale images of fashion items in 10 categories: shirts, trousers, and sneakers. Like MNIST, each image is 28x28 pixels, but the increased complexity and variability of clothing items make it a more challenging classification task. Fashion MNIST provides a more realistic and intricate dataset than MNIST, testing the unlearning methods' ability to handle real-world-like variability and ensuring that they can effectively remove learned information while maintaining performance on a moderately complex dataset.

*Data Augmentations:* conversion to 3 channels, resizing to 32x32, 50% random horizontal flipping, and per-channel normalization with a mean of  $[0.2860, 0.2860, 0.2860]$  and standard deviation of  $[0.3560, 0.3560, 0.3560]$ . We convert to 3 channels at test time, resize to 32x32, and normalize.

**UTKFace** UTKFace is a large-scale face dataset containing over 20,000 images of faces with annotations of age, gender, and ethnicity. The images vary in size and cover a wide range of ages, from 0 to 116. UTKFace is particularly interesting due to the sensitive nature of the data and the need for privacy-preserving techniques.

*Data Augmentations:* resizing to 224x224, and per-channel normalization with a mean of  $[0.485, 0.456, 0.406]$  and standard deviation of  $[0.229, 0.224, 0.225]$ . We apply the same transformation at test time.

For each dataset, the Original and Retrained models are trained using the same hyper-parameters (provided in Table 7.1)

Table 7.1: Summary of the number epochs, learning rate, and batch size for each dataset and model used to train the Original and Retrained models.

| Dataset      | Model           | Epochs | Learning Rate | Batch Size |
|--------------|-----------------|--------|---------------|------------|
| FashionMNIST | ResNet18        | 50     | 0.1           | 256        |
|              | TinyViT         | 50     | 0.1           | 256        |
| MNIST        | ResNet18        | 50     | 0.1           | 256        |
|              | TinyViT         | 50     | 0.1           | 256        |
| CIFAR-10     | ResNet18        | 182    | 0.1           | 256        |
|              | (LiRA) ResNet18 | 91     | 0.1           | 256        |
|              | TinyViT         | 182    | 0.1           | 256        |
| CIFAR-100    | ResNet18        | 182    | 0.1           | 256        |
|              | TinyViT         | 182    | 0.1           | 256        |
| UTKFace      | ResNet18        | 50     | 0.1           | 128        |
|              | TinyViT         | 50     | 0.1           | 128        |

### 7.3.3 Neural Network Architectures

We consider two families, ResNet (Residual Network) [96] and ViT (Vision Transformer) [58], which are prominent architectures in computer vision. We consider ResNet18 and a TinyViT[183] with approximately 11M learnable parameters for a fair comparison between two fundamentally different architectures. This provides insights into the impact of architectural differences on the unlearning process and helps understand the trade-offs between convolutional and transformer-based models regarding reliability and computational efficiency.

**ResNet: ResNet18** Introduced by He et al. [96], it facilitates the training of deep networks through shortcut connections, which mitigates the problem of vanishing gradients. The ResNet18 is known for its balance between performance and computational efficiency.

**ViT: TinyViT** Vision Transformer (ViT), introduced by Dosovitskiy et al. [58], adapts the transformer architecture to image classification by treating images as sequences of patches. We consider TinyViT from Wu et al. [183], as it is a compact version of ViT designed to be parameter-efficient while maintaining high performance.

### 7.3.4 Privacy Evaluation

#### Unlearning-Membership Inference Attack (U-MIA)

A common approach to evaluate the quality of unlearning methods is to attack the unlearned models with a form of Membership inference Attack (MIA). Membership Inference Attacks attempt to determine whether a specific data point was part of the model train data. The efficacy of the Membership Inference Attack has been used as a metric to evaluate the success of unlearning algorithms. A general approach to such an attack is as follows. Assume  $f_\theta$  is a trained model with parameters  $\theta$ , and let  $\mathcal{L}$  be a loss function, such as the cross-entropy loss. Then, compute the losses for each sample from two sets of data  $A$  and  $B$  (of equal size) and train a binary classification model such as logistic regression with labels  $y_i^A = 1$  for points  $i$  in  $A$  and  $y_i^B = 0$  for points  $i$  in  $B$ . An accuracy score from the classifier close to 1.0 indicates that the classifier can perfectly distinguish between samples from  $A$  and  $B$  based on the loss values. A score of 0.5 indicates that the ability to distinguish is close to random.

#### Unlearning -Likelihood Ratio Attack (U-LiRA)

The performance of a general MIA can be improved by considering, *e.g.*, a per-sample attack such as LiRA [26, 93]. For any given point, we wish to determine whether the outputs from the unlearned models differ from those of models that have never seen the data point. To assess the attack robustly, we evaluate it across multiple models, using shadow models trained on various retain/forget sets. Specifically, we first train  $n$  models based on  $n$  splits of the training data. This train data is then split into ten random retain and forget splits; hence, we unlearn a total of  $10 \times n$  models. We then perform hyper-parameter sweeps, similar to what we do in the original results and unlearn using the optimal hyper-parameters, except that we consider  $\frac{n}{2}$  sweeps and conduct 200 trials per sweep to determine the best hyper-parameters. In our setting, we set  $n = 64$ .

## 7.4 Experimental Evaluation

**Experiments.** We evaluate the 18 recent MU methods as described in Section 7.3.1 across 5 benchmark datasets: MNIST [124], FashionMNIST [186], CIFAR-10 [117], CIFAR-100 [117], and UTK-Face [200]. These datasets vary in difficulty, number of classes, instances per class, and image sizes. We consider two model architectures: a TinyViT and a ResNet18 model. Hence, we evaluated nine different combinations of models and architectures: ResNet18 and TinyViT on MNIST, FashionMNIST, CIFAR-10, CIFAR-100, and ResNet18 on UTKFace. More information on the data sets, hyperparameters, and data augmentations used to train the original and retrained models is provided in the appendix 7.3.2. *We construct the forget set by sampling 10% of  $\mathcal{D}$ .*

The performance of the MU methods can change across datasets, model configurations, and model initializations; a reliable MU method remains consistent across these changes. For each method, dataset, and model combination, we unlearn from the Original models initialized using 10 different seeds and consider the average performance across seeds.

A further observation is that prior research tends to compare MU methods with default hyperparameters, potentially leading to a less competitive performance of the method. To ensure that each method performs optimally, we perform three hyperparameter sweeps to find the best set of hyperparameters for each method. To ensure a fair comparison, we use the same number of searches for each method. Each hyper-parameter sweep uses 100 trials to minimize four loss functions: Retain Loss ( $\mathcal{L}_{\text{Retain}}$ ), Forget Loss ( $\mathcal{L}_{\text{Forget}}$ ), Val Loss ( $\mathcal{L}_{\text{Val}}$ ), and Val MIA ( $\mathcal{L}_{\text{Val-MIA}}$ ) given by

$$\mathcal{L}_{\text{Retain}} = \alpha \times |\text{RA}(f_U) - \text{RA}(f_R)|, \quad \mathcal{L}_{\text{Forget}} = \beta \times |\text{FA}(f_U) - \text{FA}(f_R)|, \quad (7.6)$$

$$\mathcal{L}_{\text{Val}} = \gamma \times |\text{VA}(f_U) - \text{VA}(f_R)|, \quad \mathcal{L}_{\text{Val-MIA}} = \eta \times \text{Disc}(\mathcal{D}_V, f_U), \quad (7.7)$$

where the  $\mathcal{L}_{\text{Retain}}$  captures the divergence in accuracy between the retrained and unlearned model over the  $\mathcal{D}_R$ ,  $\mathcal{L}_{\text{Forget}}$  and  $\mathcal{L}_{\text{Val}}$  capture the divergence over  $\mathcal{D}_F, \mathcal{D}_V$  respectively and  $\mathcal{L}_{\text{Val-MIA}}$  captures whether the loss distributions over  $\mathcal{D}_F$  and  $\mathcal{D}_V$  are distinguishable from one another via the discernibility score defined in Section 7.3. We set  $\alpha = \beta = \gamma = \frac{1}{3}$  and  $\eta = 1$  as we found these values to balance the importance of importance retention and resilience to

Membership Inference Attacks. Per unlearn method, we use the hyperparameter configuration that minimizes the four loss terms when evaluating the method. Thus, for each unlearning method, we first unlearn 300 models to do the hyper-parameter sweep and then unlearn 10 models with the best set of hyper-parameters, leading to 5,580 models per dataset for a given architecture, for a total of 50,220 for the 9 dataset/model combinations.

**Ranking.** A challenge in comparing MU method performance comes from the potential proximity of the evaluation metrics. As a simple example, suppose we have four methods  $U_1, \dots, U_4$  with accuracies: 98%, 99%, 50%, 1%, respectively; if we rank the methods, the rank itself would not be representative of the fact that e.g.  $U_1$  and  $U_2$  are much above  $U_3$  and  $U_4$ . To enable distinctions based on proximities, we use Agglomerative Clustering and define cut-off points such that we obtain three clusters: (1) Best performers (G1), (2) Average performers (G2), and (3) Worst performers (G3).

If a method does not produce 10 usable models, one per original model, it is assigned to a Failed group (F). For each method, we count the times it appears in the three groups (with nine being the maximum). To obtain a final ranking of the methods, we first rank them using the number of times they appear in the Best Performers group (G1); if ties occur, we break them with the Average Performers (G2) group. If ties persist, the Worst Performers (G3) group is the final tie-breaker. This method ensures a clear and fair ranking by considering each performance group in order of importance.

### 7.4.1 Main Results

Table 7.2 presents the main results of our evaluations on MU methods based on Retention Deviation and Indiscernibility. The results for the run-time efficiency are shown in Table 7.3.

**On the reliability of baselines.** The commonly used baseline FT trains the original model only on the retain set for several epochs to enable the model to forget information about the forget set. In our evaluation, FT performs best based on the Retention Deviation, and is ranked third based on Indiscernibility. The latter observation may be that FT does not explicitly unlearn the forget set or perturb the model parameters. Based on these results, it is a reasonable baseline to evaluate against. We, however, remark that since the mechanism

Table 7.2: Ranking by performance on Retention Deviation and Indiscernibility across datasets and architectures. We count the number of times each method appears in the Best Performers group (G1), Average performance group (G2) and Worst performers group (G3) (see §7.4). The final rank is computed based on the number of times the method appears in G1—with occurrences in G2 and G3 used to break ties if needed. If a method does not produce any usable models, it is assigned to a Failed group (F). Three methods appear in the top 3 for both performance measures: MSG (1st and 1st), CT (3rd and 1st), and KDE (3rd and 2nd).

