

University of Sheffield

Hardness-Aware Learning: Taxonomy, Benchmarking, and Fairness Applications



Pawel Pukowski

Supervisor: Prof. Haiping Lu

This thesis is submitted for the degree of
Doctor of Philosophy

March 27, 2026

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: Pawel Pukowski

Signature: Pawel Pukowski

Date: March 27, 2026

Acknowledgments

I would like to express my gratitude to my supervisor, Professor Haiping Lu, and my co-supervisor, Professor Markus Szymik, for their guidance and support throughout my PhD. Their patience with my constantly changing research interests and constant attempts to keep me on track has been greatly appreciated. I would also like to thank Dr. Joachim Spoerhase for his valuable contributions and deep theoretical insights in our joint projects. I am especially grateful to Professor Venet Osmani, whose guidance, insight, patience, support, and dedication were instrumental to the completion of this thesis and to shaping my future career.

I would also like to thank my labmates and colleagues in Lab 136 — Sina Tabakhi, Rea Nkhumise, Dr. Juan José Giraldo, Dr. Lawrence Schöbs, Dr. Chunchao Ma, Dr. Xianyuan Liu, Dr. Prasun Chandra Tripathi, Mohammad Naimul Islam Suvon, Alan Thomas, Muhammad Ahmad Leo, Dr. Peizhen Bai, Dr. Shuo Zhou, Haolin Wang, Jiayang Zhang, Wenrui Fan, Lalu Muhammad Riza, and many others. They brought life and humanity to otherwise work-filled days, providing everyday conversations about culture and life that helped us all take our minds off the challenges of our PhD journey. I am particularly thankful to Dr. Joab R. Winkler for his thoughtful conversations, encouragement, and constant support, which helped me maintain perspective and motivation through this journey.

Finally, I would like to thank my parents and brother for their love and unwavering support. Knowing there was always someone there for me during uncertain times has meant more than words can express. I am also deeply grateful to my closest friends — Kacper Łodziński, Divjot, and Oscar Palma Chaundler — whose insistence on maintaining a healthy work-life balance kept me grounded and sane. I am indebted to my former teacher, Professor Jacek Własak, whose early encouragement inspired my interest in research and ultimately led me to pursue this path. Finally, I wish to thank all the individuals I have met throughout my life — peers, mentors, and teachers — too numerous to name individually, yet each of whom has, in their own way, influenced my path and contributed to who I am today.

Abstract

Class bias—systematic recall gaps across classes—continues to limit the fairness of classification models. Most explanations attribute this bias to data imbalance, assuming that unequal sample counts are its primary cause. However, this interpretation overlooks differences in class hardness that persist even in balanced datasets. Since prior studies rarely separate these effects, it remains unclear whether reported fairness gains arise from one or the other, constraining advances in hardness-aware learning. The focus of this thesis is twofold: bringing coherence to a fragmented field and showcasing its practical value.

To systematize existing approaches, we first review model- and data-based hardness estimators, which have evolved as largely independent research lines. We introduce a unified taxonomy that organizes both families and the types of hard samples they identify, providing a conceptual foundation for hardness-aware research. Building on this, we perform the first comparative benchmark between these two families, highlighting key limitations of data-based estimation. A complementary stability analysis further shows that single-seed evaluations—common in prior work—produce unreliable hardness rankings, motivating ensemble-based estimation.

To demonstrate why hardness-aware learning matters, we conduct two case studies assessing whether hardness-based resampling can improve fairness in balanced datasets. We find that it consistently reduces class-level recall and F1 gaps and can also enhance overall performance under suitable conditions. These outcomes depend on three key factors: the accuracy of hardness estimation, the quality of oversampled data, and the degree of induced data imbalance. Together, these findings establish hardness imbalance as a distinct and measurable phenomenon and position hardness-aware learning as a principled framework for studying and mitigating it.

Contents

List of Figures	vi
List of Tables	x
Notation	xi
1 Introduction	1
1.1 Motivation: Why Fairness Requires Addressing Hardness Imbalance, Not Just Data Imbalance?	1
1.2 Research Questions: How to Verify if Hardness-Based Resampling Improves Fairness?	2
1.3 Structure and Contributions	4
2 Foundations of Probability and Machine Learning	7
2.1 Supervised Classification Problem	7
2.2 Probability Foundations	8
2.3 Fundamentals of Machine Learning	10
2.3.1 Neural Networks	10
2.3.2 Learning	12
2.4 Statistical Estimators and Evaluation Metrics	14
2.5 Dimensionality Reduction	15
3 How to Estimate Hardness? Unified Taxonomy of Factors Causing Hardness	17
3.1 Introduction	17
3.2 Objective and Subjective hardness	19
3.3 Data-Based Hardness Estimators	20
3.3.1 Manifold Assumption and its Implications on Learning	21
3.3.2 Dimensionality-Based Estimators	22
3.3.3 Volume-Based Estimators	24
3.3.4 Topology-Based Estimators	25
3.3.5 Curvature-Based Estimators	26
3.3.6 Neighborhood-Based Estimators	28
3.4 Providing Geometry-Based Taxonomy of Hard Samples	29
3.4.1 Geometry- and Topology-Related Hardness	29
3.4.2 Sampling-Related Hardness	31

3.4.3	Error-Related Hardness	32
3.5	Model-Based Hardness Estimators	33
3.5.1	Confidence-Based Estimators	33
3.5.2	Margin-Based Estimators	34
3.5.3	Uncertainty-Based Estimators	35
3.5.4	Forgetting-Based Estimators	37
3.5.5	Loss-Based Estimators	38
3.5.6	Gradient-Based Estimators	39
3.5.7	Other Estimators	41
3.6	Providing Model-Based Taxonomy of Factors Causing Hardness	41
3.7	Unifying taxonomies of hard samples	44
3.7.1	Identifying research gaps	46
3.8	Conclusion	47
4	Which Hardness Estimator to Use? Verifying Estimate Robustness and Benchmark of Data- and Model-Based Estimators	48
4.1	Introduction	48
4.2	Reviewing Benchmarking Strategies of Hardness Estimators	51
4.2.1	Benchmarking in Works Involving Data-Based Hardness Estimators	51
4.2.2	Benchmarking in Works Involving Model-Based Hardness Estimators	52
4.2.3	Benchmark papers	53
4.2.4	Summary	54
4.3	Background on Stability of Hardness Estimates	55
4.4	Methodology	56
4.4.1	Comparing Data- and Model-Based Hardness Estimators	57
4.4.2	Instance-Level Stability and Agreement of Model-Based Hardness Estimators	58
4.4.3	Experimental Setup	60
4.5	Results and Discussion	62
4.5.1	Stability on Top- t Most Extreme Samples	62
4.5.2	Verifying Estimator Agreement Through Jaccard Overlap	65
4.5.3	Benchmarking Model- and Data-Based Estimators	67
4.6	Conclusion	69
5	Can Hardness-Based Resampling Improve Fairness in Balanced Setting?	71
5.1	Introduction	71
5.2	Background: From Class Imbalance to Hardness Imbalance	73
5.2.1	Types of Imbalance	73
5.2.2	Standard Techniques for Imbalance Correction	74
5.2.3	Measuring Quality of Generative Models	75
5.3	Formal Experimental Design	76
5.4	Methodology	78
5.4.1	Detailed Experimental Design	78
5.4.2	Experimental Setup	79
5.5	Results and analysis	81

<i>CONTENTS</i>	vi
5.5.1 Observations	81
5.6 Conclusion	90
6 Conclusion and Future Work	93
6.1 Conclusion	93
6.2 Future Work	95
Appendices	110
A Additional Experimental Results	111

List of Figures

1.1	The purpose of this thesis is twofold. First, we aim to bring structure to the field by providing a taxonomy of hard samples in Chapter 3 and addressing benchmarking of hardness estimators in Chapter 4. Second, we demonstrate the practical value of hardness-aware learning in Chapter 5, showing that introducing data imbalance through hardness-based resampling can improve both fairness and overall performance under certain conditions. Together, these efforts aim to consolidate current understanding and motivate further exploration of hardness as a key dimension of data-centric learning.	3
2.1	Visual representation of Multi-layer perceptron including the notation used in the main text.	10
3.1	We propose two complementary taxonomies: one derived from the literature on data-based hardness estimators, and one derived from the literature on model-based hardness estimators. The figure presents both taxonomies together and illustrates how the factors from each perspective relate to one another.	19
3.2	(a) A sequence of Vietoris–Rips (VR) complexes constructed on a 2D point cloud by increasing the pairwise distance threshold t . A k -simplex is added whenever all $\binom{k+1}{2}$ edges among $k + 1$ points have length $\leq t$. This results in a growing complex: as t increases, new edges and higher-order simplices (e.g., filled triangles) appear, yielding a nested sequence $K_0 \subseteq K_3 \subseteq K_5 \subseteq K_7$. (b) The barcode summarizes the persistence of topological features across this filtration: each bar represents the birth and death of a feature. For example, a loop feature is absent in K_5 but appears in K_7 , indicating that its birth lies within the interval between these two filtration steps. Dotted lines in (b) correspond to the thresholds shown in (a).	26

4.1	Our literature review revealed that many works use hardness estimates obtained over a single run. This raises concerns regarding stability and reliability of such hardness estimates. To address this we investigate the impact of the ensemble size on the hardness estimates, revealing that for some downstream tasks single-run-based hardness estimates are extremely unreliable. After establishing that we perform the first comparative benchmarking between model- and data-based hardness estimators, finding that current data-based estimators are not able to achieve good performance and identify curse of dimensionality as the main cause.	49
4.2	Estimated class-level intrinsic dimensions (IDs) using TwoNN. We provide this Figure to visualize the difference between the extrinsic dimensions—3072—and IDs for classes in CIFAR-10 and CIFAR-100. It is clear that the class manifolds are significantly lower dimensional than the space that they occupy.	57
4.3	Percentage change in the pruned indices (Eq. 4.1) when increasing the ensemble size by one, across two pruning thresholds (10%, 50%) and two scenarios (easy vs. hard pruning). High PCR indicates low stability of hardness estimates. We find that: (i) Forgetting is very unstable at identifying the easiest samples on simple datasets; and (ii) instant estimators are generally less stable than those using training dynamics. Results were averaged over five ensembles of randomly chosen models and std is not reported for clarity.	63
4.4	We report changes to the sample count of a class after hardness-based resampling (Absolute Difference) averaged across all classes when increasing the ensemble size used to estimate hardness by one. High Absolute Difference indicates low stability in resampling ratios and class-level hardness estimates. Unlike in Figure 4.3, we find Forgetting to be competitive with other dynamics-based hardness estimators indicating that struggling at identifying top- t easiest samples does not translate to difficulty in estimating class-level hardness. At the same time we notice large instability of EL2N, despite this estimator yielding similar stability to other hardness estimators at the instance level. The values are averaged over five ensembles of randomly chosen models and std is not reported for clarity.	64
4.5	We report Jaccard overlap (Eq. 4.3) between the set of pruned samples selected by various hardness estimators across two scenarios (easy vs. hard pruning) and five pruning thresholds. Low overlap indicates strong disagreement on which samples are considered most extreme, which can significantly affect downstream tasks. Notably, (i) estimator choice heavily influences pruning outcomes (columns 1 and 2), and (ii) incorporating learning dynamics (e.g., AUM vs. iAUM) leads to marked shifts in selected samples (column 3). . . .	66