| Rank | Method | Retention Deviation |    |    |   | Rank | Method | Indiscernibility |    |    |   |
|------|--------|---------------------|----|----|---|------|--------|------------------|----|----|---|
|      |        | G1                  | G2 | G3 | F |      |        | G1               | G2 | G3 | F |
| 1    | FT     | 8                   | 1  | 0  | 0 | 1    | CT     | 9                | 0  | 0  | 0 |
| 1    | MSG    | 8                   | 1  | 0  | 0 | 1    | MSG    | 9                | 0  | 0  | 0 |
| 2    | PRMQ   | 7                   | 2  | 0  | 0 | 2    | CFW    | 7                | 2  | 0  | 0 |
| 3    | CT     | 7                   | 1  | 1  | 0 | 2    | RNI    | 7                | 2  | 0  | 0 |
| 3    | KDE    | 7                   | 1  | 1  | 0 | 2    | KDE    | 7                | 2  | 0  | 0 |
| 3    | CFW    | 7                   | 1  | 1  | 0 | 3    | FT     | 6                | 3  | 0  | 0 |
| 4    | FCS    | 6                   | 3  | 0  | 0 | 3    | PRMQ   | 6                | 3  | 0  | 0 |
| 4    | SalUN  | 6                   | 3  | 0  | 0 | 3    | SalUN  | 6                | 3  | 0  | 0 |
| 5    | NG+    | 5                   | 4  | 0  | 0 | 4    | SRL    | 6                | 2  | 1  | 0 |
| 5    | SRL    | 5                   | 4  | 0  | 0 | 5    | NG+    | 5                | 4  | 0  | 0 |
| 6    | SCRUB  | 4                   | 3  | 1  | 1 | 5    | FCS    | 5                | 4  | 0  | 0 |
| 7    | BT     | 2                   | 7  | 0  | 0 | 6    | SCRUB  | 5                | 3  | 0  | 1 |
| 7    | RNI    | 2                   | 7  | 0  | 0 | 7    | BT     | 4                | 5  | 0  | 0 |
| 8    | CF-k   | 2                   | 3  | 2  | 2 | 8    | CF-k   | 1                | 2  | 4  | 2 |
| 9    | IU     | 1                   | 0  | 2  | 6 | 9    | EU-k   | 1                | 2  | 2  | 4 |
| 10   | EU-k   | 0                   | 5  | 0  | 4 | 10   | GA     | 0                | 4  | 4  | 1 |
| 11   | GA     | 0                   | 1  | 7  | 1 | 11   | IU     | 0                | 0  | 3  | 6 |
| 12   | FF     | 0                   | 0  | 0  | 9 | 12   | FF     | 0                | 0  | 0  | 9 |

underlying FT (training on the Retain set to maintain performance) is common to many other methods, these methods may inherit its susceptibility to MIA. Another common baseline, GA which performs gradient ascent on the forget set, performs poorly across both metrics. Its more recent variation, NG+, which uses an additional retain set correction, ranks fifth for both metrics, making it a more suitable baseline.

**On the reliability of newly proposed unlearning methods.** MSG obtained the first rank in both performance retention deviation and indiscernibility. Its unique approach identifies the parameters in the Convolutional layers that most contribute to the information to be forgotten. This strategy, differing from FT, allows MSG to retain performance from the Retain set while modifying the weights that are more relevant to the Forget set. PRMQ ranks second in

Table 7.3: Run Time Efficiency on ResNet for the top performing methods. CT is the fastest on average, MSG runs up to 5x faster than naive retraining.

| Unlearner | CIFAR-10 | CIFAR-100 | MNIST | FashionMNIST | UTKFace | Average |
|-----------|----------|-----------|-------|--------------|---------|---------|
| MSG       | 6.80     | 4.49      | 4.29  | 3.32         | 7.57    | 5.29    |
| CFW       | 4.67     | 6.17      | 4.29  | 4.90         | 5.54    | 5.11    |
| PRMQ      | 4.93     | 4.34      | 3.77  | 3.77         | 5.88    | 4.54    |
| CT        | 17.49    | 11.82     | 5.83  | 4.47         | 13.34   | 10.59   |
| KDE       | 6.33     | 3.98      | 3.27  | 3.22         | 8.19    | 5.00    |
| FT        | 8.15     | 5.16      | 5.07  | 4.29         | 5.78    | 5.69    |

performance retention, making it one of the top performers. However, it suffers from the same lower performance in Indiscernibility as FT. PRMQ, however, does not leverage the Forget set; instead, it performs a form of knowledge distillation by attempting to reproduce the results of the original models on the Retain set. Additionally, during the pruning phase, it reinitializes the weights for the MLP and Convolutional layers. CT ranks third in Performance Retention and first in Indiscernibility. It is interesting to note that CT and MSG are both consistently among the top performers in Indiscernibility, whereas FT performs poorly.

**On the robustness across architectures.** A critical aspect of a good MU method is its ability to generalize across various DNN architectures. We conducted experiments with both ResNet18 and TinyViT. Methods such as CT, despite being tailored to Convolutional Neural Networks (CNN) models, still perform competitively when applied to Vision Transformers. While proposed for CNN layers, methods such as CT and MSG work well on Vision Transformer as one can leverage the 2D Convolutions used in Positional Encoders. We provide additional details on the ranking based on architectures in Appendix 7.4.3.

**On the speed of the unlearn methods.** MSG, the best MU candidate, runs 7.6x faster than retraining from scratch on UTKFace and 3.3x on FashionMNIST. On average, MSG runs 5.3x faster than retraining. The fastest method is CT, which achieves a speedup of 17.5x compared to Retraining. This speedup stems from its simple approach: transposing the convolutional layer weights (Table 7.3).

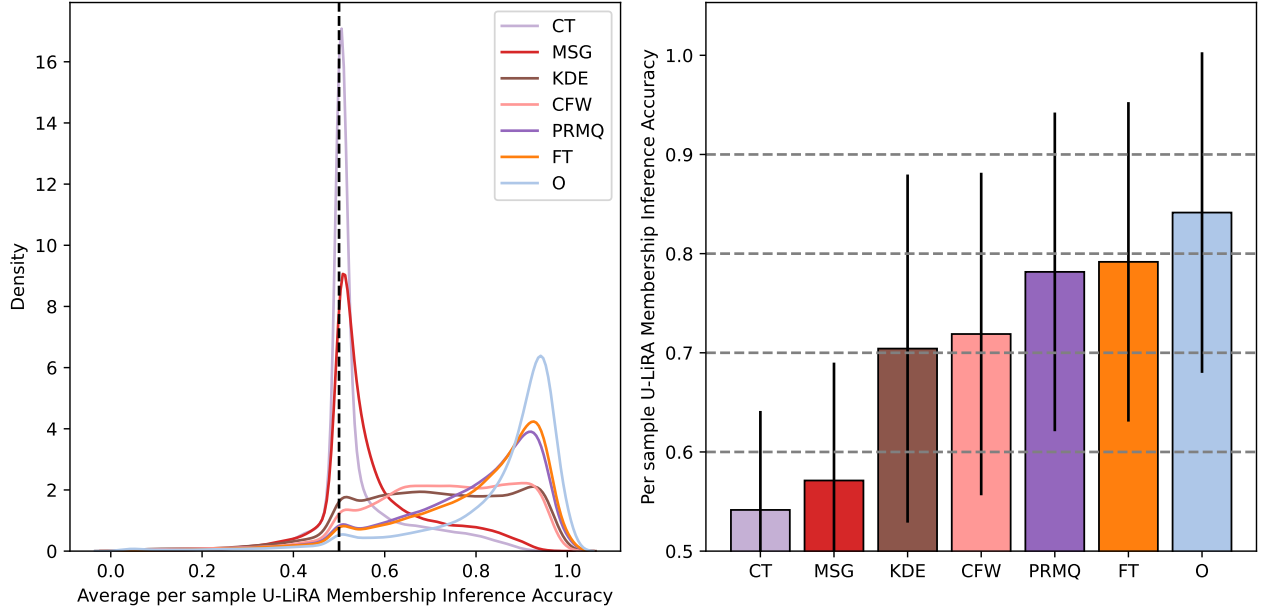


Figure 7.1: U-LiRA on CIFAR-10 on ResNet18 models. Both CT and MSG, which ranked first against U-MIA, showed great resilience against the U-LiRA attack.

**On the performance when evaluated against a stronger MIA.** From the results above, we note that there exist MU methods that perform fast and reliably across datasets, architectures, and initializations. Specifically, CT, FT, and MSG seem strong candidates for reliable MU methods. However, as has also been highlighted in prior work [174], MIA has been debated as a strong metric, as its ability to assess MU is hampered by its ability to infer data membership. In line with this, recent works have introduced more powerful variations on MIA [191] with [119] proposing the stronger U-LiRA attack for MU. For the best performers in Table 7.2, we apply the attack setup from [93] where we generate a total of 640 models (with varying train, retain, and forget sets) and for each unlearn method, perform a hyperparameter sweep to find the best configuration (for details, see Appendix 7.3.4). We determine for each data point its U-LiRA Inference Accuracy and report for each method its average and standard deviation (see Figure 7.1). From this, we conclude that MSG, as well as CT, resist U-LiRA attacks. Thus, Both MSG and CT rank first in terms of Performance Retention and Indiscernibility based on U-MIA and are robust against a stronger variation of MIA.

## 7.4.2 Per dataset results

Here, we present the ResNet18 and the TinyViT results across datasets.

Table 7.4: MNIST - ResNet18: Retain Accuracy (RA), Forget Accuracy (FA), Test Accuracy (TA), Retain Retention (RR), Forget Retention (FR), Test Retention (TR), Performance Retention Deviation (RetDev), Indiscinerbility concerning the Test Set (Indisc), U-MIA on the Test set (T-MIA) and RunTime Efficiency (RTE)

|           | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| unlearner |      |      |      |      |      |      |        |        |       |       |
| BT        | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.50  | 9.76  |
| CF-k      | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.98   | 0.51  | 5.86  |
| CFW       | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 4.29  |
| CT        | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 5.83  |
| EU-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FCS       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 2.11  |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 0.99   | 0.51  | 5.07  |
| GA        | 0.98 | 0.98 | 0.97 | 0.98 | 0.99 | 0.98 | 0.04   | 0.99   | 0.51  | 33.97 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| KDE       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 3.27  |
| MSG       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 4.29  |
| NG+       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 3.12  |
| O         | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.98   | 0.51  | 1.10  |
| PRMQ      | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 3.77  |
| R         | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00  |
| RNI       | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 4.70  |
| SCRUB     | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| SRL       | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 0.99 | 0.01   | 0.99   | 0.51  | 8.55  |
| SalUN     | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 0.01   | 0.99   | 0.50  | 3.19  |

**ResNet18** We provide the tables with Retain Accuracy (RA), Forget Accuracy (FA), Test Accuracy (TA), Retain Retention (RR), Forget Retention (FR), Test Retention (TR), Performance Retention Deviation (RetDev), Indiscinerbility concerning the Test Set (Indisc), U-MIA on the Test set (T-MIA) and RunTime Efficiency (RTE) for every dataset using the ResNet18 model on MNIST (Table 7.4), FashionMNIST (Table 7.5), CIFAR-10 (Table 7.6), CIFAR-100 (Table 7.7) and UTKFace (Table 7.8). In general, CIFAR-100 provides the most visible differences, as the performance on the retain set is much higher than on the test. Datasets such as MNIST and FashionMNIST tend to show smaller differences between the methods as the performance on both the Retain and Test sets are similar.

**TinyViT** We provide the tables with RA, FA, TA, RR, FR, TR, RetDev, Indisc, T-MIA and RTE for MNIST (Table 7.9), FashionMNIST (Table 7.10), CIFAR-10 (Table 7.11) and CIFAR-100 (Table 7.12) using the TinyViT model.

Table 7.5: FashionMNIST - ResNet18

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| BT        | 1.00 | 0.96 | 0.92 | 1.00 | 1.03 | 0.99 | 0.04   | 0.98   | 0.51  | 13.57 |
| CF-k      | 0.98 | 0.97 | 0.91 | 0.98 | 1.05 | 0.98 | 0.09   | 0.92   | 0.54  | 16.31 |
| CFW       | 1.00 | 0.95 | 0.92 | 1.00 | 1.02 | 0.99 | 0.03   | 0.97   | 0.51  | 4.90  |
| CT        | 1.00 | 0.92 | 0.92 | 1.00 | 0.99 | 0.99 | 0.02   | 0.99   | 0.50  | 4.47  |
| EU-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FCS       | 0.98 | 0.93 | 0.91 | 0.98 | 1.00 | 0.98 | 0.04   | 0.98   | 0.51  | 2.79  |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 1.00 | 0.95 | 0.92 | 1.00 | 1.02 | 1.00 | 0.02   | 0.98   | 0.51  | 4.29  |
| GA        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| IU        | 1.00 | 1.00 | 0.93 | 1.00 | 1.08 | 1.00 | 0.08   | 0.87   | 0.56  | 14.97 |
| KDE       | 1.00 | 0.93 | 0.92 | 1.00 | 1.00 | 1.00 | 0.01   | 0.99   | 0.50  | 3.22  |
| MSG       | 1.00 | 0.93 | 0.91 | 1.00 | 1.00 | 0.99 | 0.01   | 0.98   | 0.51  | 3.32  |
| NG+       | 0.99 | 0.94 | 0.91 | 0.99 | 1.01 | 0.99 | 0.03   | 0.99   | 0.49  | 3.21  |
| O         | 1.00 | 1.00 | 0.93 | 1.00 | 1.08 | 1.00 | 0.08   | 0.87   | 0.56  | 1.11  |
| PRMQ      | 0.98 | 0.93 | 0.91 | 0.98 | 1.00 | 0.99 | 0.04   | 0.98   | 0.51  | 3.77  |
| R         | 1.00 | 0.93 | 0.92 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00  |
| RNI       | 0.98 | 0.93 | 0.91 | 0.98 | 1.00 | 0.98 | 0.04   | 0.98   | 0.51  | 3.25  |
| SCRUB     | 0.95 | 0.93 | 0.90 | 0.95 | 1.00 | 0.98 | 0.08   | 0.97   | 0.51  | 5.62  |
| SRL       | 1.00 | 0.97 | 0.92 | 1.00 | 1.04 | 1.00 | 0.05   | 0.98   | 0.51  | 24.85 |
| SalUN     | 0.99 | 0.97 | 0.92 | 0.99 | 1.04 | 0.99 | 0.06   | 0.98   | 0.51  | 25.02 |

Table 7.6: CIFAR-10 - ResNet18

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| BT        | 0.94 | 0.87 | 0.84 | 0.94 | 1.00 | 0.96 | 0.10   | 0.97   | 0.48  | 51.96 |
| CF-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| CFW       | 1.00 | 0.81 | 0.80 | 1.00 | 0.92 | 0.92 | 0.16   | 1.00   | 0.50  | 4.67  |
| CT        | 1.00 | 0.82 | 0.81 | 1.00 | 0.93 | 0.93 | 0.14   | 0.99   | 0.50  | 17.49 |
| EU-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FCS       | 0.99 | 0.86 | 0.84 | 0.99 | 0.98 | 0.96 | 0.07   | 0.98   | 0.49  | 22.53 |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 1.00 | 0.84 | 0.82 | 1.00 | 0.96 | 0.95 | 0.09   | 1.00   | 0.50  | 8.15  |
| GA        | 0.91 | 0.89 | 0.81 | 0.91 | 1.02 | 0.93 | 0.18   | 0.92   | 0.54  | 91.48 |
| IU        | 0.95 | 0.94 | 0.84 | 0.95 | 1.08 | 0.97 | 0.16   | 0.91   | 0.55  | 64.61 |
| KDE       | 0.98 | 0.84 | 0.80 | 0.98 | 0.96 | 0.92 | 0.15   | 0.97   | 0.52  | 6.33  |
| MSG       | 1.00 | 0.85 | 0.83 | 1.00 | 0.97 | 0.95 | 0.08   | 0.99   | 0.51  | 6.80  |
| NG+       | 0.97 | 0.89 | 0.85 | 0.98 | 1.02 | 0.97 | 0.07   | 0.98   | 0.51  | 12.89 |
| O         | 0.96 | 0.96 | 0.85 | 0.96 | 1.10 | 0.98 | 0.16   | 0.89   | 0.55  | 1.08  |
| PRMQ      | 1.00 | 0.86 | 0.83 | 1.00 | 0.98 | 0.95 | 0.07   | 0.98   | 0.51  | 4.93  |
| R         | 1.00 | 0.87 | 0.87 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00  |
| RNI       | 1.00 | 0.83 | 0.81 | 1.00 | 0.95 | 0.93 | 0.12   | 0.99   | 0.50  | 3.60  |
| SCRUB     | 0.99 | 0.85 | 0.85 | 0.99 | 0.97 | 0.98 | 0.07   | 0.99   | 0.50  | 2.57  |
| SRL       | 0.99 | 0.93 | 0.84 | 0.99 | 1.06 | 0.97 | 0.10   | 0.98   | 0.49  | 5.52  |
| SalUN     | 0.98 | 0.90 | 0.84 | 0.98 | 1.04 | 0.97 | 0.08   | 0.96   | 0.48  | 18.04 |

Table 7.7: CIFAR-100 - ResNet18. CIFAR-100 provides the most visible comparison as there is a large gap in performance between the Retain Set and Test set, this leads to much larger RetDev scores.

|           | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| Unlearner |      |      |      |      |      |      |        |        |       |       |
| BT        | 0.98 | 0.68 | 0.54 | 0.98 | 1.25 | 0.99 | 0.27   | 0.95   | 0.48  | 9.39  |
| CF-k      | 1.00 | 0.83 | 0.56 | 1.00 | 1.53 | 1.02 | 0.55   | 0.73   | 0.63  | 5.91  |
| CFW       | 0.98 | 0.43 | 0.43 | 0.98 | 0.79 | 0.78 | 0.44   | 1.00   | 0.50  | 6.17  |
| CT        | 0.99 | 0.53 | 0.53 | 0.99 | 0.97 | 0.97 | 0.07   | 0.99   | 0.49  | 11.82 |
| EU-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FCS       | 0.98 | 0.54 | 0.55 | 0.98 | 0.99 | 1.01 | 0.04   | 0.92   | 0.54  | 3.02  |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 0.98 | 0.55 | 0.54 | 0.98 | 1.02 | 0.98 | 0.05   | 0.99   | 0.50  | 5.16  |
| GA        | 0.34 | 0.33 | 0.24 | 0.34 | 0.60 | 0.44 | 1.61   | 0.90   | 0.55  | 39.97 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| KDE       | 0.99 | 0.52 | 0.51 | 0.99 | 0.95 | 0.94 | 0.11   | 0.99   | 0.50  | 3.98  |
| MSG       | 0.91 | 0.38 | 0.38 | 0.91 | 0.69 | 0.69 | 0.71   | 1.00   | 0.50  | 4.49  |
| NG+       | 0.89 | 0.59 | 0.49 | 0.89 | 1.08 | 0.89 | 0.29   | 0.98   | 0.49  | 12.14 |
| O         | 0.98 | 0.98 | 0.56 | 0.98 | 1.81 | 1.02 | 0.85   | 0.53   | 0.73  | 1.10  |
| PRMQ      | 0.97 | 0.47 | 0.46 | 0.97 | 0.86 | 0.85 | 0.32   | 1.00   | 0.50  | 4.34  |
| R         | 1.00 | 0.55 | 0.55 | 1.00 | 1.00 | 1.00 | 0.00   | 0.99   | 0.49  | 1.00  |
| RNI       | 0.99 | 0.45 | 0.45 | 0.99 | 0.83 | 0.82 | 0.36   | 0.98   | 0.49  | 3.65  |
| SCRUB     | 0.97 | 0.50 | 0.53 | 0.97 | 0.91 | 0.96 | 0.15   | 0.98   | 0.51  | 3.81  |
| SRL       | 1.00 | 0.55 | 0.52 | 1.00 | 1.00 | 0.95 | 0.06   | 0.98   | 0.49  | 3.67  |
| SalUN     | 0.98 | 0.49 | 0.51 | 0.98 | 0.91 | 0.93 | 0.18   | 0.99   | 0.49  | 10.66 |

Table 7.8: UTKFace - ResNet18

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE    |
|-----------|------|------|------|------|------|------|--------|--------|-------|--------|
| BT        | 1.00 | 0.74 | 0.73 | 1.00 | 1.00 | 0.96 | 0.04   | 0.99   | 0.50  | 12.48  |
| CF-k      | 1.00 | 1.00 | 0.75 | 1.00 | 1.34 | 0.99 | 0.35   | 0.70   | 0.65  | 5.35   |
| CFW       | 1.00 | 0.76 | 0.76 | 1.00 | 1.02 | 1.00 | 0.02   | 1.00   | 0.50  | 5.54   |
| CT        | 1.00 | 0.75 | 0.76 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.50  | 13.34  |
| EU-k      | 0.72 | 0.61 | 0.59 | 0.72 | 0.82 | 0.77 | 0.68   | 0.99   | 0.51  | 11.42  |
| FCS       | 0.90 | 0.70 | 0.70 | 0.91 | 0.94 | 0.93 | 0.23   | 0.99   | 0.50  | 4.33   |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -      |
| FT        | 1.00 | 0.76 | 0.77 | 1.00 | 1.02 | 1.01 | 0.04   | 1.00   | 0.50  | 5.78   |
| GA        | 0.49 | 0.47 | 0.40 | 0.49 | 0.63 | 0.53 | 1.34   | 0.92   | 0.54  | 235.10 |
| IU        | 1.00 | 1.00 | 0.76 | 1.00 | 1.34 | 1.01 | 0.35   | 0.62   | 0.69  | 33.77  |
| KDE       | 0.99 | 0.79 | 0.76 | 0.99 | 1.06 | 1.00 | 0.07   | 0.97   | 0.52  | 8.19   |
| MSG       | 1.00 | 0.80 | 0.76 | 1.00 | 1.08 | 1.00 | 0.08   | 0.96   | 0.52  | 7.57   |
| NG+       | 0.94 | 0.80 | 0.72 | 0.95 | 1.07 | 0.95 | 0.18   | 0.99   | 0.51  | 6.73   |
| O         | 1.00 | 1.00 | 0.76 | 1.00 | 1.34 | 1.01 | 0.35   | 0.61   | 0.69  | 1.09   |
| PRMQ      | 0.91 | 0.72 | 0.72 | 0.91 | 0.97 | 0.95 | 0.17   | 1.00   | 0.50  | 5.88   |
| R         | 1.00 | 0.75 | 0.76 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00   |
| RNI       | 0.96 | 0.75 | 0.73 | 0.96 | 1.01 | 0.96 | 0.08   | 0.98   | 0.51  | 5.11   |
| SCRUB     | 0.80 | 0.76 | 0.69 | 0.80 | 1.01 | 0.92 | 0.29   | 0.94   | 0.53  | 4.64   |
| SRL       | 1.00 | 0.80 | 0.73 | 1.00 | 1.08 | 0.97 | 0.11   | 0.99   | 0.51  | 12.05  |
| SalUN     | 0.97 | 0.79 | 0.73 | 0.98 | 1.06 | 0.96 | 0.12   | 0.97   | 0.52  | 36.80  |

Table 7.9: MNIST - TinyViT

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE    |
|-----------|------|------|------|------|------|------|--------|--------|-------|--------|
| BT        | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 1.00   | 0.50  | 4.39   |
| CF-k      | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.51  | 123.88 |
| CFW       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 0.99   | 0.50  | 5.15   |
| CT        | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 5.92   |
| EU-k      | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.50  | 11.93  |
| FCS       | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.49  | 7.42   |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -      |
| FT        | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 7.69   |
| GA        | 0.97 | 0.97 | 0.96 | 0.97 | 0.98 | 0.97 | 0.08   | 0.99   | 0.51  | 390.99 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -      |
| KDE       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 5.35   |
| MSG       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 5.21   |
| NG+       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 0.99   | 0.50  | 2.74   |
| O         | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 1.00 | 0.01   | 0.99   | 0.51  | 0.97   |
| PRMQ      | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 6.88   |
| R         | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00   |
| RNI       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 8.03   |
| SCRUB     | 0.99 | 0.99 | 0.99 | 0.99 | 1.00 | 1.00 | 0.01   | 1.00   | 0.50  | 3.54   |
| SRL       | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 | 0.01   | 0.98   | 0.51  | 16.37  |
| SalUN     | 1.00 | 1.00 | 0.99 | 1.00 | 1.01 | 0.99 | 0.01   | 0.99   | 0.50  | 43.67  |