4.6	We report the Pearson and Spearman rank correlations between all pairs of class-level hardness estimates obtained from raw input data, including their correlations with <i>Subjective Accuracy</i> and <i>Objective Accuracy</i> , which serve as proxies for ground truth. Compared to Fig. 4.5, the stronger agreement between estimators suggests that class-level hardness is easier to estimate than instance-level hardness. On top of that, the weak correlation between volume and ID — despite theoretical expectations of alignment — further supports the view that data-based estimates on raw input are unreliable.	68
4.7	We report correlation values between data-based hardness metrics (ID, volume, curvature) and class-level accuracy, as well as between ID and volume, for different PCA projection dimensions. on CIFAR-10 and CIFAR-100. Higher correlation values imply greater performance of data-based estimators. Our key insights are: (i) expected monotonic improvement with decreasing d is absent, suggesting that occasional high-correlation spikes are likely noise rather than a trend, and (ii) the same is the case for ID vs Volume, which are theoretically related, indicating that PCA reduction does not consistently improve data-based estimators.	69
5.1	Training an ensemble of ten ResNet18 networks on CIFAR-10 (left) and CIFAR-100 (right) reveals large recall gaps across classes, despite the balanced nature of these datasets. Paired with significantly larger recall gaps across classes for CIFAR-100 than CIFAR-10, this shows class- and dataset-level hardness discrepancies, which we call hardness imbalance. We highlight the (very clearly) visible negative effects of this imbalance, even on datasets that are data-balanced, and believe that it can be addressed by hardness-based resampling—oversampling hard classes and undersampling easy ones.	72
5.2	Case Study 1 (CIFAR-10): Analyzing changes in various fairness metrics on pruned versions of CIFAR-10 reveals that hardness-based resampling reduces the recall gap across classes at the cost of increasing the precision gap, with the improvements in recall overwhelming the drops in precision as indicated by F1 gap. We find that hardness-based resampling yields similar improvements no matter the pruning rate. Furthermore, it seems to be highly reliant on the quality of samples used for oversampling, as random- and SMOTE-based resampling was not able to meaningfully impact the performance gap. We report means over four ensembles of four models trained on 4 distinct $\mathcal{D}^{CLP}$ and do not report std for clarity.	82
5.3	Case Study 1 (CIFAR-100): Just like on CIFAR-10 we notice reduction in recall- and F1-based fairness metrics indicating reduction in performance gap across classes. This time the changes are more pronounced, most likely due to more linear-like hardness spectrum on this dataset (see Figure 5.1). The importance of high quality samples also becomes more apparent on CIFAR-100, as resampling using holdout set yields twice the improvement achieved by using SMOTE or random oversampling.	83

5.4	Case Study 2 (CIFAR-10): We notice that, in most cases, the higher the degree of the introduced data imbalance (x-axis) the larger the effect on fairness when high quality synthetic samples are available. We expect the existence of some threshold α beyond which the drawback from too heavy data imbalance starts to overwhelm the benefits of addressing hardness imbalance.	86
5.5	Case Study 2 (CIFAR-100): The effect of hardness-based imbalance is similar on both datasets here, unlike the fivefold more pronounced effect seen on CIFAR-100 in the first case study. We attribute this to the subpar quality of EDM-generated samples compared to the real samples from the holdout set. This is evidenced by the discrepancy between the forecasted fairness metrics from the first study and the higher values actually observed in this figure (e.g., Δ_{he} was forecasted to drop below 0.14 for $\alpha = 1$).	87
A.1	True negatives across classes on CIFAR-10.	111
A.2	True positives across classes on CIFAR-10.	112
A.3	False negatives across classes on CIFAR-10.	112
A.4	False positives across classes on CIFAR-10.	112
A.5	True negatives across classes on CIFAR-100.	113
A.6	True positives across classes on CIFAR-100.	113
A.7	False negatives across classes on CIFAR-100.	113
A.8	False positives across classes on CIFAR-100.	114

List of Tables

3.1	Summary of model-based hardness estimators reviewed in this chapter. Methods are listed in the order in which they were introduced in the section. . . .	42
3.2	Categorization of model-based hardness estimators based on the categorization of hard samples that they assume.	43
4.1	Overview of works by category. Model-based and data-based hardness studies differ in their benchmarking strategies, with the latter relying primarily on proxy-based evaluation.	54
5.1	Case Study 1 (CIFAR-10): t-statistics (with p-values) for fairness metrics under different pruning strategies. Significant results ($p < 0.05$) are bolded . .	84
5.2	Case Study 1 (CIFAR-100): t-statistics (with p-values) for fairness metrics under different pruning strategies. Significant results ($p < 0.05$) are bolded . .	85
5.3	Case Study 2 (CIFAR-10): t-statistics (with p-values) for fairness metrics under different oversampling strategies. Significant results ($p < 0.05$) are bolded	88
5.4	Case Study 2 (CIFAR-100): t-statistics (with p-values) for fairness metrics under different oversampling strategies. Significant results ($p < 0.05$) are bolded	89

Symbols and Notations

Sets and Spaces

- $\mathcal{D} = \{(x_i, \hat{y}_i)\}_{i=1}^N$: dataset of N input-label pairs
- $\mathcal{D}_c$: subset of dataset $\mathcal{D}$ containing only samples belonging to class c
- $x_i \in \mathbb{R}^d$: i -th input vector of dimension d
- $\hat{y}_i \in \{1, \dots, K\}$: true label corresponding to x_i
- x^{ho} : a sample from a hold-out (validation) set
- $\mathcal{N}(x)$: neighborhood of x in input or latent space
- $\mathbb{P}$: data-generating (underlying) distribution over (x, y)
- $\mathcal{M}_c$: class manifold for class c
- K : number of classes

Models and Predictions

- $\theta^{\mathcal{D}}$: parameters obtained by training the model M on dataset $\mathcal{D}$ (we omit $\mathcal{D}$ in cases where it is obvious)
- $M(x; \theta)$: neural network with parameters θ
- $z = M(x; \theta) \in \mathbb{R}^K$: output logit vector; latent representation of x
- z_j : logit corresponding to class j
- $z^{(t)}$: logit after t training epochs
- $p = \text{softmax}(z)$: predicted class probabilities
- $p_j = \mathcal{P}_j(x; \theta)$: predicted probability of x for class j given model parameters θ
- $y = \arg \max \mathcal{P}(x)$: predicted label
- $\bar{\mathcal{P}}(x)$: average predicted distribution across models or epochs

Learning and Optimization

- $\mathcal{L}(x, \hat{y}; \theta)$: loss function (e.g., cross-entropy)
- T : total number of training epochs

Latent Representations and Geometry

- Z_c : latent representations of samples from class c
- $\bar{Z}_c$: centroid of class c in latent space
- Σ : empirical covariance matrix of a class
- I : identity matrix
- $\text{tr}(\cdot)$: matrix trace
- $\det(\cdot)$: matrix determinant
- V_c : estimated volume of class manifold $\mathcal{M}_c$
- G_c : average Gaussian curvature of class manifold $\mathcal{M}_c$
- $\text{ID}(x)$: intrinsic dimension estimated at point x
- ID_c : intrinsic dimension of class c
- d : intrinsic dimension
- D : extrinsic dimension

Neighborhood-Based Quantities

- $N_k^{\text{same}}(x)$: k -nearest neighbors of x from the same class
- $N_k^{\text{other}}(x)$: k -nearest neighbors of x from different classes
- $d_{\text{avg}}^{\text{same}}(x)$: average distance to $N_k^{\text{same}}(x)$
- $d_{\text{avg}}^{\text{other}}(x)$: average distance to $N_k^{\text{other}}(x)$
- $\text{CI}(x)$: class impurity of the local neighborhood around x

$\mathbb{I}(\cdot)$: indicator function $\|\cdot\|_2$: L2 norm e_y : one-hot encoding of class y

We use the terms ‘hard’ and ‘difficult’ interchangeably. When we say that a data sample is ‘hard’ we mean that it is hard to learn and classify.

Chapter 1

Introduction

1.1 Motivation: Why Fairness Requires Addressing Hardness Imbalance, Not Just Data Imbalance?

Classification tasks—ranging from medical diagnosis (Esteva et al. 2017) and credit score evaluation (Lessmann et al. 2015) to spam detection (Crawford et al. 2015) and autonomous driving (Turay & Vladimirova 2022)—have driven much of modern ML research. In essence, the main goal is to substitute or assist humans in solving classification problems across diverse domains. Over the decades, classification has evolved from early linear models to kernel methods and now to various deep learning architectures and foundational models, reflecting the field’s growing theoretical and computational sophistication (Alom et al. 2019, Awais et al. 2025). Consequently, many classification models have already matched or even surpassed human performance in benchmark settings (He et al. 2015, Buetti-Dinh et al. 2019, Nasiri et al. 2025).

Despite these advances, ML classification systems remain far from universally deployed in high-stakes domains. This hesitation stems not from conservatism but from persistent issues in reliability, transparency, and fairness—here understood as the equitable predictive performance across classes, rather than fairness in the socio-ethical sense. Many of these problems originate in the data, with data imbalance—the disproportionate representation of certain classes—being one of the most extensively studied and persistent issues (Johnson & Khoshgoftaar 2019). This has led to the introduction of various methods for addressing this problem, which can be generally divided into two categories. The first, incorporating reweighing approaches, aims at modifying the learning algorithm by putting higher emphasis on minority samples (Fernández et al. 2018). The second, involving resampling approaches, focuses on removing the data imbalance altogether (Carvalho et al. 2025). These two categories cover the overwhelming majority of existing strategies for dealing with data imbalance and are deeply embedded in both research practice and applied pipelines.

While this extensive focus on data imbalance has yielded valuable insights and practical solutions, this emphasis may have inadvertently narrowed our conceptual framework for understanding imbalance in classification problems. The prevailing discourse often treats imbalance as synonymous with unequal class distributions, leading to an implicit assumption that minority classes are inherently disadvantaged and that achieving balanced performance requires primarily addressing sample size disparities. However, this perspective overlooks

a fundamental distinction: not all classes within a dataset are equally difficult to classify correctly. This is empirically observed across multiple studies, which show that accuracy discrepancies across classes persist even in balanced settings (Ma, Jiao, Liu, Li, Yang & Liu 2023, Kaushik et al. 2024), and that accuracy on tail classes can even exceed that of head classes (Sinha et al. 2020). This highlights that the conceptual framework based solely on data imbalance is insufficient to fully explain accuracy discrepancies across classes. A more correct framework should incorporate various types of hardness—which would each provide unique reasoning behind model’s difficulty in correctly classifying certain classes—with data imbalance being only one among several factors contributing to hardness imbalance. However, to the best of our knowledge, hardness imbalance as a systematic bias across classes has not been systematically theorized, measured, or benchmarked as rigorously as data imbalance. Thus, while fairness in classification is often pursued through balancing class counts, we argue that achieving complete fairness actually requires addressing disparities in intrinsic class hardness.

Even though the notion of hardness imbalance has not been systematically conceptualized in its own right, the concept of hardness is not a novel one and appears in multiple subfields. In curriculum learning, hardness is used to determine the order in which samples are learned to improve efficiency by starting with easy samples and slowly introducing harder samples to the curriculum (Wang et al. 2021, Soviany et al. 2022). In active learning, hardness is used to identify the most informative samples that are to be manually labeled after the initial exploration phase (Tharwat & Schenck 2023, Li et al. 2024). In data pruning, hardness is used to determine the data samples that are least valuable to the learning process and can be removed without negatively affecting the accuracy of the final networks (Sorscher et al. 2022). In noise identification, hardness is used to identify challenging samples for removal (Pleiss et al. 2020). Across these diverse applications, hardness is thus treated primarily as a practical tool to improve training efficiency, labeling cost or dataset quality, rather than as a quantity of intrinsic interest. This means that hardness plays only a secondary role in those studies. There also exist many works focusing on data imbalance, which explicitly aim at improving fairness, that incorporate hardness into their methods (e.g., Focal Loss (Lin et al. 2017) or ADASYN (He et al. 2008)). Unfortunately, they either omit explicit recognition of hardness imbalance or lack ablation studies disentangling it from data imbalance. This omission is a crucial problem, as it makes it unclear whether simple methods that are known to address data imbalance can also mitigate hardness imbalance. Furthermore, the fragmented nature of this field makes it hard to understand the current state-of-the-art or make meaningful progress.