Table 7.10: FashionMNIST - TinyViT

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| BT        | 0.97 | 0.94 | 0.91 | 0.97 | 1.01 | 0.99 | 0.04   | 0.98   | 0.51  | 3.48  |
| CF-k      | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| CFW       | 0.99 | 0.94 | 0.92 | 0.99 | 1.01 | 1.00 | 0.02   | 0.99   | 0.51  | 5.40  |
| CT        | 0.98 | 0.92 | 0.91 | 0.98 | 0.99 | 0.99 | 0.04   | 0.99   | 0.50  | 6.00  |
| EU-k      | 0.95 | 0.94 | 0.91 | 0.95 | 1.01 | 0.99 | 0.07   | 0.97   | 0.51  | 5.34  |
| FCS       | 0.98 | 0.93 | 0.91 | 0.98 | 1.01 | 0.99 | 0.04   | 0.98   | 0.51  | 4.58  |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 0.99 | 0.94 | 0.92 | 1.00 | 1.01 | 1.00 | 0.02   | 0.98   | 0.51  | 5.12  |
| GA        | 0.92 | 0.91 | 0.85 | 0.92 | 0.99 | 0.93 | 0.17   | 0.93   | 0.53  | 50.72 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| KDE       | 1.00 | 0.94 | 0.92 | 1.00 | 1.02 | 1.00 | 0.02   | 0.98   | 0.51  | 3.38  |
| MSG       | 0.96 | 0.92 | 0.91 | 0.96 | 1.00 | 0.99 | 0.05   | 0.99   | 0.51  | 8.21  |
| NG+       | 0.97 | 0.92 | 0.91 | 0.97 | 1.00 | 0.99 | 0.05   | 0.98   | 0.51  | 13.65 |
| O         | 1.00 | 1.00 | 0.92 | 1.00 | 1.08 | 1.00 | 0.08   | 0.89   | 0.56  | 0.97  |
| PRMQ      | 0.98 | 0.94 | 0.91 | 0.98 | 1.01 | 0.99 | 0.04   | 0.98   | 0.51  | 4.67  |
| R         | 1.00 | 0.93 | 0.92 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00  |
| RNI       | 0.97 | 0.94 | 0.91 | 0.97 | 1.01 | 0.99 | 0.05   | 0.97   | 0.51  | 5.00  |
| SCRUB     | 0.96 | 0.95 | 0.91 | 0.96 | 1.03 | 0.99 | 0.08   | 0.96   | 0.52  | 9.56  |
| SRL       | 0.98 | 0.94 | 0.91 | 0.98 | 1.01 | 0.99 | 0.04   | 1.00   | 0.50  | 9.41  |
| SalUN     | 0.97 | 0.94 | 0.91 | 0.97 | 1.01 | 0.99 | 0.05   | 0.99   | 0.51  | 6.53  |

Table 7.11: CIFAR-10 - TinyViT

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE    |
|-----------|------|------|------|------|------|------|--------|--------|-------|--------|
| BT        | 0.91 | 0.91 | 0.85 | 0.91 | 1.02 | 0.97 | 0.14   | 0.99   | 0.50  | 4.13   |
| CF-k      | 0.99 | 0.89 | 0.84 | 0.99 | 1.00 | 0.95 | 0.06   | 0.96   | 0.52  | 5.18   |
| CFW       | 0.98 | 0.87 | 0.84 | 0.99 | 0.98 | 0.96 | 0.07   | 0.98   | 0.51  | 28.85  |
| CT        | 0.98 | 0.82 | 0.81 | 0.98 | 0.93 | 0.92 | 0.17   | 1.00   | 0.50  | 23.48  |
| EU-k      | 0.90 | 0.90 | 0.84 | 0.90 | 1.02 | 0.95 | 0.16   | 0.97   | 0.52  | 43.25  |
| FCS       | 0.98 | 0.84 | 0.83 | 0.98 | 0.95 | 0.94 | 0.13   | 0.99   | 0.49  | 5.15   |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -      |
| FT        | 1.00 | 0.87 | 0.84 | 1.00 | 0.98 | 0.95 | 0.07   | 0.98   | 0.51  | 5.85   |
| GA        | 0.85 | 0.85 | 0.80 | 0.85 | 0.96 | 0.91 | 0.29   | 0.97   | 0.52  | 514.77 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -      |
| KDE       | 0.97 | 0.86 | 0.84 | 0.97 | 0.97 | 0.96 | 0.11   | 0.99   | 0.50  | 5.27   |
| MSG       | 1.00 | 0.85 | 0.83 | 1.00 | 0.96 | 0.94 | 0.10   | 0.99   | 0.51  | 7.38   |
| NG+       | 0.93 | 0.86 | 0.85 | 0.93 | 0.97 | 0.96 | 0.14   | 0.99   | 0.50  | 4.10   |
| O         | 0.92 | 0.92 | 0.86 | 0.92 | 1.04 | 0.97 | 0.15   | 0.95   | 0.53  | 0.97   |
| PRMQ      | 1.00 | 0.87 | 0.84 | 1.00 | 0.98 | 0.95 | 0.07   | 0.99   | 0.51  | 4.00   |
| R         | 1.00 | 0.89 | 0.88 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00   |
| RNI       | 0.97 | 0.84 | 0.81 | 0.98 | 0.95 | 0.92 | 0.15   | 0.98   | 0.51  | 6.50   |
| SCRUB     | 1.00 | 0.84 | 0.84 | 1.00 | 0.95 | 0.95 | 0.10   | 0.99   | 0.50  | -      |
| SRL       | 0.97 | 0.88 | 0.84 | 0.97 | 0.99 | 0.96 | 0.08   | 0.99   | 0.49  | 8.52   |
| SalUN     | 0.96 | 0.89 | 0.85 | 0.96 | 1.00 | 0.96 | 0.08   | 0.99   | 0.50  | 8.30   |

Table 7.12: CIFAR-100 - TinyViT

| Unlearner | RA   | FA   | TA   | RR   | FR   | TR   | RetDev | Indisc | T-MIA | RTE   |
|-----------|------|------|------|------|------|------|--------|--------|-------|-------|
| BT        | 0.82 | 0.66 | 0.57 | 0.82 | 1.11 | 0.96 | 0.33   | 0.94   | 0.53  | 31.63 |
| CF-k      | 0.24 | 0.18 | 0.18 | 0.24 | 0.31 | 0.30 | 2.15   | 0.98   | 0.51  | 5.97  |
| CFW       | 0.98 | 0.58 | 0.56 | 0.98 | 0.97 | 0.95 | 0.10   | 0.99   | 0.51  | 9.18  |
| CT        | 0.97 | 0.55 | 0.55 | 0.98 | 0.93 | 0.93 | 0.17   | 0.99   | 0.49  | 9.80  |
| EU-k      | 0.61 | 0.60 | 0.49 | 0.61 | 1.01 | 0.81 | 0.59   | 0.90   | 0.55  | 19.29 |
| FCS       | 0.89 | 0.60 | 0.58 | 0.89 | 1.01 | 0.98 | 0.13   | 0.97   | 0.48  | 7.05  |
| FF        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| FT        | 1.00 | 0.56 | 0.55 | 1.00 | 0.94 | 0.91 | 0.15   | 1.00   | 0.50  | 5.90  |
| GA        | 0.60 | 0.58 | 0.46 | 0.60 | 0.97 | 0.77 | 0.66   | 0.87   | 0.56  | 91.91 |
| IU        | -    | -    | -    | -    | -    | -    | -      | -      | -     | -     |
| KDE       | 0.93 | 0.58 | 0.57 | 0.94 | 0.98 | 0.96 | 0.13   | 0.99   | 0.50  | 4.03  |
| MSG       | 0.97 | 0.57 | 0.56 | 0.97 | 0.95 | 0.93 | 0.15   | 1.00   | 0.50  | 5.89  |
| NG+       | 0.84 | 0.57 | 0.55 | 0.84 | 0.95 | 0.92 | 0.29   | 0.96   | 0.52  | 3.01  |
| O         | 0.87 | 0.87 | 0.61 | 0.87 | 1.46 | 1.02 | 0.61   | 0.74   | 0.63  | 0.98  |
| PRMQ      | 0.95 | 0.62 | 0.57 | 0.95 | 1.03 | 0.96 | 0.13   | 0.95   | 0.52  | 5.58  |
| R         | 1.00 | 0.60 | 0.60 | 1.00 | 1.00 | 1.00 | 0.00   | 1.00   | 0.50  | 1.00  |
| RNI       | 0.85 | 0.52 | 0.51 | 0.85 | 0.87 | 0.85 | 0.42   | 0.99   | 0.50  | 5.38  |
| SCRUB     | 0.77 | 0.64 | 0.57 | 0.77 | 1.08 | 0.96 | 0.35   | 0.93   | 0.53  | 6.17  |
| SRL       | 0.98 | 0.57 | 0.57 | 0.98 | 0.96 | 0.96 | 0.11   | 0.97   | 0.49  | 5.92  |
| SalUN     | 0.97 | 0.58 | 0.57 | 0.97 | 0.97 | 0.96 | 0.10   | 0.98   | 0.49  | 7.03  |

### 7.4.3 Per architectures rankings

Here, we present the rankings across datasets for ResNet18 (Table 7.13) and TinyVit (Table 7.14). Some methods, such as RNI or NG+, are less efficient regarding indiscernibility on the ViT architectures. However, methods such as SCRUB are less efficient regarding Retention Deviation on the ViT architecture.

Table 7.13: Ranking on ResNet

| Rank | Method | Retention Deviation |    |    |   | Rank | Method | Indiscernibility |    |    |   |
|------|--------|---------------------|----|----|---|------|--------|------------------|----|----|---|
|      |        | G1                  | G2 | G3 | F |      |        | G1               | G2 | G3 | F |
| 1    | FT     | 5                   | 0  | 0  | 0 | 1    | CFW    | 5                | 0  | 0  | 0 |
| 2    | FCS    | 4                   | 1  | 0  | 0 | 1    | CT     | 5                | 0  | 0  | 0 |
| 2    | MSG    | 4                   | 1  | 0  | 0 | 1    | MSG    | 5                | 0  | 0  | 0 |
| 3    | CT     | 4                   | 0  | 1  | 0 | 1    | RNI    | 5                | 0  | 0  | 0 |
| 3    | KDE    | 4                   | 0  | 1  | 0 | 2    | FT     | 4                | 1  | 0  | 0 |
| 4    | NG+    | 3                   | 2  | 0  | 0 | 2    | KDE    | 4                | 1  | 0  | 0 |
| 4    | PRMQ   | 3                   | 2  | 0  | 0 | 2    | NG+    | 4                | 1  | 0  | 0 |
| 4    | SalUN  | 3                   | 2  | 0  | 0 | 2    | PRMQ   | 4                | 1  | 0  | 0 |
| 5    | CFW    | 3                   | 1  | 1  | 0 | 3    | FCS    | 3                | 2  | 0  | 0 |
| 6    | SCRUB  | 3                   | 0  | 1  | 1 | 3    | SRL    | 3                | 2  | 0  | 0 |
| 7    | SRL    | 2                   | 3  | 0  | 0 | 3    | SalUN  | 3                | 2  | 0  | 0 |
| 8    | BT     | 1                   | 4  | 0  | 0 | 4    | SCRUB  | 3                | 1  | 0  | 1 |
| 8    | RNI    | 1                   | 4  | 0  | 0 | 5    | BT     | 2                | 3  | 0  | 0 |
| 9    | CF-k   | 1                   | 2  | 1  | 1 | 6    | EU-k   | 1                | 0  | 0  | 4 |
| 10   | IU     | 1                   | 0  | 2  | 2 | 7    | GA     | 0                | 3  | 1  | 1 |
| 11   | EU-k   | 0                   | 1  | 0  | 4 | 8    | CF-k   | 0                | 1  | 3  | 1 |
| 12   | GA     | 0                   | 0  | 4  | 1 | 9    | IU     | 0                | 0  | 3  | 2 |
| 13   | FF     | 0                   | 0  | 0  | 5 | 10   | FF     | 0                | 0  | 0  | 5 |

### 7.4.4 L2 Distances between model weights

The distance between the Unlearned and Retrained models has also been considered in the literature to evaluate MU [105]. Nevertheless, we observe that models end up at a similar distance to the Retrained model, with significant differences in performance. We further note that one challenging aspect of the L2 distance comparison is the different factors of Weight Decay used by the MU method. The hyper-parameter searches determine these Weight Decay factors, which can significantly vary from one unlearning method to another, making it challenging to compare methods. Furthermore, the best-performing method, MSG, is usually at the same

Table 7.14: Ranking of ViT

| Rank | Method | Retention Deviation |    |    |   | Rank | Method | Indiscernibility |    |    |   |
|------|--------|---------------------|----|----|---|------|--------|------------------|----|----|---|
|      |        | G1                  | G2 | G3 | F |      |        | G1               | G2 | G3 | F |
| 1    | CFW    | 4                   | 0  | 0  | 0 | 1    | CT     | 4                | 0  | 0  | 0 |
| 1    | MSG    | 4                   | 0  | 0  | 0 | 1    | MSG    | 4                | 0  | 0  | 0 |
| 1    | PRMQ   | 4                   | 0  | 0  | 0 | 2    | KDE    | 3                | 1  | 0  | 0 |
| 2    | CT     | 3                   | 1  | 0  | 0 | 2    | SalUN  | 3                | 1  | 0  | 0 |
| 2    | FT     | 3                   | 1  | 0  | 0 | 3    | SRL    | 3                | 0  | 1  | 0 |
| 2    | KDE    | 3                   | 1  | 0  | 0 | 4    | BT     | 2                | 2  | 0  | 0 |
| 2    | SRL    | 3                   | 1  | 0  | 0 | 4    | CFW    | 2                | 2  | 0  | 0 |
| 2    | SalUN  | 3                   | 1  | 0  | 0 | 4    | FCS    | 2                | 2  | 0  | 0 |
| 3    | FCS    | 2                   | 2  | 0  | 0 | 4    | FT     | 2                | 2  | 0  | 0 |
| 3    | NG+    | 2                   | 2  | 0  | 0 | 4    | PRMQ   | 2                | 2  | 0  | 0 |
| 4    | BT     | 1                   | 3  | 0  | 0 | 4    | RNI    | 2                | 2  | 0  | 0 |
| 4    | RNI    | 1                   | 3  | 0  | 0 | 4    | SCRUB  | 2                | 2  | 0  | 0 |
| 4    | SCRUB  | 1                   | 3  | 0  | 0 | 5    | NG+    | 1                | 3  | 0  | 0 |
| 5    | CF-k   | 1                   | 1  | 1  | 1 | 6    | CF-k   | 1                | 1  | 1  | 1 |
| 6    | EU-k   | 0                   | 4  | 0  | 0 | 7    | EU-k   | 0                | 2  | 2  | 0 |
| 7    | GA     | 0                   | 1  | 3  | 0 | 8    | GA     | 0                | 1  | 3  | 0 |
| 8    | FF     | 0                   | 0  | 0  | 4 | 9    | FF     | 0                | 0  | 0  | 4 |
| 8    | IU     | 0                   | 0  | 0  | 4 | 9    | IU     | 0                | 0  | 0  | 4 |

distance as both the Original and Retrained model. For each method, for each initialization seed, we computed the L2 distance between the unlearned model  $f_U$  and the retrained model  $f_R$ , as well as between the  $f_U$  and  $f_O$  (Figure 7.2).

Although having the same weight as the Retrained model would indicate that the unlearned model has unlearned  $\mathcal{D}_F$ , our evaluations show that distance to the Retrained model might not be an adequate evaluation metric for MU.

## 7.5 Discussion

### 7.5.1 Conclusion

The increasing focus on data privacy and the trustworthiness of machine learning models underscores the need for robust and practical methods to unlearn and remove the influence of specific data from trained models. Due to the growing size of models, we require methods that avoid computationally costly retraining from scratch. In this work, we comprehensively

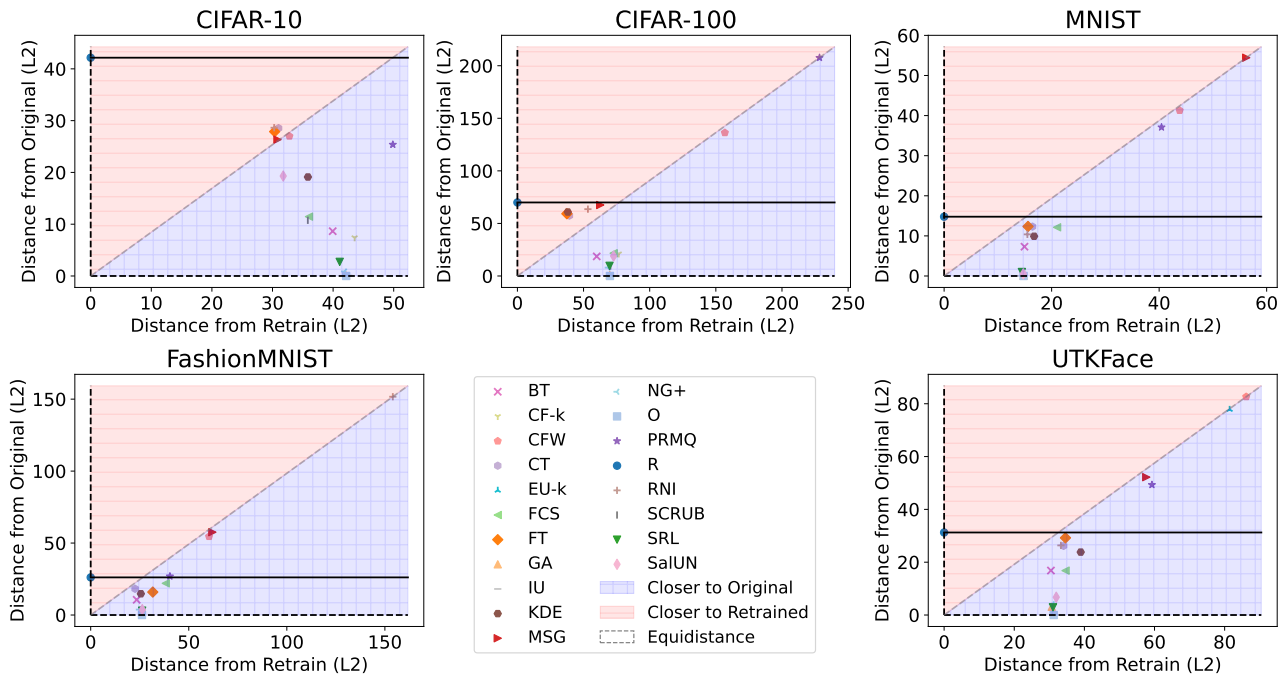


Figure 7.2: L2 Distance between the Unlearned ResNet18 models, the Original and Retrained models. None of the unlearned models gets close to the Retrained model’s weights; most unlearned Models are closer to the Original model than the Retrained model.

compared approximate unlearning methods across various models and datasets to address this critical issue.

We experimentally compared 18 methods across different datasets and architectures, focusing on assessing each method’s ability to maintain privacy and accuracy while being computationally efficient and reliable across datasets, architectures, and random seeds. Our findings indicate that Masked-Small-Gradients (MSG), which accumulates gradients via gradient descent on the data to remember and gradient ascent on the data to forget to determine which weights to update, consistently outperforms all metrics across the studied datasets, architectures, and initialization seeds. Similarly, Convolution Transpose (CT) performed strongly, which leverages the simple transposition in convolutional layers.

CT and MSG were resistant to a population-based Membership Inference Attack (MIA) and a stronger per-sample attack (U-LiRA). However, a core challenge of approximate unlearning is that these methods will only be as strong as the attacks against which they are tested. As stronger and more complex attacks emerge, some approximate unlearning methods might be less efficient than initially expected. This highlights the need for continuous evaluation and adaptation of unlearning methods to maintain their effectiveness.

### 7.5.2 Limitations

Due to computational costs, we limited our analysis to Tiny Vision Transformers and ResNet. Further investigation of other architectures could provide valuable insights. We did not investigate different amounts of unlearning samples, which some methods are known to be sensitive to [119]. We did not consider repeated deletion; instead, we assume that there is a single forget set and that the unlearning process happens once, as is common in the literature. Nonetheless, in practical applications, one might need to unlearn different smaller forget sets over time, and some unlearning methods might not work as well under such a scenario. We finally remark once again on the difficulty of evaluating for approximate unlearning [93]: while these methods provide significant gains in efficiency, novel attacks might highlight yet unknown weaknesses of the unlearning processes.

### 7.5.3 Perspectives

First, we focus on natural image data. However, machine unlearning is relevant to other data types, such as medical images or other modalities, such as time series, audio, and speech, or language data. Second, we focus on the classification task; however, other learning tasks would also greatly benefit from machine unlearning. For example, removing concepts from generative models for images [66] or poisoned data in language models [93]. Third, this work focuses on empirically benchmarking approximate machine unlearning methods. We do not provide a theoretical analysis of these methods or a rigorous comparison with exact unlearning algorithms.

## 7.6 Requirements

We ran the experiments on compute clusters with different capacities. Nonetheless, each method was tested on devices with the exact specifications when recording run times: 1 NVIDIA L4 24GB GPU and 4 Intel(R) Xeon(R) CPU @ 2.20GHz.

# Chapter 8

## Contributions and Perspectives

This chapter summarizes the work conducted in this thesis and the contributions and perspectives for future work.

### 8.1 Conclusion and Contributions

My contributions are the development of two analysis tools Feature Impact Balance (FIB) and Min-Max Relative Change Quadrant (MMRCQ), the analysis of the impact of data augmentations across different classes, the evaluation of the transferability of foundation models from healthy speech to speech from patients with dysarthria, the evaluation of self-supervised representations to missing information and the benchmark of state-of-the-art machine unlearning methods.

I explored various aspects of machine learning models through representation learning, robustness to incomplete data, and privacy via machine unlearning. This work covered four aspects of Machine Learning:

- Data Augmentations: data augmentations impact classes differently and, therefore, should be selected with care.
- Transferability: foundation models trained on healthy speech can be leveraged for automated dysarthria assessments.

- Self-Supervised Learning: representations learned through SSL can lead to high accuracy even when the object to classify is missing. Therefore, additional consideration should be given to the evaluation of the SSL model and dataset biases.
- Machine Unlearning: machine unlearning provides tools to remove the influence of specific data points from a trained model. Its further development could create opportunities for applications such as complying with privacy regulations, reducing biases, lowering costs, and eliminating undesirable patterns.