1.2 Research Questions: How to Verify if Hardness-Based Resampling Improves Fairness?

This thesis has two overarching objectives. First, it aims at bringing structure to the field of hardness-aware learning and construct a knowledge map of current insights and state-of-the-art, as well as highlight the main open questions and promising future directions. Second, it seeks to demonstrate the practical value of hardness-aware methods. Together, these objectives are designed to provide both conceptual clarity and empirical validation.

To achieve them, the work is divided into two complementary parts. The first part,

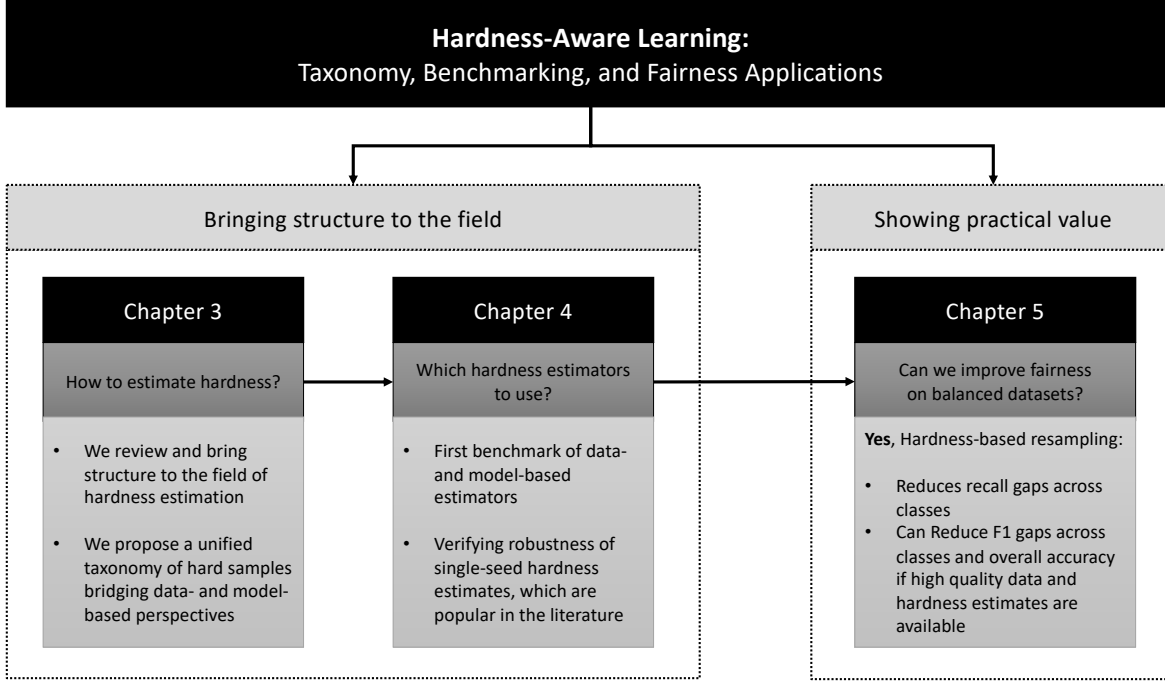


Figure 1.1: *The purpose of this thesis is twofold. First, we aim to bring structure to the field by providing a taxonomy of hard samples in Chapter 3 and addressing benchmarking of hardness estimators in Chapter 4. Second, we demonstrate the practical value of hardness-aware learning in Chapter 5, showing that introducing data imbalance through hardness-based resampling can improve both fairness and overall performance under certain conditions. Together, these efforts aim to consolidate current understanding and motivate further exploration of hardness as a key dimension of data-centric learning.*

corresponding to Research Questions Q1–Q3, focuses on structuring the insights from the literature on hardness-aware learning: outlining which factors cause hardness, describing how hardness can be estimated, how reliable these estimates are, and how to benchmark them. The second part—corresponding to Q4—investigates the practical utility of hardness-based resampling by examining whether it improves fairness and, if so, under what conditions.

1. How to estimate hardness?

- **Q1:** How can hardness estimators and the underlying factors of hardness be systematically categorized?

2. Which hardness estimators to use?

- **Q2:** Are data-based hardness estimators a viable alternative to the more widely used model-based estimators?
- **Q3:** How robust is the common practice of estimating hardness using a single random seed?

3. Does hardness-based resampling improve fairness on balanced datasets?

- **Q4:** What are the conditions under which hardness-based resampling improves fairness in balanced setting?

A schematic overview of these research questions is presented in Figure 1.1, which illustrates how they structure the thesis. Q1 will be targeted by Chapter 3, Q2 and Q3 will be targeted by Chapter 4 and Q4 will be answered in Chapter 5.

1.3 Structure and Contributions

Chapter 2 introduces the foundations of machine learning, probability and statistics, which are relevant for this thesis.

Chapter 3 brings structure and clarity to the growing but fragmented field of hardness-aware learning by reviewing the most prominent hardness estimators. The chapter organizes existing approaches into two complementary families:

- *data-based estimators*, which infer hardness from the geometry, density, or local topology of the data manifold; and
- *model-based estimators*, which infer hardness from model behaviour, such as prediction confidence, loss dynamics, or gradient signals.

For each family, we introduce a consistent categorization of methods and provide mathematical formulations of representative estimators, explaining their underlying rationale and interconnections. For data-based estimators we frame the discussion using a geometric perspective on learning (the manifold hypothesis), showing how different estimators capture complementary aspects of manifold geometry and sampling. For model-based estimators we show that almost every study implicitly adopts its own notion of sample difficulty; we aggregate these categorizations into a single, coherent, model-based taxonomy that consolidates overlapping definitions and clarifies their relationships. Together, these analyses expose the conceptual complementarity between geometric and behavioral perspectives and motivate a joint taxonomy of hard samples.

Contributions In Chapter 3 we introduce a unified taxonomy that bridges data-based and model-based perspectives. This framework combines the interpretability and theoretical grounding of data-based estimators with the empirical signals captured by model-based methods, offering a principled synthesis of previously disconnected lines of research. In the process we also identify three key open problems that motivate the rest of the thesis and future work: (i) the computational intractability of geometric estimators in high-dimensional input spaces; (ii) the under-exploration of training dynamics as a source of hardness information; and (iii) the conceptual conflation of distinct subtypes of hardness across prior studies. By consolidating insights scattered across disparate lines of work, the chapter transforms a fragmented literature into a coherent conceptual framework, providing the structural foundation needed for further progress in hardness-aware learning.

Chapter 4 builds on the conceptual foundations established in Chapter 3 by addressing the practical problem of benchmarking hardness estimators. Its purpose is to clarify why evaluating these estimators remains challenging—most notably due to the absence of

a ground-truth notion of sample difficulty—and to systematize the benchmarking strategies that have emerged to address this gap. We group existing evaluation paradigms into three broad families: downstream-task-based, injection-based, and proxy-based, each reflecting different assumptions about what constitutes a reliable indicator of sample hardness. After reviewing these approaches and the only existing benchmark study in the field, we identify two critical gaps in the literature: (i) the lack of direct comparison between model-based and data-based estimators, and (ii) the widespread reliance on single-seed evaluations. In addition, we observe that estimator robustness is typically assessed using correlations across the entire hardness spectrum—a practice that overlooks instability at the hardest and easiest extremes, where robustness matters most for downstream applications such as noise removal or pruning. These observations motivate our second and third research questions (**Q2** and **Q3**), which this chapter addresses through systematic empirical analysis. In doing so, we also revisit one of the challenges highlighted in Chapter 3—the susceptibility of data-based estimators to the curse of dimensionality—by examining whether dimensionality reduction via PCA can improve the reliability of these estimators.

Contributions In Chapter 4 we perform the first cross-family benchmark comparing data- and model-based hardness estimators. Our results show that data-based estimators perform poorly as class-level hardness measures, and identify the curse of dimensionality as the primary cause. Further experiments demonstrate that simple dimensionality reduction techniques like PCA bring too much distortion into geometric and topological structure, rendering it ineffective for preserving meaningful hardness relationships. These findings underscore the need for more geometry- and topology-aware dimensionality reduction techniques if interpretable, data-based hardness estimation is to become competitive. We also introduce a stability-based evaluation criterion that tests whether the top- t hardest or easiest samples remain consistent across random seeds—a more meaningful measure of estimator robustness than global correlation—and find that many model-based estimators are very inconsistent across seeds. Collectively, these findings establish both methodological and practical foundations for reliable benchmarking, motivate our choice of hardness estimator for subsequent chapter, and advance the broader framework of hardness-aware learning.

Chapter 5 takes a different direction from the preceding chapters, shifting from conceptual contributions to practical evaluation. Building on the benchmarking results of Chapter 4 this chapter investigates whether hardness-based resampling can improve fairness even in balanced datasets (**Q4**). This question challenges the common assumption that data balance implies fairness, offering a paradigm shift towards redistributing training focus according to sample difficulty. Answering this question is non-trivial because any observed effect depends on the quality of the data used in oversampling. To disentangle the impact of resampling from that of data synthesis, we design two complementary case studies. The first is a controlled simulation that assumes access to real data during oversampling. The second represents a realistic scenario in which oversampling relies on synthetic data. To ensure a rigorous and quantitative assessment, we introduce five complementary fairness metrics that (i) compare performance between the easiest and hardest classes, and (ii) quantify variability across all classes, allowing for a nuanced analysis of class-wise equity.

Contributions Our results show that hardness-based resampling consistently reduces recall-based class bias. Crucially, the mechanism differs from data-imbalanced settings: there, random oversampling reduces bias simply by balancing frequencies; here, with frequencies already balanced, only oversampling that adds new and informative content can help. Accordingly, random oversampling and SMOTE yield negligible changes, whereas a state-of-the-art diffusion model (Karras et al. 2022) produces substantial reductions—closing the recall gap between average easy and hard classes by 0.014 on CIFAR-10 and 0.027 on CIFAR-100.

We further find that when generative models are involved, harder synthetic samples are more beneficial: selecting samples based on hardness rather than chance nearly doubles the reduction in class bias. This suggests that improving generative models’ ability to produce hard samples—rather than the easy samples they currently favor (Wang et al. 2024)—could yield further gains. Importantly, these reductions do not come at the expense of overall accuracy, which in most cases increases slightly. The only trade-off is a modest increase in precision gaps across classes—a natural consequence of the precision-recall trade-off and the strong alignment between class-level precision and recall.

Altogether, these findings demonstrate that class bias is not merely a symptom of data imbalance, but a more fundamental phenomenon rooted in the unequal distribution of hardness across classes. If we are to build models that are truly fair—across all classes, regardless of their frequency—the field must pivot from frequency-centric corrections toward hardness-aware methods that address bias where it actually originates.

Chapter 2

Foundations of Probability and Machine Learning

The aim of this chapter is to provide the mathematical foundation of probability, machine learning and statistics to understand this thesis. I start with defining the supervised classification problem and introducing the foundations of probability, relating each concept to the classification problem. After that I describe how neural networks perform classification and learn through a simple multi-layer perceptron. I end this chapter by introducing the core statistical estimators and evaluation metrics used in this thesis and by describing dimensionality reduction.

2.1 Supervised Classification Problem

Let $X_{\mathcal{D}} = \{x_1, \dots, x_N\}$ be a set of N data samples. These samples can be of any modality, but in this thesis we restrict our experiments to image data. Let $Y_{\mathcal{D}} = \{y_1, \dots, y_K\}$ be a set of K integers, called labels, each mapped to a conceptual class. For example, in CIFAR-10 y_1 is mapped to ‘airplane’, and y_4 is mapped to ‘cat’.

A dataset $\mathcal{D}$ consists of these samples and labels, together with a surjective mapping $f_{\mathcal{D}} : X_{\mathcal{D}} \rightarrow Y_{\mathcal{D}}$ that assigns each image a label. So in CIFAR-10, $f_{\mathcal{D}}$ is supposed to map images of airplanes to y_1 , images of cats to y_4 , and so on. In practice, however, datasets often contain label noise: an image of a cat might accidentally be assigned a different label. This describes a supervised setting. In an unsupervised setting the mapping f is not available and in some cases $Y_{\mathcal{D}}$ is also not available.

In some domains—medicine, for example—a deterministic mapping f does not adequately capture reality. The same symptoms x may indicate different diseases y with different probabilities. Here, the relationship is better described by a conditional probability distribution $P(Y|X)$, which tells us, for a given x , how likely each possible y is. The strength of the connection between a specific symptom x and a specific disease y is precisely the probability $P(Y = y|X = x)$. For simplicity, however, we will focus on the deterministic case in this thesis.

It is commonly assumed that each data sample $x_i \in X_{\mathcal{D}}$ was drawn from some underlying distribution. More formally, we assume there exists a joint distribution $P(X, Y)$ over all possible inputs and labels. Our finite dataset $\mathcal{D}$ is just a small collection of samples from this

vast space obtained through P .

Recall that we have a dataset $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, f_{\mathcal{D}})$, where $f_{\mathcal{D}} : X_{\mathcal{D}} \rightarrow Y_{\mathcal{D}}$ is the observed labeling within the dataset. However, this $f_{\mathcal{D}}$ is merely a finite snapshot of a broader underlying relationship. We assume there exists a true labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$ defined over the entire input space $\mathcal{X}$. The labels we observe in $\mathcal{D}$ are then the result of applying f to our specific images x_i , possibly with some errors. In other words, $f_{\mathcal{D}}$ is what we see; f is what we wish we knew.

The aim of a classification algorithm is to learn a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$ that approximates the true labeling function f as closely as possible. Crucially, h must perform well not just on the images we already have in $X_{\mathcal{D}}$, but on all possible images from $\mathcal{X}$ —including those we have never seen. This property is known as *generalization*, and it is the central challenge of machine learning.

Ideally, we would evaluate h by measuring its performance on a fresh set of images drawn from the true underlying distribution $P(X, Y)$. In practice, however, we do not have access to such an infinite stream of labeled data, as P is inaccessible. Instead, we simulate this evaluation by partitioning our finite dataset $\mathcal{D}$ into two disjoint subsets: a training set $\mathcal{D}_{\text{train}} = (X_{\text{train}}, Y_{\text{train}}, f_{\text{train}})$ and a test set $\mathcal{D}_{\text{test}} = (X_{\text{test}}, Y_{\text{test}}, f_{\text{test}})$. The hypothesis h is learned using only $\mathcal{D}_{\text{train}}$, and its performance is measured on $\mathcal{D}_{\text{test}}$, which serves as a proxy for unseen data. The expectation is that performance on $\mathcal{D}_{\text{test}}$ will approximate performance on the true distribution P .

2.2 Probability Foundations

The previous section introduced probabilistic concepts informally. We now define them precisely, grounding each in the classification setting.

To formalize the idea of drawing an image from the world, we need a way to talk about a quantity that is not fixed—one that can take different values each time we look. This is the role of a *random variable*. We denote random variables with uppercase letters (X, Y) and their realized values with lowercase letters (x, y). In our setting, X represents the random process of drawing an image; $x_i \in X_{\mathcal{D}}$ is a specific realization. Y represents the associated label; $\hat{y}_i$ is the observed label for x_i .

A random variable alone is not enough—we also need to specify how likely each possible outcome is. This is the role of a *probability distribution*. How it is represented depends on whether the outcomes are discrete or continuous. For discrete variables like labels Y , we use a *probability mass function* $P(Y = y)$, which assigns probabilities that sum to one. $P(Y = y)$ is simply the probability that a randomly drawn image belongs to class y —in a balanced dataset, $P(Y = y) = 1/K$. For continuous variables like images X , we use a *probability density function* $p(x)$. Here, $p(x)$ describes relative likelihood: regions where $p(x)$ is high are more likely to produce samples than regions where $p(x)$ is low. In real-world terms, this captures sampling biases—for example, certain demographics being underrepresented in medical imaging datasets.

Once we have a distribution, we often want to summarize its behavior with key quantities. The *expected value* (or *expectation*) $\mathbb{E}[X]$ quantifies the average outcome under the distribution. For a discrete variable, $\mathbb{E}[X] = \sum_x xP(X = x)$; for a continuous variable, $\mathbb{E}[X] = \int_{\mathcal{X}} xp(x)dx$. The *variance* $\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$ measures how spread out the

outcomes are around this average, and its square root is the standard deviation. These summaries will become important later when we consider different sources of uncertainty in learned models (Section 3.5.3). Specifically, aleatoric uncertainty corresponds to the expected variance of the model’s predictions, while epistemic uncertainty corresponds to the variance of the expected prediction.

The distribution tells us how likely different outcomes are, but it also implicitly defines which outcomes are possible at all. The *support* of a distribution is the set of outcomes with non-zero probability (for discrete variables) or non-zero density (for continuous variables). This is distinct from the input space $\mathcal{X}$, which contains all mathematically possible images—every $32 \times 32 \times 3$ array of pixel intensities, regardless of whether it resembles anything real. The support of $P(X)$ is the set of images that can actually appear under the data-generating process: photographs of real-world scenes, not random noise. Similarly, the support of $P(X \mid Y = \text{cat})$ is the set of all possible cat images—photographs, paintings, cartoons—which is still far larger than any finite dataset. This distinction clarifies why generalization is hard: our training samples cover only a tiny fraction of the support.

The expectation (or expected value) of a random variable X quantifies its average outcome under its distribution. For a discrete variable, $\mathbb{E}[X] = \sum_x xP(X = x)$; for a continuous variable, $\mathbb{E}[X] = \int_{\mathcal{X}} xp(x)dx$. Expectation is linear: $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$. In machine learning, we often work with the expected loss $\mathbb{E}_{(X,Y) \sim P}[\mathcal{L}(h(X), Y)]$, which we approximate by the empirical average over samples. The variance $\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$ measures spread around the mean, and its square root is the standard deviation.

With these tools, we can now define precisely what we mean by $P(Y \mid X)$ —the *conditional distribution*. For a given image x , $P(Y \mid X = x)$ tells us the probability of each label. This is the central object of interest in classification: $P(Y = \text{cat} \mid X = x)$ is the probability that photograph x contains a cat. In the deterministic case, this distribution places all probability mass on a single class—this is the true labeling function f . In probabilistic settings like medical diagnosis, it assigns non-zero probability to multiple diseases given the same symptoms.

The *joint distribution* $P(X, Y)$ ties everything together. It describes the probability of observing a particular image-label pair (x, y) together, and completely characterizes the data-generating process: to sample from $P(X, Y)$, we first pick a class according to $P(Y)$, then generate an image from that class according to $P(X \mid Y)$. This joint distribution is what we would need to compute the ideal classifier, but it is unknown—we only have access to samples from it via the dataset $\mathcal{D}$.

To learn $P(Y \mid X)$ from data, we must work with the finite collection of samples $\mathcal{D}$. This requires an assumption about how these samples relate to the underlying distribution. We assume that the samples $(x_i, \hat{y}_i)$ are drawn *independently and identically distributed* (i.i.d.) from the joint distribution $P(X, Y)$. Independence means that knowing one sample tells us nothing about another; identical distribution means each sample comes from the same underlying process. This assumption justifies using the empirical average over training samples as a proxy for the true expectation, and underpins the train/test split methodology. In practice, it is often violated—for instance, when datasets contain near-duplicate images or temporal correlations.

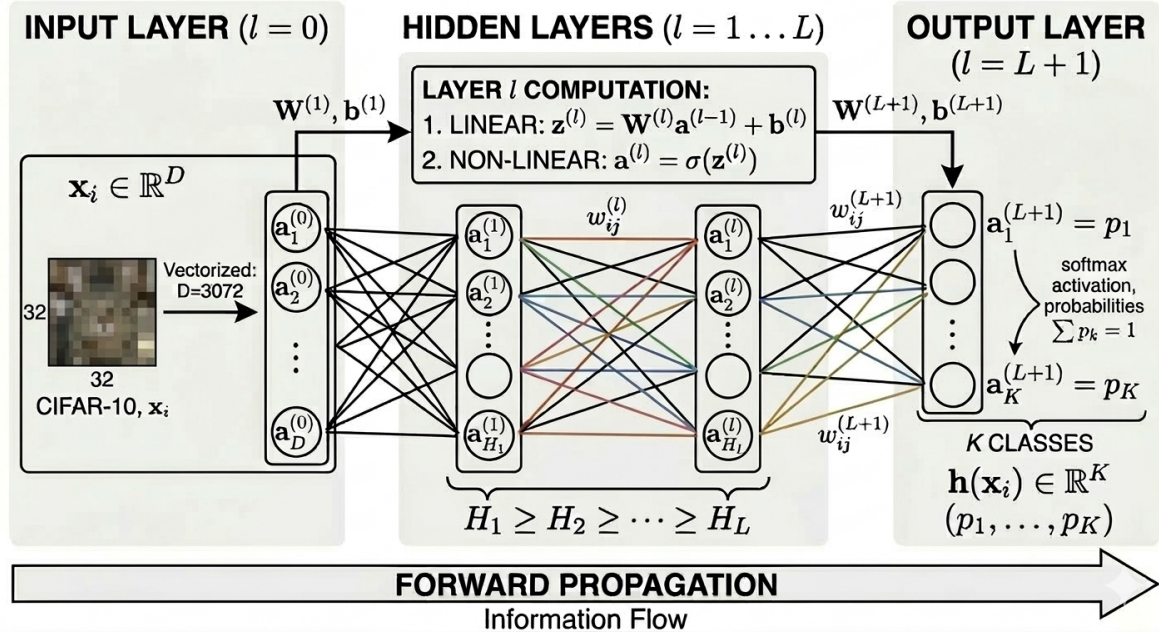


Figure 2.1: Visual representation of Multi-layer perceptron including the notation used in the main text.

2.3 Fundamentals of Machine Learning

2.3.1 Neural Networks

Neural networks are the primary tool we use to solve classification problems. A neural network is constructed by composing many simpler functions together, resulting in a single, highly complex function. For instance, the ResNet-18 architecture used in this thesis contains nearly 12 million parameters, while modern architectures have hundreds of billions of parameters. Rather than delving into the specifics of convolutional or other more complex architectures, we will focus on explaining the multilayer perceptron (MLP)—the foundational building block from which modern deep networks evolved. Understanding the MLP is sufficient for the purposes of this thesis, as the core principles of optimization and representation learning carry over directly to the more complex architectures we employ in our experiments.

An MLP consists of layers of interconnected neurons, as illustrated in Figure 2.1. The first of these is the input layer, which does not perform computation but simply holds the input data. For a given data sample $x_i \in \mathbb{R}^D$, the input layer contains D neurons, each representing one feature dimension. In CIFAR-10, each image is $32 \times 32 \times 3 = 3072$ pixels, so $D = 3072$ and each neuron corresponds to a single pixel intensity.

Following the input layer are one or more hidden layers. Let L denote the number of hidden layers, and let H_l be the number of neurons in the l -th hidden layer. While the values of H_l can be chosen arbitrarily, a common design practice is to gradually decrease the number of neurons in deeper layers, i.e., $H_1 \geq H_2 \geq \dots \geq H_L$. The universal approximation theorem guarantees that an MLP with a single hidden layer can approximate any continuous function given sufficient neurons (Cybenko 1989). However, deep networks with many layers

are preferred in practice, as they tend to learn more efficiently and yield better performance.

Each neuron in a given layer is connected to every neuron in the subsequent layer, which is why these are called fully connected (or dense) layers. The connections between layer $l - 1$ and layer l are parameterized by a weight matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{H_l \times H_{l-1}}$ and a bias vector $\mathbf{b}^{(l)} \in \mathbb{R}^{H_l}$. The weight $w_{ij}^{(l)}$ determines the strength of the connection from the j -th neuron in layer $l - 1$ to the i -th neuron in layer l .