### 8.1.1 Comparison tools: Feature Impact Balance (FIB) and Min-Max Relative Change Quadrant (MMRCQ)

In Chapter 3, I introduced the Feature Impact Balance (FIB) score and the Min-Max Relative Change Quadrant (MMRCQ) plot to complement traditional error metrics such as the Mean Squared Error (MSE). These tools allow for a more detailed analysis of errors distributed across vectors, allowing for more nuanced error detection and comparison. These tools are flexible in design and could be further applied to analyze biases, fairness, class imbalance, and error evaluation in classification and regression tasks.

### 8.1.2 Class-level data augmentation impact

In Chapter 4, I leveraged the tools introduced in Chapter 3 to evaluate the impact of data augmentation on individual classes. I showed that while data augmentations are commonly applied irrespective of the classes, some augmentations impact classes more than others. Although data augmentations are extensively used in the current deep-learning ecosystem, further research is required to understand their impact on the training process. While class-aware data augmentations exist, these methods are usually associated with generative models or address class imbalance rather than determining which augmentations are detrimental to specific classes.

### **8.1.3 Analysis of the transferability of healthy speech foundation models for automated dysarthria assessments**

In Chapter 5, I showed that one could use models trained on large-scale non-medical speech data to perform automatic dysarthria assessment. This transferability helps against data scarcity in the medical domain. By leveraging foundation models, this work exemplified how representations learned from non-medical domains can still be valuable for specialized medical tasks. Furthermore, we developed a visualization tool to analyze per-speaker speech impairments, making a practical contribution and providing an accessible way to visualize model outputs.

### **8.1.4 Evaluation of the robustness of Self-Supervised visual feature extractors to corrupted inputs**

In Chapter 6, I evaluated the robustness of self-supervised learning (SSL) representations when given incomplete or corrupted data inputs. I used segmentation masks to simulate missing data. This chapter provided insights into how SSL models perform when the object of interest for a downstream task is absent from the input. We showed that the characteristics of the datasets were important as it was possible to achieve high classification accuracy even without the object of interest in the picture. This could be ascribed to a potential lack of variation in the background of existing datasets and the ability of the different self-supervised models to extract contextual information sufficient to make predictions even in the absence of the object of interest.

### **8.1.5 Evaluation of the Machine Unlearning methods across datasets, initializations**

Finally, Chapter 7 addressed the need for machine unlearning, a mechanism for removing the influence of specific data points from trained models to ensure privacy and compliance with regulations such as the "right to be forgotten." We benchmarked 18 machine unlearning methods from the literature and an unlearning competition. We evaluated different unlearning methods and insights into unlearning strategies based on privacy, computational efficiency, and

performance retention. The need for reliable machine unlearning is increasing due to privacy concerns and potential compliance with data regulatory bodies. Still, it could also provide solutions to correct undesirable patterns from trained models, such as removing copyrighted content from trained models and addressing biases.

## 8.2 Insights toward the application of Artificial Intelligence in Healthcare

Various aspects hinder the development and deployment of artificial intelligence in healthcare, one of which is the scarcity of data when building models. This can be ascribed to challenges, such as the sensitivity of the data, difficulty in collecting and annotating, and heterogeneity between patients and forms of different ailments. Medical data can contain sensitive information, such as genetic markers or whether the patient has a specific disease, making it difficult to share such data publicly. Collecting and annotating medical data can be challenging as it can require specialized equipment, such as MRI and PET scanners, and domain knowledge from trained experts, such as identifying lesions on brain scans, identifying specific patterns in speech data, or conducting cognitive tests. Transfer learning is another promising approach for addressing data scarcity. By leveraging models pre-trained on large natural image datasets, researchers can adapt these models for medical tasks [130, 148]. Similarly, self-supervised learning (SSL) offers opportunities to utilize large-scale unlabeled datasets. SSL can help integrate data from various studies within the same modality—for example, brain scans from different cohorts—facilitating the learning of general representations that capture diverse conditions and settings. These representations can then be applied to multiple downstream tasks. However, further research is needed to assess the reliability and robustness of such SSL models (Chapter 6). Advances in Machine Unlearning and the ability to selectively remove information from trained models help ensure that information extractable from trained models cannot be used to infer specific sensitive properties from the model or reconstruct the training set, mitigating the risks associated with sharing trained models.

## 8.3 Perspectives

One could build on top of this thesis’s contributions in the following directions:

### 8.3.1 Feature Impact Balance and Min-Max Relative Change Quadrant

One could leverage the Feature Impact Balance (FIB) score and MMRCQ plot to analyze various feature extractors’ feature space on contribution vectors associated with explainable AI or to evaluate biases. The MMRCQ could monitor changes in longitudinal data in domains such as finance, healthcare, or climate.

### 8.3.2 Data Augmentations

Further research is required to understand the impact of data augmentation on the training process and on a subset of classes. There is renewed interest in the effect of regularization methods and their impact on specific classes [7]. Furthermore, given the ubiquitous applications of data augmentation in deep learning pipelines, a better understanding of their effects on biases is desirable [110, 144].

### 8.3.3 Dysarthria Automated Assessments

Self-Supervised Learning-based feature extraction open new opportunities for tasks that suffer from data scarcity such as Automated Dysarthria Assessments. Further work on the adaptability and robustness of such pre-trained Self-Supervised models could further improve performance [101] and lead to more reliable automated assessments. Additionally, other approaches such as designing data augmentations [107, 125] could further help alleviate data scarcity.

### 8.3.4 Self-Supervised Learning Robustness

The evaluation of the Self-Supervised model often relies on their assessment of downstream tasks; as such, additional attention should be paid to the datasets used in their assessment and their potential biases [151, 126, 128].

### 8.3.5 Machine Unlearning

Finally, further evaluations in machine unlearning are necessary to ensure the reliability and practicality of unlearning methods for real-world deployment [93]. In this work, I focused on imaging data, while unlearning can be extended to other domains, such as removing content used to train large language models or inhibiting content generation [66].

# Bibliography

- [1] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, October 2016. arXiv:1607.00133 [cs, stat].
- [2] Bassam Ali Al-Qatab and Muntaz Begum Mustafa. Classification of Dysarthric Speech According to the Severity of Impairment: an Analysis of Acoustic Features. *IEEE Access*, 9:18183–18194, 2021. Conference Name: IEEE Access.
- [3] Relja Arandjelović and Andrew Zisserman. Look, Listen and Learn, May 2017.
- [4] Relja Arandjelović and Andrew Zisserman. Objects that Sound. *arXiv:1712.06651 [cs, eess]*, July 2018. arXiv: 1712.06651.
- [5] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242. PMLR, July 2017. ISSN: 2640-3498.
- [6] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, October 2020. arXiv:2006.11477 [cs, eess].
- [7] Randall Balestriero, Leon Bottou, and Yann LeCun. The Effects of Regularization and Data Augmentation are Class Dependent. October 2022.
- [8] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi

- Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A Cookbook of Self-Supervised Learning, April 2023. arXiv:2304.12210 [cs].
- [9] Ms. Aayushi Bansal, Dr. Rewa Sharma, and Dr. Mamta Kathuria. A Systematic Review on Data Scarcity Problem in Deep Learning: Solution and Applications. *ACM Computing Surveys*, 54(10s):1–29, January 2022.
- [10] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers, September 2022. arXiv:2106.08254 [cs].
- [11] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. *arXiv:2105.04906 [cs]*, January 2022. arXiv:2105.04906.
- [12] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence Functions in Deep Learning Are Fragile, February 2021. arXiv:2006.14651 [cs, stat].
- [13] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and D. A. Forsyth. Unrestricted Adversarial Examples via Semantic Manipulation, March 2020. arXiv:1904.06347 [cs].
- [14] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.
- [15] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele

- Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models, July 2022. arXiv:2108.07258 [cs].
- [16] Florian Bordes, Samuel Lavoie, Randall Balestriero, Nicolas Ballas, and Pascal Vincent. A surprisingly simple technique to control the pretraining bias for better transfer: Expand or Narrow your representation, April 2023. arXiv:2304.05369 [cs].
- [17] Alexei Botchkarev. Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019. arXiv:1809.03006 [cs, stat].
- [18] Aleksander Botev, Matthias Bauer, and Soham De. Regularising for invariance to data augmentation improves supervised learning, March 2022. arXiv:2203.03304 [cs, stat].
- [19] Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159, May 2021. ISSN: 2375-1207.
- [20] Xavier F. Cadet, Sara Ahmadi-Abhari, and Hamed Haddadi. FIB: A Method for Evaluation of Feature Impact Balance in Multi-Dimensional Data, July 2022. arXiv:2207.04500 [cs].
- [21] Xavier F. Cadet, Ranya Aloufi, Sara Ahmadi-Abhari, and Hamed Haddadi. A study on the impact of self-supervised learning on automatic dysarthric speech assessment. In *2024 IEEE international conference on acoustics, speech, and signal processing workshops (ICASSPW)*, pages 630–634, 2024.

- [22] Xavier F. Cadet, Ranya Aloufi, Alain Miranville, Sara Ahmadi-Abhari, and Hamed Haddadi. Evaluating The Robustness of Self-Supervised Representations to Background/Foreground Removal, June 2023. arXiv:2306.01398 [cs].
- [23] Xavier F. Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep Unlearn: Benchmarking Machine Unlearning, October 2024. arXiv:2410.01276 [cs].
- [24] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.
- [25] Yinzhi Cao and Junfeng Yang. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, San Jose, CA, May 2015. IEEE.
- [26] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, San Francisco, CA, USA, May 2022. IEEE.
- [27] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, July 2019. arXiv:1802.08232 [cs].
- [28] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features, March 2019. arXiv:1807.05520 [cs].
- [29] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv:2006.09882 [cs]*, January 2021. arXiv: 2006.09882.
- [30] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers, May 2021. arXiv:2104.14294 [cs].
- [31] Guoguo Chen, Carolina Parada, and Georg Heigold. Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4087–4091, May 2014. ISSN: 2379-190X.

- [32] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1691–1703. PMLR, November 2020. ISSN: 2640-3498.
- [33] Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. Fast Model DeBias with Machine Unlearning. *Advances in Neural Information Processing Systems*, 36:14516–14539, December 2023.
- [34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv:2002.05709 [cs, stat]*, June 2020. arXiv: 2002.05709.
- [35] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. *arXiv:2011.10566 [cs]*, November 2020. arXiv: 2011.10566.
- [36] Yanbei Chen, Massimiliano Mancini, Xiatian Zhu, and Zeynep Akata. Semi-Supervised and Unsupervised Deep Visual Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–23, 2022.
- [37] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks, January 2015. arXiv:1412.0233 [cs].
- [38] Alexander Chowdhury, Jacob Rosenthal, Jonathan Waring, and Renato Umeton. Applying Self-Supervised Learning to Medicine: Review of the State of the Art and Medical Implementations. *Informatics*, 8(3):59, September 2021. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [39] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks using an Incompetent Teacher, May 2023. arXiv:2205.08096 [cs].
- [40] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-Shot Machine Unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

- [41] NeurIPS 2023 Machine Unlearning competition 1st-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458721>, 2023. accessed on january 31, 2024.
- [42] NeurIPS 2023 Machine Unlearning competition 2nd-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459200>, 2023.
- [43] NeurIPS 2023 Machine Unlearning competition 3rd-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459334>, 2023.
- [44] NeurIPS 2023 Machine Unlearning competition 4th-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459148>, 2023.
- [45] NeurIPS 2023 Machine Unlearning competition 5th-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458531>, 2023.
- [46] NeurIPS 2023 Machine Unlearning competition 6th-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458740>, 2023.
- [47] NeurIPS 2023 Machine Unlearning competition 7th-place team. <https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459095>, 2023.
- [48] R. Dennis Cook and Sanford Weisberg. *Residuals and Influence in Regression*. New York: Chapman and Hall, 1982.
- [49] Thomas H. Cormen, Charles Eric Leiserson, Ronald Linn Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT Press, Cambridge, Massachusetts London, fourth edition edition, 2022.
- [50] Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuexing Liu, Xiaoxue Long, Yang Wen, Lina Lu, Yaxin Shen, Yan Chen, Dinggang Shen, Xiaokang Yang, Haidong Zou, Bin Sheng, and Weiping Jia. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature Communications*, 12(1):3242, May 2021. Publisher: Nature Publishing Group.
- [51] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 1 edition, February 2020.