Forward Propagation

Information flows through the network via a process called forward propagation. Let $\mathbf{a}^{(l-1)} \in \mathbb{R}^{H_{l-1}}$ be the vector of activations from the previous layer. The current layer first computes a linear transformation:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}. \quad (2.1)$$

This pre-activation is then passed through a non-linear *activation function* $\sigma(\cdot)$ to produce the layer's output:

$$\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}). \quad (2.2)$$

Common choices for σ include the logistic sigmoid, the hyperbolic tangent ($\tanh$), and the rectified linear unit (ReLU - $\max(0, x)$). The non-linearity is essential—without it, stacking multiple layers would collapse into a single linear transformation, severely limiting the network's expressive power.

For the first hidden layer ($l = 1$), the input $\mathbf{a}^{(0)}$ is simply the data sample x_i . After passing through all L hidden layers, we obtain the final hidden representation $\mathbf{a}^{(L)}$. This is then mapped to the output layer, which for classification problems typically uses a softmax activation to produce a probability distribution over the K classes.

The entire MLP can thus be viewed as a single composite function:

$$h(x_i) = \text{softmax}(\mathbf{W}^{(L+1)} \sigma(\dots \sigma(\mathbf{W}^{(1)} \mathbf{a}^{(0)} + \mathbf{b}^{(1)}) \dots) + \mathbf{b}^{(L+1)}), \quad (2.3)$$

where we have absorbed the final layer transformation into the notation and $\mathbf{x}_i$ is a vectorized form of x_i . This function $h : \mathbb{R}^D \rightarrow \mathbb{R}^K$ is our hypothesis—the classifier we aim to train to approximate f . Hence, for a given input x_i , the output $h(x_i)$ is a vector of K numbers between 0 and 1 that sum to 1, which we interpret as the predicted probability of each class.

Inference

The hypothesis h can be used to make predictions on new data—a process called inference. Given an input x_i , we compute $h(x_i)$ and then select the class with the highest predicted probability:

$$y_i = \arg \max_k [h(x_i)]_k, \quad (2.4)$$

where $[h(x_i)]_k$ denotes the k -th element of the output vector. This decision rule—choosing the most probable class—follows directly from the principle of maximum likelihood. This predicted label y_i may or may not coincide with the true label, which we denote from now on as $\hat{y}_i$. The true label $\hat{y}_i$ is the ground truth—the label that would be assigned by the ideal function f .

2.3.2 Learning

Loss Function

The discrepancy between the predicted label y_i and the ground truth $\hat{y}_i$ is what we aim to minimize during training. More precisely, we need a way to quantify how wrong the network's predictions are for each individual sample. This is the role of a loss function $\mathcal{L}(h(x_i), \hat{y}_i)$, which measures the cost of predicting $h(x_i)$ when the true label is $\hat{y}_i$. Intuitively, if the network assigns high probability to the correct class, the loss should be small; if it assigns low probability to the correct class, the loss should be large. Hence, the goal of learning is to find the parameters of the network—all weight matrices $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$ —that minimize this loss averaged over the entire training data. This average is known as the *empirical risk*:

$$\mathcal{R}_{\text{train}} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(h(x_i), \hat{y}_i). \quad (2.5)$$

For classification problems, the most common choice is the cross-entropy loss. Recall that for a given input x_i , the network outputs a probability vector $h(x_i) = [p_1, p_2, \dots, p_K]^\top$, where $p_k = P(Y = k \mid x_i)$ is the predicted probability of class k . If the ground truth label for this sample is $\hat{y}_i$, we can represent it as a one-hot vector $\mathbf{y}_{\text{true}} \in \{0, 1\}^K$ that has a 1 at position $\hat{y}_i$ and 0 elsewhere. The cross-entropy loss is then defined as:

$$\mathcal{L}_{\text{CE}}(h(x_i), \hat{y}_i) = -\log p_{\hat{y}_i} = -\sum_{k=1}^K [\mathbf{y}_{\text{true}}]_k \log p_k. \quad (2.6)$$

In words, it is the negative logarithm of the probability assigned to the correct class. This loss has two desirable properties. First, it is minimized when the network assigns probability 1 to the correct class and 0 to all others. Second, minimizing cross-entropy is equivalent to making the model's predictions as correct as possible given the training data—in other words, it directly implements the principle of maximum likelihood estimation.

Back Propagation

To minimize $\mathcal{R}_{\text{train}}$, we need to adjust the network's parameters in a direction that reduces the loss. This requires computing the gradient of the loss with respect to every weight and bias in the network. The algorithm that accomplishes this efficiently is called back propagation.

Back propagation relies on the chain rule. The loss $\mathcal{L}$ depends on the output layer's activations, which depend on the previous layer's activations, and so on back to the input. By applying the chain rule, we can compute the gradient of $\mathcal{L}$ with respect to each parameter layer by layer, working backwards from the output to the input.

For a single sample x_i , let us denote the gradient of the loss with respect to the pre-activation $\mathbf{z}^{(l)}$ as $\boldsymbol{\delta}^{(l)} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(l)}}$. Starting from the output layer, we compute $\boldsymbol{\delta}^{(L+1)}$ directly from the loss. Then, for each layer l from $L + 1$ down to 1, we propagate the gradient backwards:

$$\boldsymbol{\delta}^{(l)} = \left((\mathbf{W}^{(l+1)})^\top \boldsymbol{\delta}^{(l+1)} \right) \odot \sigma'(\mathbf{z}^{(l)}) \quad (2.7)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)} (\mathbf{a}^{(l-1)})^\top \quad (2.8)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)}, \quad (2.9)$$

where $\odot$ denotes element-wise multiplication and σ' is the derivative of the activation function.

Gradient Descent

The gradients computed via backpropagation tell us how to adjust each parameter to reduce the loss for a single sample x_i . To improve performance across the entire training set, we need the direction that reduces the average loss:

$$\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{train}} = \frac{1}{N} \sum_{i=1}^N \nabla_{\boldsymbol{\theta}} \mathcal{L}(h(x_i), \hat{y}_i), \quad (2.10)$$

where $\boldsymbol{\theta}$ denotes all network parameters concatenated—all weight matrices $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$. In gradient descent, we iteratively update the parameters by stepping opposite this gradient:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{train}}, \quad (2.11)$$

where $\eta > 0$ is a constant called *learning rate* which controls the convergence speed. This full-batch version uses all N samples per update, which is computationally prohibitive for large datasets and can lead to poor generalization.

Stochastic gradient descent (SGD) addresses these issues by approximating the gradient using a small random mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ of M samples:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{1}{M} \sum_{i \in \mathcal{B}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(h(x_i), \hat{y}_i). \quad (2.12)$$

This approximation makes each update computationally feasible, introduces noise that helps escape local minima, and often leads to better generalization.

2.4 Statistical Estimators and Evaluation Metrics

We have established the probabilistic foundations underlying data generation and how neural networks operate. We now turn to tools for summarizing and analyzing the finite samples we actually have—the dataset $\mathcal{D}$. These statistical quantities serve two purposes: they help us understand the structure of our data, and they provide metrics for evaluating model performance.

Given a set of observations $x_1, \dots, x_N$ of a random variable X , the sample mean, defined as:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i, \quad (2.13)$$

estimates the expected value $\mathbb{E}[X]$. In classification, we might compute the mean of all images belonging to a particular class to obtain a “prototype” or average representation of that class. More commonly, we use the sample mean when normalizing data—subtracting the mean from each sample centers the data and improves training stability.

While the mean describes the central tendency of the data, we often also need to know how spread out the observations are around this center. The sample variance, defined as:

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^2, \quad (2.14)$$

estimates this spread. Its square root, the sample standard deviation $\hat{\sigma}$, expresses the spread in the original units of the data. These quantities are used in feature scaling (e.g., standardizing features to zero mean and unit variance) and in assessing the variability of model predictions.

The mean and variance describe individual variables in isolation. To understand how two variables relate to each other, we turn to covariance. For two random variables X and Z with paired observations (x_i, z_i) , the sample covariance is:

$$\widehat{\text{Cov}}(X, Z) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu}_X)(z_i - \hat{\mu}_Z). \quad (2.15)$$

A positive value indicates that when X is above its mean, Z tends to be above its mean as well; a negative value indicates an inverse relationship.

For multivariate data, we need a single object that encodes two complementary types of information: how much each feature varies on its own, and how features vary together. The *covariance matrix* $\Sigma \in \mathbb{R}^{D \times D}$ does exactly this. Its diagonal entries Σ_{jj} capture the variance of each individual feature—whether a feature like “square footage” is tightly clustered or widely spread across samples. Its off-diagonal entries Σ_{jk} capture how features co-vary, extending the pairwise covariance concept we just defined to every pair of dimensions simultaneously.

Evaluation Metrics

Covariance is scale-dependent, making it difficult to interpret strength. The Pearson correlation coefficient, defined as:

$$\rho_{X,Z} = \frac{\text{Cov}(X, Z)}{\sigma_X \sigma_Z}, \quad (2.16)$$

normalizes covariance by the product of standard deviations, producing a dimensionless value between -1 and 1 . Pearson correlation indicates whether two sets of numbers move together linearly—for example, whether model predictions increase as ground truth values increase.

Spearman rank correlation extends this idea to monotonic relationships of any form. Rather than operating on raw values, it replaces each value with its rank within its set, then computes Pearson correlation on these ranks. The result, also between -1 and 1 , captures whether Z consistently increases when X increases, regardless of whether the relationship is linear. This makes Spearman useful for nonlinear trends and robust to outliers.

Both Pearson and Spearman correlation measure the strength of a relationship, but neither measures agreement. Two sets of numbers can be perfectly correlated yet systematically different—for example, predictions that are always twice the ground truth yield Pearson correlation 1 despite being wrong. More importantly, a correlation coefficient summarizes the global relationship between two sets of numbers with a single value. For large datasets, this global statistic can mask considerable local structure: substantial subsets of the data may exhibit very different relationships while the overall correlation remains high. A high correlation therefore does not guarantee consistency across all regions of the data, a distinction that becomes important in Chapter 4.

While the previous metrics apply to continuous variables, the Jaccard index measures similarity between sets. For two sets A and B , the Jaccard overlap is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (2.17)$$

the ratio of the size of their intersection to the size of their union. The value ranges from 0 (no overlap) to 1 (identical sets). Unlike correlation, which summarizes trends, Jaccard provides an intuitive measure of agreement: it tells us the fraction of elements common to both sets, making it particularly useful when we care about exact matches rather than relationships.

2.5 Dimensionality Reduction

Real-world data, particularly images, often lives in a space of very high dimension. A 32×32 color image, for instance, exists in $\mathbb{R}^{3072}$ —yet we know that not all dimensions are equally informative. Reducing an image’s resolution still allows us to recognize objects, suggesting that the effective degrees of freedom required to represent the data are far lower than the raw pixel count. This observation points to a common assumption underlying dimensionality reduction: that the data contains substantial redundancy, with many features either correlated or irrelevant, and that a lower-dimensional representation can capture most of the information. In other words, the *intrinsic dimension* of the data—the minimum number of parameters needed to describe it without significant loss—is often much smaller than the *extrinsic dimension* of the input space.

The *curse of dimensionality* refers to the various phenomena that arise when analyzing data in high-dimensional spaces. As dimension increases, data becomes sparse; points grow farther apart, and the amount of data needed to maintain the same density grows exponentially. This sparsity makes learning and inference difficult. Yet neural networks manage to operate effectively in such spaces, a fact often explained by the low intrinsic dimension of the data: if the data effectively lies in a lower-dimensional subspace, then the network only needs to contend with that smaller dimensionality. This is precisely why dimensionality reduction is possible, as it aims to find a representation of the data whose dimensionality matches the intrinsic dimension as closely as possible, while losing minimal information in the process.

Principal Component Analysis (PCA) is one of the most widely used linear dimensionality reduction methods. PCA finds a sequence of orthogonal directions, called principal components, that capture maximal variance in the data. The first principal component is the direction along which the data varies most; each subsequent component is orthogonal to the previous ones and captures the next largest remaining variance.

Given a dataset $x_1, \dots, x_N$ with $x_i \in \mathbb{R}^D$, PCA first centers the data by subtracting the sample mean $\hat{\mu}$. It then computes the eigenvectors of the covariance matrix:

$$\Sigma = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})(x_i - \hat{\mu})^\top. \quad (2.18)$$

The eigenvectors $v_1, \dots, v_D$ are the principal components, ordered by their corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$, which quantify the variance captured by each component. In other words, PCA identifies the axes along which the data spreads most (the principal components) and ranks them by the amount of variation they capture (their eigenvalues).

To reduce the dimension to a target $d < D$ (ideally close to the true intrinsic dimension), PCA projects each point onto the first d components:

$$x_i^{\text{PCA}} = [v_1, v_2, \dots, v_d]^\top (x_i - \hat{\mu}) \in \mathbb{R}^d. \quad (2.19)$$

The first d eigenvalues also provide a measure of how much information is retained. The fraction of total variance preserved by the first d components is $\sum_{j=1}^d \lambda_j / \sum_{j=1}^D \lambda_j$, which can guide the choice of d .

Many other dimensionality reduction techniques exist—such as t-SNE, UMAP, and autoencoders—each with different strengths, particularly for capturing non-linear structure. However, they are outside the scope of this thesis, and PCA suffices for the purposes of our analysis.

Chapter 3

How to Estimate Hardness? Unified Taxonomy of Factors Causing Hardness

3.1 Introduction

The concept of hardness is a pervasive and fundamental undercurrent in modern machine learning. It appears, often implicitly, across a vast array of subfields. In curriculum learning, it dictates the order in which examples are presented to a model, starting with the “easy” ones (Wang et al. 2021, Soviany et al. 2022). In data pruning and active learning, it serves as the criterion for identifying which samples are most valuable to retain or query (Settles 2009, Ren et al. 2021, Toneva et al. 2019, Paul et al. 2021, Sorscher et al. 2022). In robust learning, it helps distinguish between noisy, mislabeled examples and genuinely challenging, informative ones (Stutz et al. 2020, Pleiss et al. 2020, Agarwal et al. 2022, Maini et al. 2022, Jia et al. 2023, Yang et al. 2024). Whether it is called “difficulty”, “uncertainty”, “complexity”, or “importance” the underlying notion—that data points are not all equal and that this inequality is a source of valuable information—is central to improving model efficiency, robustness, and performance.

Despite its prevalence, the field of hardness-aware learning is surprisingly fragmented and suffers from a fundamental paradox: a widely used concept built on an unstable foundation. First, there is no commonly accepted formal definition of “hardness.” The term is typically treated as a heuristic (Zhu et al. 2024), operationalized through a diverse set of estimators whose relationships and relative merits are poorly understood. Second, this lack of a unifying framework has resulted in a literature that is highly fragmented. A simple search for “hardness-aware learning” on Google Scholar returns a meager 19 results as of March 2026—a number that starkly underrepresents the concept’s true footprint, as it is often buried as a methodological detail rather than treated as a subject of study in its own right. Consequently, there are no comprehensive literature reviews that map the landscape of hardness-aware learning, and taxonomies for either the estimators used to measure hardness or the factors that cause it are virtually non-existent. A few prior works have proposed such taxonomies: two focus on data-based hardness but are now dated, centering on simple measures like feature overlap rather than incorporating recent insights from geometry

or topology (Lorena et al. 2018, 2019). A more recent model-based taxonomy appears in a benchmarking paper (Seedat et al. 2024), yet its classification of hardness types is largely intuitive and not grounded in a systematic review of the literature, and it does not fully detail how different estimator families relate to one another. Most works remain isolated, with rare cross-references between communities, leading to situations where the same hardness estimator is repeatedly proposed as a novel contribution in different fields. For instance, a method that adjusts hardness using the margin between the highest and second-highest logits was presented as a novel extension of focal loss (Tao et al. 2023), though an essentially identical estimator had already been used two years earlier in the label-noise literature (Paul et al. 2021). Such parallel developments are a natural consequence of fragmentation, and they highlight the difficulty of building cumulatively on prior work. The sole benchmarking paper on hardness-aware learning, published at ICLR in 2024, explicitly acknowledges many of these challenges (Seedat et al. 2024). In over a decade of scattered research, such a systematic treatment remains a rare exception.

This fragmentation is not accidental; it stems from a prevailing mindset that treats hardness as a tool rather than an object of study. For most practitioners, hardness is simply a means to an end—a component in a pipeline for curriculum learning, noise detection, or data pruning. When the focus is on the downstream task, the intricacies of the hardness estimator itself are often overlooked. Why invest effort in understanding the fundamental properties of hardness when it is merely a step in achieving a more “important” objective? While pragmatic, this perspective has led to a fractured field where it is exceedingly difficult to assess the state-of-the-art, compare methods, or build upon prior work. The lack of a shared vocabulary and a coherent knowledge structure stifles progress, causing research to be duplicated and insights to remain trapped within sub-communities.

The aim of this chapter is to provide a structured entry point into hardness-aware learning. Rather than attempting an exhaustive review of all related fields, We focus on two core questions: *how is hardness estimated?* and *what factors cause it?* To answer these, we draw on a selection of representative works from areas such as curriculum learning, data pruning, active learning, and noise identification—fields where hardness estimators are most commonly used. Based on this material, we develop a set of taxonomies intended to help readers orient themselves in the current landscape.

Specifically, the chapter makes the following contributions:

- **A distinction between data-based and model-based hardness.** We separate hardness estimators into those that treat hardness as an intrinsic property of a data point and those that view it as emerging from the interaction between a data point and a model.
- **A taxonomy of data-based hardness estimators.** For the data-based stream, we categorize the most common estimators according to their underlying principles.
- **Taxonomies of factors causing hardness.** We propose two taxonomies: one derived from the data-based literature and one from the model-based literature. The two taxonomies are presented together in 3.1, which illustrates how the factors relate to each other.

The main objective is to offer readers—whether new to hardness-aware learning or already working in one of its application areas—a clearer picture of how hardness is estimated and

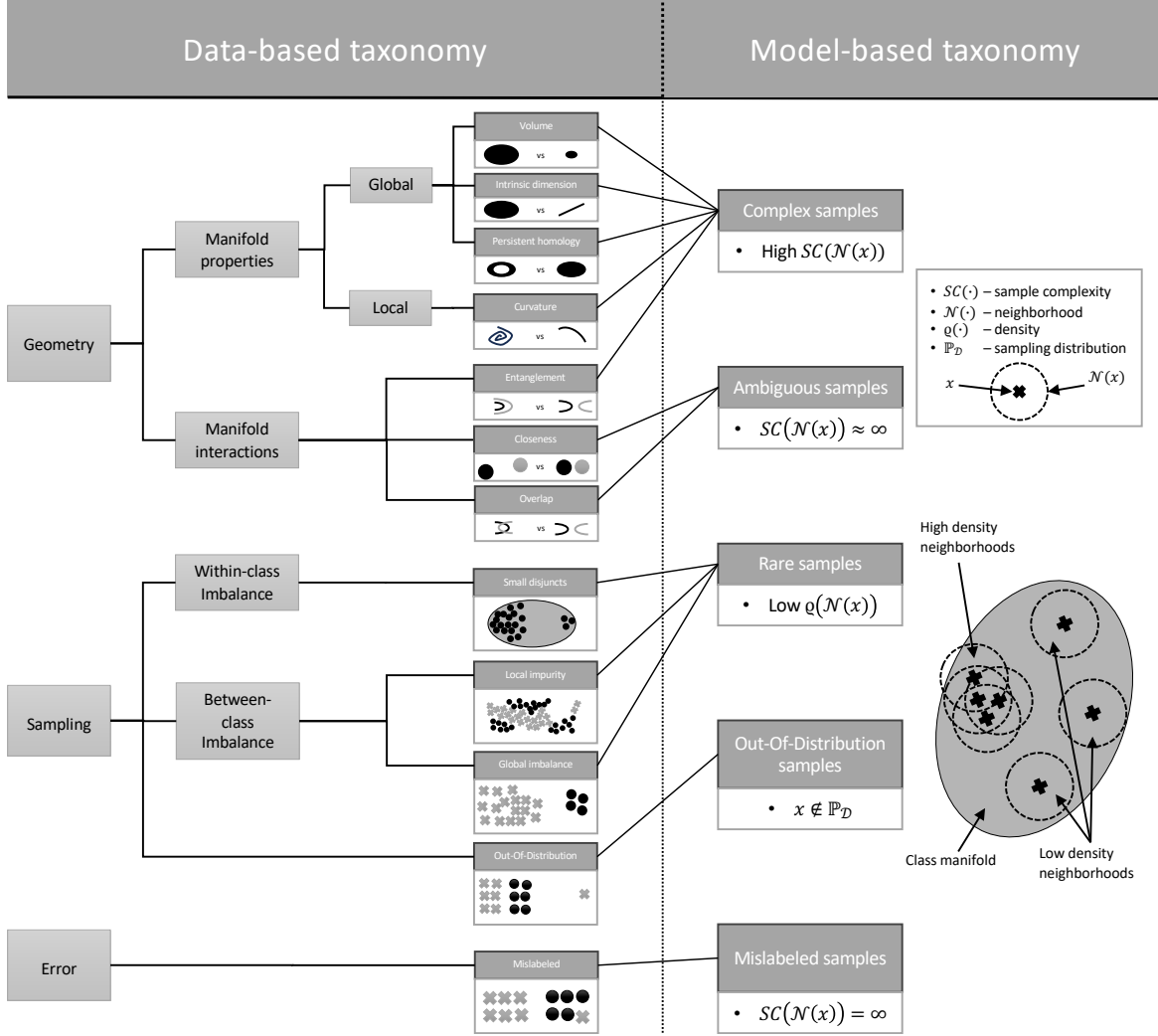


Figure 3.1: We propose two complementary taxonomies: one derived from the literature on data-based hardness estimators, and one derived from the literature on model-based hardness estimators. The figure presents both taxonomies together and illustrates how the factors from each perspective relate to one another.

what factors give rise to it. By organizing existing approaches in a way that is largely agnostic to downstream tasks, we hope to make it easier to navigate the literature, compare methods, and build on prior work more systematically.

3.2 Objective and Subjective hardness

We ground our categorization of hardness estimators—into model-based and data-based—in an intuitive, real-world understanding of hardness. Every person is familiar with the notion of difficulty: in formal education, exam questions vary in how easily they can be solved. Some are answered almost effortlessly; others demand complex reasoning. Students often prioritize easier topics when studying, postponing harder material in the hope it will not be tested.

Importantly, this perception of difficulty is subjective: what one student finds easy, another may find challenging. Geometry might be intuitive for one learner but daunting for another, while probability poses the opposite pattern. Nevertheless, some aspects of difficulty are widely shared: most would agree that computing $\frac{d}{dx} x^2$ is simpler than $\frac{d}{dx} \frac{\sin^2(x^3)}{\sin^3(x^2)}$.

A similar phenomenon emerges in machine learning. When training neural networks on classification tasks, certain samples are consistently learned more quickly than others, often across different architectures—suggesting that some aspects of hardness are intrinsic to the data. At the same time, hardness is also shaped by the model: different architectures induce different inductive biases and representations, so a sample that is easy for one network may be harder for another. In other words, hardness has both a data-intrinsic component and a model-dependent component. Importantly, the degree to which model-dependent biases influence hardness estimates remains underexplored. It is not yet clear whether architectural differences introduce only minor artifacts or fundamentally alter which samples are considered hard. Despite this uncertainty, the conceptual distinction between data-intrinsic and model-dependent hardness provides a useful organizing principle.

Motivated by this dual nature, we propose a high-level categorization of hardness estimators (and the works that use them) into two families:

- **Data-based estimators** quantify the inherent difficulty of a sample by analyzing properties of the data alone—such as geometry, dimensionality, or feature-space imbalances—independently of any model.
- **Model-based estimators** quantify hardness by examining the model’s behavior during training or inference, using signals like confidence, uncertainty, forgetting events, or learning dynamics. These estimators capture the effects of hardness as they manifest in the interaction between data and model.