- [52] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. page 8, 2009.
- [53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [54] Terrance DeVries and Graham W. Taylor. Dataset Augmentation in Feature Space, February 2017. arXiv:1702.05538 [cs, stat].
- [55] Rohan Raju Dhanakshirur, Mrinal Tyagi, Britty Baby, Ashish Suri, Prem Kalra, and Chetan Arora. VideoCutMix: Temporal Segmentation of Surgical Videos in Scarce Data Scenarios. In Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume 15006, pages 725–735. Springer Nature Switzerland, Cham, 2024. Series Title: Lecture Notes in Computer Science.
- [56] Sander Dieleman, Kyle W. Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459, June 2015. arXiv:1503.07077 [astro-ph, stat].
- [57] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised Visual Representation Learning by Context Prediction, January 2016. arXiv:1505.05192 [cs].
- [58] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, June 2021. arXiv: 2010.11929.
- [59] Joseph R. Duffy. *Motor speech disorders: substrates, differential diagnosis, and management*. Mosby, St. Louis, Mo, 3. ed edition, 2013.
- [60] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

- [61] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations. December 2018.
- [62] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, February 2017. Number: 7639 Publisher: Nature Publishing Group.
- [63] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016. tex.added-at: 2020-08-20T11:33:21.000+0200 tex.interhash: c7b667cac6031282160a9e94d5a118f8 tex.intrahash: 43a2175512dc8b9d8855fa7a763cdc3e tex.timestamp: 2020-08-20T11:33:21.000+0200.
- [64] Tiago H. Falk, Wai-Yip Chan, and Fraser Shein. Characterization of atypical vocal source excitation, temporal dynamics and prosody for objective measurement of dysarthric word intelligibility. *Speech Communication*, 54(5):622–631, June 2012.
- [65] Chongyu Fan, Jiancheng Liu, Alfred Hero, and Sijia Liu. Challenging Forgets: Unveiling the Worst-Case Forget Sets in Machine Unlearning, July 2024. arXiv:2403.07362.
- [66] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation, March 2024. arXiv:2310.12508 [cs].
- [67] Vitaly Feldman. Does Learning Require Memorization? A Short Tale about a Long Tail, January 2021. arXiv:1906.05271 [cs, stat].
- [68] Vitaly Feldman and Chiyuan Zhang. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation, August 2020. arXiv:2008.03703 [cs, stat].
- [69] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. Language, speech, and communication. MIT Press, Cambridge, Mass, 1998.

- [70] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12043–12051, March 2024.
- [71] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, Denver Colorado USA, October 2015. ACM.
- [72] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. *Proceedings of the ... USENIX Security Symposium. UNIX Security Symposium*, 2014:17–32, August 2014.
- [73] Weifeng Fu. Application of an Isolated Word Speech Recognition System in the Field of Mental Health Consultation: Development and Usability Study. *JMIR medical informatics*, 8(6):e18677, June 2020.
- [74] Shungo Fujii, Yasunori Ishii, Kazuki Kozuka, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Data Augmentation by Selecting Mixed Classes Considering Distance Between Classes, September 2022. arXiv:2209.05122 [cs, stat].
- [75] Gautam Rajendrakumar Gare and John Michael Galeotti. Exploiting Class Similarity for Machine Learning with Confidence Labels and Projective Loss Functions, March 2021. arXiv:2103.13607 [cs, eess, stat].
- [76] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style, September 2015. arXiv:1508.06576.
- [77] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. *arXiv:1803.07728 [cs]*, March 2018. arXiv:1803.07728.
- [78] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

- [79] Simon J. Godsill and Peter J. W. Rayner. *Digital Audio Restoration*. Springer London, London, 1998.
- [80] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards Adversarial Evaluations for Inexact Machine Unlearning, February 2023. arXiv:2201.06640 [cs].
- [81] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-Privacy Forgetting in Deep Networks, June 2021. arXiv:2012.13431 [cs].
- [82] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks, March 2020. arXiv:1911.04933 [cs, stat].
- [83] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observations, October 2020. arXiv:2003.02960 [cs, math, stat].
- [84] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, 2016.
- [85] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac Machine Learning, October 2020. arXiv:2010.10981 [cs].
- [86] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A Kernel Statistical Test of Independence. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [87] Jean-Bastien Grill, Florian Strub, Florent Althé, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv:2006.07733 [cs, stat]*, September 2020. arXiv: 2006.07733.
- [88] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiuotė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying Large Language Model Generalization with Influence Functions, August 2023. arXiv:2308.03296 [cs, stat].

- [89] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified Data Removal from Machine Learning Models, August 2020. arXiv:1911.03030 [cs, stat].
- [90] Kyle Hall, Andy Huang, and Seyed Reza Shahamiri. An Investigation to Identify Optimal Setup for Automated Assessment of Dysarthric Intelligibility using Deep Learning Technologies. *Cognitive Computation*, 15(1):146–158, January 2023.
- [91] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12):2639–2664, 2004. Place: US Publisher: MIT Press.
- [92] Trevor J. Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, New York, 2nd ed edition, 2009.
- [93] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy, March 2024. arXiv:2403.01218 [cs].
- [94] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, New Orleans, LA, USA, June 2022. IEEE.
- [95] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning, March 2020. arXiv:1911.05722 [cs].
- [96] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385.
- [97] Hossein Hosseini and Radha Poovendran. Semantic Adversarial Examples, March 2018. arXiv:1804.00499 [cs].
- [98] H. Hotelling. RELATIONS BETWEEN TWO SETS OF VARIATES. *Biometrika*, 28(3-4):321–377, December 1936.
- [99] Lei Hsiung, Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Towards Compositional Adversarial Robustness: Generalizing Adversarial Training to Composite Semantic Perturbations, November 2022. arXiv:2202.04235 [cs].

- [100] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *arXiv:2106.07447 [cs, eess]*, June 2021. arXiv: 2106.07447.
- [101] Shujie Hu, Xurong Xie, Mengzhe Geng, Zengrui Jin, Jiajun Deng, Guinan Li, Yi Wang, Mingyu Cui, Tianzi Wang, Helen Meng, and Xunying Liu. Self-Supervised ASR Models and Features for Dysarthric and Elderly Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3561–3575, 2024. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.
- [102] Andy Huang, Kyle Hall, Catherine Watson, and Seyed Reza Shahamiri. A Review of Automated Intelligibility Assessment for Dysarthric Speakers. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 19–24, October 2021.
- [103] Yushan Huang, Ranya Aloufi, Xavier Cadet, Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. Low-Energy On-Device Personalization for MCUs, October 2024. arXiv:2403.08040 [cs].
- [104] Hiroshi Inoue. Data Augmentation by Pairing Samples for Images Classification, April 2018. arXiv:1801.02929 [cs, stat].
- [105] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate Data Deletion from Machine Learning Models, February 2021. arXiv:2002.10077 [cs, stat].
- [106] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model Sparsity Can Simplify Machine Unlearning, January 2024. arXiv:2304.04934 [cs].
- [107] Yishan Jiao, Ming Tu, Visar Berisha, and Julie Liss. Simulating dysarthric speech for training data augmentation in clinical speech applications, April 2018. arXiv:1804.10325.
- [108] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, February 2019. arXiv:1902.06162 [cs].

- [109] Amlu Anna Joshy and Rajeev Rajan. Automated Dysarthria Severity Classification: A Study on Acoustic Features and Deep Learning Techniques. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 30:1147–1157, 2022.
- [110] Lamin Juwara, Alaa El-Hussuna, and Khaled El Emam. An evaluation of synthetic data augmentation for mitigating covariate bias in health data. *Patterns*, 5(4):100946, April 2024.
- [111] Heejin Kim, Mark Hasegawa-Johnson, Adrienne Perlman, Jon Gunderson, Thomas Huang, Kenneth Lloyd Watkin, and Simone Frame. Dysarthric speech database for universal access research: INTERSPEECH 2008 - 9th Annual Conference of the International Speech Communication Association. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1741–1744, 2008.
- [112] Yunjung Kim, Raymond D. Kent, and Gary Weismer. An acoustic study of the relationships among neurologic disease, dysarthria type, and severity of dysarthria. *Journal of speech, language, and hearing research: JSLHR*, 54(2):417–429, April 2011.
- [113] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything, April 2023. arXiv:2304.02643 [cs].
- [114] Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of Neural Network Models: A Survey of Functional and Representational Measures, August 2024. arXiv:2305.06329 [cs].
- [115] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions, March 2017. arXiv:1703.04730 [cs, stat] version: 1.
- [116] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3519–3529. PMLR, May 2019. ISSN: 2640-3498.
- [117] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 2009.

- [118] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [119] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards Unbounded Machine Unlearning, October 2023. arXiv:2302.09880 [cs].
- [120] Aarre Laakso and Garrison Cottrell. Content and cluster analysis: Assessing representational similarity in neural systems. *Philosophical Psychology*, 13(1):47–76, March 2000. Publisher: Routledge \_eprint: <https://doi.org/10.1080/09515080050002726>.
- [121] Colin Lea, Zifang Huang, Lauren Tooley, Jaya Narain, Dianna Yee, Panayiotis Georgiou, Tien Dung Tran, Jeffrey P. Bigham, and Leah Findlater. From User Perceptions to Technical Improvement: Enabling People Who Stutter to Better Use Speech Recognition, February 2023.
- [122] Colin Lea, Dianna Yee, Jaya Narain, Zifang Huang, Lauren Tooley, Jeffrey P. Bigham, and Leah Findlater. Latent Phrase Matching for Dysarthric Speech, June 2023.
- [123] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, December 1989.
- [124] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. Conference Name: Proceedings of the IEEE.
- [125] Wing-Zin Leung, Mattias Cross, Anton Ragni, and Stefan Goetze. Training Data Augmentation for Dysarthric Automatic Speech Recognition by Text-to-Dysarthric-Speech Synthesis, June 2024. arXiv:2406.08568.
- [126] Yi Li and Nuno Vasconcelos. REPAIR: Removing Representation Bias by Dataset Resampling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9564–9573, Long Beach, CA, USA, June 2019. IEEE.
- [127] Haohe Liu, Qiuqiang Kong, Qiao Tian, Yan Zhao, DeLiang Wang, Chuanzeng Huang, and Yuxuan Wang. VoiceFixer: Toward General Speech Restoration with Neural Vocoder, October 2021. arXiv:2109.13731 [cs, eess].