These two families are almost always treated separately in the literature. Research either focuses on how model-based estimators inform downstream tasks (pruning, curriculum learning, label-noise detection) or explores geometric properties of data manifolds to motivate learning criteria. Consequently, existing taxonomies of hard samples are built primarily from one perspective—either model-based (Seedat et al. 2024) or data-based (Lorena et al. 2018, 2019)—with little effort to bridge the two. This is something we try to address in this chapter.

3.3 Data-Based Hardness Estimators

There exist multiple foundational perspectives from which neural networks and classification problems can be studied. Among the most prominent are:

- The **geometric** perspective, which relies on the manifold hypothesis and emphasizes how data and decision boundaries are shaped and transformed across layers (Olah 2014, Naitzat et al. 2020).
- The **probabilistic or Bayesian** perspective, which treats model outputs as estimates of uncertainty and approaches learning as inference under uncertainty (Lampinen & Vehtari 2001, Magris & Iosifidis 2023).

- The **information-theoretic** perspective, which seeks to quantify and understand learning in terms of information compression, mutual information, and entropy (Tishby & Zaslavsky 2015, Kawaguchi et al. 2023).

These viewpoints offer complementary insights, each highlighting different facets of what it means for a data instance to be “learned” or “difficult.” In this thesis, we adopt the geometric perspective, grounding our discussion in the manifold assumption.

3.3.1 Manifold Assumption and its Implications on Learning

Many datasets in machine learning are high-dimensional. For instance, each color image in CIFAR-10 (Krizhevsky et al. 2009) is represented as a point in $\mathbb{R}^{3072}$, and a single gene expression profile may span tens of thousands of features. However, the number of available training samples is often similar or even orders of magnitude smaller than the number of features. How, then, do deep neural networks manage to learn effectively in such settings, seemingly sidestepping the curse of dimensionality?

One popular explanation is the manifold hypothesis (Narayanan & Mitter 2010, Fefferman et al. 2016, Berner et al. 2021). According to this view, real-world data, while technically residing in high-dimensional spaces, actually occupy only a tiny, structured subset of that space—a manifold of much lower intrinsic dimensionality. Thus, while the data representation is high-dimensional, its structure is not. In other words, we can consider classes not as a collection of data samples but rather as structures that have certain geometry and topology. More formally, we would say that for a given class, data is sampled from a data distribution the support of which is a manifold with lower intrinsic dimensionality than the extrinsic dimension. This shifts our focus from individual data points to the underlying manifolds themselves, treating each class as a distinct geometric object, with its own properties like intrinsic dimension or curvature, rather than a mere set of examples.

In the input space, these class manifolds are entangled—they are interwoven in complex, non-linear ways that prevent linear separation. This entanglement is what makes classification challenging in the original representation. Our goal, then, is to transform this entangled configuration into a clean, disentangled structure in the output space. As established in Chapter 2, the output of a neural network is a vector of floats—one per class—representing the model’s confidence that the input belongs to each class. The ideal behavior is that the model is perfectly confident and correct for every sample. Mathematically, this ideal corresponds to the outputs for all samples of a given class collapsing to a single point—the vertex corresponding to that class—with these vertices forming a perfect K -simplex in the output space.

A neural network provides the mapping between the input and output spaces. It is composed of a series of linear and non-linear functions, each of which can be understood geometrically as a transformation that stretches, rotates, and folds the space (Olah 2014). As samples pass through the network, their representation is transformed layer by layer, gradually reshaping the corresponding manifolds. The objective is that, by the time it reaches the output layer, the configuration of these manifolds approximates the ideal simplex.

Learning is the process of finding the specific transformations that achieve this. This is accomplished through a loss function—in our case, cross-entropy—and backward propagation. The cross-entropy loss is defined as $\mathcal{L} = -\log p_{\hat{y}}$. Minimizing it forces the centroids

of the class manifolds to align with the standard basis vectors (the vertices of the simplex) and the manifolds themselves to collapse to points (Weinan & Wojtowysch 2022). Thus, the learning objective becomes geometric: we seek the set of transformations that bring us as close as possible to a perfect simplex in the output space. This perspective is backed by empirical research, which showed that:

- phenomenon of class manifolds collapsing onto standard basis vectors, termed *neural collapse*, emerges in overparametrized regime when training beyond zero error, and towards zero loss (Papayan et al. 2020, Kothapalli 2023);
- geometric and topological features of class manifolds get simplified as they pass through the layers of a well-trained network, with no such pattern emerging for untrained networks (Brahma et al. 2015, Naitzat et al. 2020, Magai 2023, Suresh et al. 2024);
- the distance of the data sample to its closest latent class centroid is a good estimate of hardness (Paul et al. 2021).

This perspective naturally implies the existence of class-level hardness. Intuitively, some classes will be harder to disentangle from others and to collapse onto the corresponding standard basis vector. This intuition is supported by theoretical work on sample complexity—the number of samples required for a learning algorithm to achieve high accuracy with high probability. For binary classification, Narayanan & Niyogi (2009) proved that sample complexity depends on the intrinsic dimension of the data manifold and the curvature of the decision boundary separating the classes. In the context of our framework, this can be interpreted as showing that the intrinsic dimension and curvature of class manifolds, as well as their interaction (entanglement, closeness, and overlap) with each other directly affect how many examples are needed to learn to separate them (class-level hardness).

The existence of class-level hardness naturally gives rise to dataset-level and instance-level notions of hardness. A dataset is hard if it contains a substantial number of hard classes; a data sample is hard if it belongs to a hard class or, as a plausible hypothesis, if it lies in a region of a class manifold that exhibits high geometric or topological complexity (e.g., close to the decision boundary, in high-curvature zones, or near topological singularities). In the remainder of this section, we survey the literature on data-based estimators and construct a knowledge map to identify which geometric and topological properties have been proposed or empirically shown to influence learning hardness.

3.3.2 Dimensionality-Based Estimators

Ansuini et al. (2019) studied how the intrinsic dimensionality (ID) of data representations evolves across layers of a deep neural network. They observed that for trained models, the ID tends to increase in early layers and decrease in later layers, particularly in the output layer, where it strongly correlates with test accuracy. This suggests that first few epochs aim at disentangling class manifolds, increasing ID of the dataset in the process, and later ones focus on collapsing class manifolds onto corresponding standard basis vectors—a pattern consistent with findings of Ciceri et al. (2024). Importantly, this observation didn’t hold for untrained networks. They also find that the ID of the latent representation of the dataset from the last hidden layer correlates strongly with classification performance.

To estimate the ID, they employ the Two Nearest Neighbors (TwoNN) method of Facco et al. (2017), which relies on the assumption that data points are locally uniformly sampled from a d -dimensional manifold and can be approximated by a Poisson point process. In such a manifold, the number of points within a ball of radius r scales as $N(r) \propto r^d$, implying that for each data point x_i , the ratio $\mu_i = T_2(x_i)/T_1(x_i)$, where $T_j(x_i)$ is the distance from x_i to its j -th nearest neighbor, is a realization of a random variable μ that follows a Pareto distribution:

$$p(\mu) = d \cdot \mu^{-(d+1)}. \quad (3.1)$$

Based on this distributional form, the intrinsic dimension d can be estimated via maximum likelihood over a collection of such μ_i ratios. In practice, one computes these ratios for each sample in the dataset and fits the Pareto distribution using maximum likelihood, resulting in a closed-form estimator for the global intrinsic dimension.

$$ID(\mathbf{X}) = \left(\frac{1}{n} \sum_{i=1}^n \log \mu_i \right)^{-1}, \quad (3.2)$$

where $\mu \geq 1$ and $X = \{x_i, \dots, x_n\}$ denotes the set of data points whose intrinsic dimension is being estimated. Alternatively, one may use a localized version by applying the same procedure within individual classes, as we later do.

Pope et al. (2021) extend this line of inquiry by focusing on estimating the intrinsic dimension of raw input datasets rather than latent representations. They find that the estimated ID of a dataset is negatively correlated with state-of-the-art classification accuracy, reinforcing the notion that lower-dimensional manifolds are easier to learn. Moreover, they empirically validate the theoretical predictions of Narayanan & Niyogi (2009), showing that higher intrinsic dimensionality increases the number of samples required to achieve a given generalization error, while the extrinsic dimension has minimal influence. To assess the utility of ID estimation methods, they conduct experiments on synthetic datasets generated via GANs, which allow precise control over intrinsic dimensionality. For estimation, they adopt the maximum likelihood approach of Levina & Bickel (2004), which models the count of neighboring points within a fixed radius as a Poisson process under local uniformity and Euclidean assumptions. Specifically,

$$ID(x_i) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \left(\frac{T_k(x_i)}{T_j(x_i)} \right) \right]^{-1}. \quad (3.3)$$

This local estimate is later averaged over all data samples, providing a global estimate. The choice of k controls the bias–variance tradeoff: small k leads to high-variance estimates due to noise, while large k can introduce bias by violating the local Euclidean assumption—since distant neighbors may lie outside the locally linear region of the manifold.

Ma et al. (2024) build on the above findings by analyzing intrinsic dimensionality at the class level. They show that the negative correlation between intrinsic dimension and classification accuracy not only holds at the dataset level, as shown by Pope et al. (2021), but also at the level of individual classes. Specifically, they find that classes whose latent representations occupy lower-dimensional manifolds tend to have higher classification accuracy. This

class-level correlation is most pronounced when using representations from deeper layers of the network, although they do not consider the input layer in their experiments, in contrast to Pope et al. (2021) who measure ID directly in input space. These findings are consistent with the geometric perspective adopted in this thesis: a higher ID in the final hidden layer suggests that the network has difficulty in collapsing the corresponding class manifold toward its associated standard basis vector, indicating class hardness. Ma et al. argue that during training, networks implicitly learn space transformations that reduce ID imbalance across classes. To improve this rebalancing process, they propose Intrinsic Dimension Regularization (IDR), which penalizes high-ID class representations during training. They estimate ID of each class through the following formula:

$$\text{ID} = \frac{(\text{tr}(\Sigma))^2}{\text{tr}(\Sigma^2)}, \quad (3.4)$$

where Σ is the covariance matrix of the class’s latent representations. This quantity approximates the number of significant directions in which local variance is distributed. An estimated intrinsic dimension of k implies that variance is spread across approximately k orthogonal directions, suggesting that the manifold locally behaves as k -dimensional. Their regularization loss is defined as:

$$L_{\text{ID}} = \sum_{c=1}^K -\log \left(\frac{ID_c^{-1}}{\max\{ID_1^{-1}, \dots, ID_K^{-1}\}} \right), \quad (3.5)$$

with K being the number of classes. In practice, this term is smoothed and scaled before being added to the training objective (see the original paper for details). Empirically, they show that IDR reduces class-level imbalance by improving performance on hard, high-ID classes, with only minor degradation on easier ones.

3.3.3 Volume-Based Estimators

Since intrinsic dimension has been shown, both theoretically and empirically, to correlate with hardness, it is natural to expect similar trends for Lebesgue measure of the region occupied by the data—colloquially referred to as *volume* in the literature—which often increases with intrinsic dimension. Ma, Jiao, Liu, Li, Yang & Liu (2023) explore this connection and show that the volume associated with each latent class manifold, when appropriately scaled, is negatively correlated with classification accuracy. This complements their findings in Ma et al. (2024), where higher intrinsic dimensionality was similarly associated with lower accuracy. In their experiments they define a measure called *semantic scale* for each class, which captures the volume of the class manifold weighted by its average distance to the centroids of other classes. They estimate the volume itself via:

$$V_c = \frac{1}{2} \log_2 \det(I + \Sigma) = \frac{1}{2} \log_2 \det \left(I + \frac{D}{m} (Z_c - \bar{Z}_c)(Z_c - \bar{Z}_c)^T \right), \quad (3.6)$$

where m is the number of samples in class c , D is the dimensionality of the latent representation space, Z_c is the $D \times m$ matrix whose columns are the latent representations of samples belonging to class c , $\bar{Z}_c$ is the $D \times m$ matrix with each column equal to the mean vector of

the class, computed as the average across the m samples, and $\Sigma = \frac{D}{m}(Z_c - \bar{Z}_c)(Z_c - \bar{Z}_c)^T$ is the scaled covariance matrix of the class representations. The determinant of the covariance matrix estimates the volume of the corresponding covariance ellipsoid — the ellipsoid that captures the spread of data around the class mean. This ellipsoid may provide a coarse geometric approximation of the class manifold. However, it tends to significantly overestimate the true manifold volume when the manifold is highly curved or has an intrinsic dimension substantially lower than the extrinsic dimension.