- [128] Alexandra Sasha Luccioni and David Rolnick. Bugs in the Data: How ImageNet Misrepresents Biodiversity, August 2022. arXiv:2208.11695 [cs].
- [129] Qiang Ma, Liu Li, Emma C. Robinson, Bernhard Kainz, and Daniel Rueckert. Weakly Supervised Learning of Cortical Surface Reconstruction from Segmentations. In Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume 15011, pages 766–777. Springer Nature Switzerland, Cham, 2024. Series Title: Lecture Notes in Computer Science.
- [130] Christos Matsoukas, Johan Fredin Haslum, Moein Sorkhei, Magnus Soderberg, and Kevin Smith. What Makes Transfer Learning Work for Medical Images: Feature Reuse & Other Factors. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9215–9224, New Orleans, LA, USA, June 2022. IEEE.
- [131] Viviana Mendoza Ramos, Charlotte Paulyn, Leen Van den Steen, Maria E. Hernandez-Diaz Huici, Marc De Bodt, and Gwen Van Nuffelen. Effect of boost articulation therapy (BArT) on intelligibility in adults with dysarthria. *International Journal of Language & Communication Disorders*, 56(2):271–282, March 2021.
- [132] X. Menendez-Pidal, J.B. Polikoff, S.M. Peters, J.E. Leonzio, and H.T. Bunnell. The Nemours database of dysarthric speech. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 3, pages 1962–1965 vol.3, October 1996.
- [133] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, May 2018.
- [134] Claire Mitchell, Matthew Gittins, Sarah Tyson, Andy Vail, Paul Conroy, Lizz Paley, and Audrey Bowen. Prevalence of aphasia and dysarthria among inpatient stroke survivors: describing the population, therapy provision and outcomes on discharge. *Aphasiology*, 35(7):950–960, July 2021.
- [135] Tom M. Mitchell. *Machine Learning*. McGraw-Hill series in computer science. McGraw-Hill, New York, 1997.

- [136] Ari S. Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation, October 2018. arXiv:1806.05759 [cs, stat].
- [137] Francisco J. Moreno-Barea, Fiammetta Strazzer, José M. Jerez, Daniel Urda, and Leonardo Franco. Forward Noise Adjustment Scheme for Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 728–734, November 2018.
- [138] Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. Re-thinking Model Inversion Attacks Against Deep Neural Networks, June 2023. arXiv:2304.01669.
- [139] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A Survey of Machine Unlearning, October 2022. arXiv:2209.02299 [cs].
- [140] P. Niyogi, F. Girosi, and T. Poggio. Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86(11):2196–2209, November 1998. Conference Name: Proceedings of the IEEE.
- [141] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, August 2017. arXiv:1603.09246 [cs].
- [142] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, April 2023. arXiv:2304.07193 [cs].
- [143] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, June 2012. ISSN: 1063-6919.
- [144] Ioannis Pastaltzidis, Nikolaos Dimitriou, Katherine Quezada-Tavarez, Stergios Aidinlis, Thomas Marquenie, Agata Gurzawska, and Dimitrios Tzovaras. Data augmentation for

- fairness-aware machine learning: Preventing algorithmic bias in law enforcement systems. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 2302–2314, New York, NY, USA, June 2022. Association for Computing Machinery.
- [145] Kexin Pei, Linjie Zhu, Yinzhi Cao, Junfeng Yang, Carl Vondrick, and Suman Jana. Towards Practical Verification of Machine Learning: The Case of Computer Vision Systems, December 2022. arXiv:1712.01785 [cs].
  - [146] Senthil Purushwalkam and Abhinav Gupta. Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases, July 2020. arXiv:2007.13916 [cs].
  - [147] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability, November 2017. arXiv:1706.05806 [cs, stat].
  - [148] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding Transfer Learning for Medical Imaging. *arXiv:1902.07208 [cs, stat]*, October 2019. arXiv: 1902.07208.
  - [149] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep Learning for Medical Image Processing: Overview, Challenges and Future, April 2017. arXiv:1704.06825 [cs].
  - [150] Morgane Rivi re, Armand Joulin, Pierre-Emmanuel Mazar , and Emmanuel Dupoux. Unsupervised pretraining transfers well across languages, February 2020. arXiv:2002.02848 [cs, eess].
  - [151] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021.
  - [152] Sebastian Schelter. “Amnesia” – Towards Machine Learning Models That Can Forget User Data Very Fast. 2020.
  - [153] Sebastian Schelter, Stefan Grafberger, and Ted Dunning. HedgeCut: Maintaining randomised trees for low-latency machine unlearning. In *Proceedings of the 2021 international*

*conference on management of data*, SIGMOD '21, pages 1545–1557, New York, NY, USA, 2021. Association for Computing Machinery. Number of pages: 13 Place: Virtual Event, China.

- [154] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember What You Want to Forget: Algorithms for Machine Unlearning, July 2021. arXiv:2103.03279 [cs].
- [155] Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy, May 2023. arXiv:2305.06360 [cs].
- [156] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models, March 2017. arXiv:1610.05820 [cs, stat].
- [157] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, July 2019.
- [158] Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht. Text Data Augmentation for Deep Learning. *Journal of Big Data*, 8(1):101, July 2021.
- [159] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias, May 2020. arXiv:2001.03152 [cs].
- [160] Dimitris Spathis, Ignacio Perez-Pozuelo, Soren Brage, Nicholas J. Wareham, and Cecilia Mascolo. Self-supervised transfer learning of physiological representations from free-living wearable data. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, pages 69–78, New York, NY, USA, April 2021. Association for Computing Machinery.
- [161] Dimitris Spathis, Ignacio Perez-Pozuelo, Laia Marques-Fernandez, and Cecilia Mascolo. Breaking away from labels: The promise of self-supervised machine learning in intelligent health. *Patterns*, 3(2), February 2022. Publisher: Elsevier.
- [162] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

- [163] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond Memorization: Violating Privacy Via Inference with Large Language Models, May 2024. arXiv:2310.07298.
- [164] Shobhita Sundaram and Neha Hulkund. GAN-based Data Augmentation for Chest X-ray Classification, July 2021. arXiv:2107.02970 [cs, eess].
- [165] Richard S. Sutton and Andrew Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, second edition edition, 2020.
- [166] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data Augmentation using Random Image Cropping and Patching for Deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931, September 2020. arXiv:1811.09030 [cs].
- [167] Luke Taylor and Geoff Nitschke. Improving Deep Learning with Generic Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547, November 2018.
- [168] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. arXiv:2302.13971.
- [169] Kelvin Tran, Lingfeng Xu, Gabriela Stegmann, Julie Liss, Visar Berisha, and Rene Utianski. Investigating the Impact of Speech Compression on the Acoustics of Dysarthric Speech. In *Interspeech 2022*, pages 2263–2267. ISCA, September 2022.
- [170] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On Data Augmentation for GAN Training. *IEEE Transactions on Image Processing*, 30:1882–1897, January 2021.
- [171] Eleni Triantafillou, Fabian Pedregosa, Jamie Hayes, Peter Kairouz, Isabelle Guyon, Meghdad Kurmanji, Gintare Karolina Dziugaite, Peter Triantafillou, Kairan Zhao, Lisheng Sun Hosoya, Julio C. S. Jacques Junior, Vincent Dumoulin, Ioannis Mitliagkas, Sergio Es-

- calera, Jun Wan, Sohier Dane, Maggie Demkin, and Walter Reade. Neurips 2023 - machine unlearning, 2023.
- [172] Trieu H. Trinh, Minh-Thang Luong, and Quoc V. Le. Selfie: Self-supervised Pretraining for Image Embedding, July 2019. arXiv:1906.02940 [cs, eess, stat].
- [173] Ayush Tripathi, Swapnil Bhosale, and Sunil Kumar Kopparapu. Improved Speaker Independent Dysarthria Intelligibility Classification Using Deepspeech Posteriors. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6114–6118, May 2020. ISSN: 2379-190X.
- [174] Yiwen Tu, Pingbang Hu, and Jiaqi Ma. Towards Reliable Empirical Machine Unlearning Evaluation: A Game-Theoretic View, April 2024. arXiv:2404.11577 [cs].
- [175] Lester Phillip Violeta, Wen-Chin Huang, and Tomoki Toda. Investigating Self-supervised Pretraining Frameworks for Pathological Speech Recognition, June 2022. arXiv:2203.15431 [cs, eess].
- [176] Haoyu Wang, Kehan Li, Jihua Zhu, Fan Wang, Chunfeng Lian, and Jianhua Ma. Weakly Supervised Tooth Instance Segmentation on 3D Dental Models with Multi-label Learning. In Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume 15009, pages 723–733. Springer Nature Switzerland, Cham, 2024. Series Title: Lecture Notes in Computer Science.
- [177] Kuan-Chieh Wang, Yan Fu, Ke Li, Ashish Khisti, Richard Zemel, and Alireza Makhzani. Variational Model Inversion Attacks, January 2022. arXiv:2201.10787.
- [178] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3462–3471, July 2017. ISSN: 1063-6919.
- [179] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. Conference Name: IEEE Transactions on Image Processing.

- [180] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine Unlearning of Features and Labels, August 2023. arXiv:2108.11577 [cs].
- [181] Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [182] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending Speech Separation to Noisy Environments, July 2019. arXiv:1907.01160 [cs, eess, stat].
- [183] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. TinyViT: Fast Pretraining Distillation for Small Vision Transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13681, pages 68–85. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science.
- [184] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional BERT Contextual Augmentation. In João M. F. Rodrigues, Pedro J. S. Cardoso, Jânio Monteiro, Roberto Lam, Valeria V. Krzhizhanovskaya, Michael H. Lees, Jack J. Dongarra, and Peter M.A. Sloot, editors, *Computational Science – ICCS 2019*, pages 84–95, Cham, 2019. Springer International Publishing.
- [185] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination, May 2018. arXiv:1805.01978 [cs].
- [186] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, September 2017. arXiv:1708.07747 [cs, stat] version: 2.
- [187] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or Signal: The Role of Image Backgrounds in Object Recognition, June 2020. arXiv:2006.09994 [cs].

- [188] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A Simple Framework for Masked Image Modeling, April 2022. arXiv:2111.09886 [cs].
- [189] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine Un-learning: A Survey, June 2023. arXiv:2306.03558 [cs].
- [190] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech Processing Universal PERFORMANCE Benchmark. In *Interspeech 2021*, pages 1194–1198. ISCA, August 2021.
- [191] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced Membership Inference Attacks against Machine Learning Models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, Los Angeles CA USA, November 2022. ACM.
- [192] Victoria Young and Alex Mihailidis. Difficulties in automatic speech recognition of dysarthric speakers and implications for speech-based applications used by the elderly: a literature review. *Assistive technology: the official journal of RESNA*, 22(2):99–112; quiz 113–114, 2010.
- [193] Victoria Young and Alex Mihailidis. Difficulties in Automatic Speech Recognition of Dysarthric Speakers and Implications for Speech-Based Applications Used by the Elderly: A Literature Review. *Assistive Technology*, 22(2):99–112, June 2010. Publisher: Taylor & Francis \_eprint: <https://www.tandfonline.com/doi/pdf/10.1080/10400435.2010.483646>.
- [194] Hang Yu, Aishan Liu, Gengchao Li, Jichen Yang, and Chongzhi Zhang. Progressive Diversified Augmentation for General Robustness of DNNs: A Unified Approach. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 30:8955–8967, 2021.
- [195] Zhengjun Yue, Erfan Loweimi, Heidi Christensen, Jon Barker, and Zoran Cvetkovic. Dysarthric Speech Recognition From Raw Waveform with Parametric CNNs. In *Interspeech 2022*, pages 31–35. ISCA, September 2022.

- [196] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into Deep Learning, February 2023. arXiv:2106.11342 [cs].
- [197] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A Review on Machine Unlearning. *SN Computer Science*, 4(4):337, April 2023.
- [198] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization, April 2018. arXiv:1710.09412 [cs, stat].
- [199] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization, March 2016.
- [200] Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder, March 2017. arXiv:1702.08423 [cs].
- [201] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image Augmentations for GAN Training, June 2020. arXiv:2006.02595 [cs, eess, stat].
- [202] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation, November 2017. arXiv:1708.04896 [cs].
- [203] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT, May 2023. arXiv:2302.09419.
- [204] Günter M. Ziegler. *Lectures on polytopes*. Number 152 in Graduate texts in mathematics. Springer-Verlag, New York, 1995.