As in their work on intrinsic dimension, the authors propose a regularization term that penalizes high semantic scale, and show that it reduces performance gaps across classes on both balanced and imbalanced datasets. However, due to the absence of ablation studies, it remains unclear whether these improvements arise specifically from volume compression, controlled inter-class distances, or a combination of both. Finally, they observe that both semantic scale and accuracy grow together as new data is added, but begin to plateau simultaneously. While the authors do not offer an explanation for this phenomenon, we hypothesize that it reflects a decline in the informational uniqueness of newly added samples. In the early stages of training, each additional sample tends to introduce novel geometric variation, making class separability easier. As the dataset grows, however, most new samples are drawn from already well-represented regions of the manifold, offering limited additional structure. This results in diminishing returns with respect to both accuracy and semantic scale, highlighting the need to consider the quality of data when oversampling.

3.3.4 Topology-Based Estimators

Another characteristic of the complexity of class manifolds is their *persistent homology*, which captures topological features that persist across scales. These features are quantified by *Betti numbers* β_n , which count the number of n -dimensional holes in the data. For example, β_0 represents the number of connected components, β_1 counts 1-dimensional loops (as in a ring), β_2 corresponds to 2-dimensional voids (such as the hollow of a torus), and so on.

To compute Betti numbers in practice, one first constructs a *simplicial complex* on top of the data. The most commonly used construction is the *Vietoris–Rips complex*, where a k -simplex is formed among $k+1$ data points if all pairwise distances are below a fixed threshold t . A k -simplex is the k -dimensional generalization of a triangle: a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a filled-in triangle, a 3-simplex is a tetrahedron, and so on. To capture topological changes across scales, a *filtration* is applied: a nested sequence of complexes $K_0 \subseteq K_1 \subseteq \dots \subseteq K_{\mathcal{T}}$ is formed by gradually increasing the distance threshold t . Persistent homology then tracks the birth and death of topological features—e.g., connected components and loops—as t varies, resulting in a *persistence diagram* or *barcode* summarizing the lifetime of each feature (see Figure 3.2 for visualization).

Naitzat et al. (2020) observed that neural networks simplify the topological structure of class manifolds as data propagates through layers. Specifically, they show that the Betti numbers decrease progressively with depth, indicating a topological simplification aligned with the geometric perspective. This effect is more pronounced when using ReLU activations compared to tanh, potentially explaining why ReLU networks tend to train more effectively. They further find that network depth affects the distribution of topological simplification: in shallow networks, most of the reduction in Betti numbers occurs in the final layers, whereas in deeper networks, the reduction is more gradual and distributed across layers. This insight

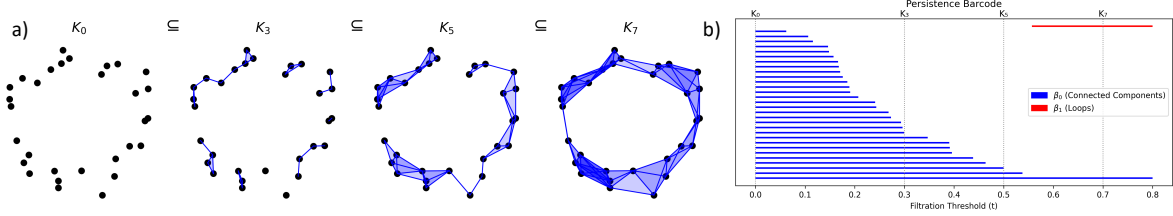


Figure 3.2: (a) A sequence of Vietoris-Rips (VR) complexes constructed on a 2D point cloud by increasing the pairwise distance threshold t . A k -simplex is added whenever all $\binom{k+1}{2}$ edges among $k+1$ points have length $\leq t$. This results in a growing complex: as t increases, new edges and higher-order simplices (e.g., filled triangles) appear, yielding a nested sequence $K_0 \subseteq K_3 \subseteq K_5 \subseteq K_7$. (b) The barcode summarizes the persistence of topological features across this filtration: each bar represents the birth and death of a feature. For example, a loop feature is absent in K_5 but appears in K_7 , indicating that its birth lies within the interval between these two filtration steps. Dotted lines in (b) correspond to the thresholds shown in (a).

provides a possible explanation for why deeper architectures can outperform shallow ones, despite the universal approximation theorem guaranteeing representational capacity in both cases.

Magai (2023) expand the findings of Naitzat et al. (2020) by showing that topological simplification of data manifolds is a consistent phenomenon across a wide range of architectures, including convolutional neural networks, vision transformers, ConvMixers, and large language models. Instead of analyzing individual Betti numbers, they use a scalar summary statistic, PH_{dim} , which they show empirically to be an effective estimator of intrinsic dimensionality, thereby connecting persistent homology with intrinsic dimension. They find a strong negative correlation between test accuracy and PH_{dim} of the last layer’s manifold representations.

These insights align with the geometric perspective introduced in 3.3.1. In a well-trained network, Betti numbers decrease across layers because class manifolds are progressively collapsed onto the standard basis vectors. This is why it is common in the literature to describe class manifolds as being “simplified” as they propagate through the layers of a well-trained network. Nevertheless, whether the topological properties of class manifolds captured by Betti numbers correlate with class-level hardness remains an open question.

3.3.5 Curvature-Based Estimators

Kienitz et al. (2022) conducted an empirical study to assess how geometric properties such as entanglement, intrinsic dimensionality, and extrinsic dimensionality affect sample complexity in classification tasks. Using both synthetic and real-world datasets, including SVHN (Netzer et al. 2011) and CIFAR-10, they found that entanglement between class manifolds was the strongest predictor of increased sample complexity. In contrast, extrinsic dimensionality had negligible influence, while intrinsic dimensionality only affected learning difficulty in datasets that already exhibited high entanglement.

To estimate entanglement and curvature, the authors propose two proxies. The first is the classification accuracy of a linear support vector classifier (LSVC), which serves as a global

measure of linear separability. The second is based on the distribution of singular values of the Hessian obtained by fitting second-order Taylor approximation to the decision boundary. This approximation quantifies how sharply the decision boundary bends around those points, capturing the degree to which the boundary locally deviates from linearity. While this Hessian-based curvature score is more geometrically informative, it is also significantly more computationally expensive. The authors claim that LSVC accuracy and the Hessian-based metric generally agree on the relative complexity of datasets, despite some noticeable differences, and advocate LSVC as a computationally cheaper alternative.

Crucially, both estimators reflect a blend of decision boundary curvature, which can serve as a proxy for *curvature of class manifolds*, and *class entanglement*. However, because these factors are not explicitly separated in either estimator, it remains difficult to determine whether a given hardness score arises from intrinsic manifold complexity or from entanglement between class manifolds. Subsequent works address this limitation by proposing methods that isolate curvature-specific effects more directly.

Isolating Curvature from Entanglement

Kaufman & Azencot (2023) investigate how the curvature of data representations evolves through the layers of deep neural networks, measuring curvature at the dataset level. They observe a characteristic pattern where curvature significantly increases in the initial layers, remains relatively stable through the middle layers, and increases again in the final layer. Notably, the difference in curvature between the penultimate and output layer correlates strongly with model accuracy. The underlying reasons for this pattern remain an open question, and likely require further study incorporating class-level analyses of curvature changes, which, by definition, provide more informative insights than dataset-level estimates.

To estimate the curvature, the authors adopt the Curvature Aware Manifold Learning (CAML) method, introduced by Li (2018). The methodology is conceptually related to that of Kienitz et al. (2022), with two key differences. First, the Hessian is obtained by fitting a second-order Taylor approximation to the embedding map $f : \mathbb{R}^d \rightarrow \mathbb{R}^D$, which represents the underlying projection from latent class manifolds to the input space. This map is precisely the object that decoders in generative models such as autoencoders or VAEs attempt to approximate. The local approximation of f at a point x is constructed by projecting nearby points into a tangent space centered at x , and spanned by the top d singular vectors, obtained from the SVD of the centered neighborhood matrix, where the intrinsic dimension d is approximated using TwoNN.

Second, they analyze the eigenvalues of the Hessian rather than its singular values. The use of eigenvalues enables them to capture both the magnitude and the sign of curvature—distinguishing between convex and concave directions—whereas singular values used by Kienitz et al. (2022) reflect only the absolute curvature. This approach is, in a sense, more mathematically principled, as it targets the curvature of the manifold itself rather than that of the decision boundary. However, it also necessitates reliance on finite observed data, whereas working with the decision boundary allows sampling additional points to improve the approximation’s fidelity.

Ma, Jiao, Liu, Yang, Liu & Li (2023) study how three geometric quantities — class separation, manifold volume, and curvature — evolve throughout training, and how they influence model accuracy and generalization. They report that both (1) the positive cor-

relation between accuracy and class separation, and (2) the negative correlation between accuracy and curvature become stronger as training progresses. In addition, the separation between class manifolds increases over epochs, lending further support to the geometric perspective on learning. Interestingly, they observe a significant drop in curvature during the early epochs of training, followed by a more gradual decline. This trend complements the findings of Kaufman & Azencot (2023), who examined how curvature evolves across layers of a trained model. However, it remains unclear whether generalization is better predicted by the final curvature of the latent representation or by the curvature gap used by Kaufman et al. Finally, to encourage flatter class manifolds, Ma, Jiao, Liu, Yang, Liu & Li (2023) introduce a curvature-based regularization term:

$$L_{Curvature} = \sum_{c=1}^K -\log \left(\frac{G_c^{-1}}{\max\{G_1^{-1}, \dots, G_K^{-1}\}} \right), \quad (3.7)$$

where G_i is the estimated Gaussian curvature of the manifold of class i . As with their volume- and dimension-based regularizers, it was empirically showed that this term reduces the generalization gap.

While the overall curvature estimation framework of Ma, Jiao, Liu, Yang, Liu & Li (2023) follows the same high-level structure as that of Kaufman & Azencot (2023), the implementation differs slightly in the choice of tangent space dimensionality. Specifically, Ma et al. estimate the tangent space at each point using PCA with a fixed number of components, rather than adapting it to the locally estimated intrinsic dimension. This results in a curvature estimate that may be less faithful to the local geometry of the data manifold, and introduces additional hyperparameter.

3.3.6 Neighborhood-Based Estimators

All the above-mentioned families of hardness estimators focus on measuring the complexity of the class manifolds. However, hardness can also emerge due to sampling related issues which cannot be captured by measuring the properties of the manifolds. To measure the hardness that emerges from sampling-related issues neighborhood-based hardness estimators are the most optimal, as they assume that a sample's difficulty is reflected in its distances to nearby samples and the surrounding class distribution. Several simple yet effective metrics include:

$$d_{\text{avg}}^{\text{same}}(x) = \frac{1}{|\mathcal{N}_k^{\text{same}}(x)|} \sum_{x' \in \mathcal{N}_k^{\text{same}}(x)} \|x - x'\|_2, \quad (3.8)$$

$$d_{\text{avg}}^{\text{other}}(x) = \frac{1}{|\mathcal{N}_k^{\text{other}}(x)|} \sum_{x' \in \mathcal{N}_k^{\text{other}}(x)} \|x - x'\|_2, \quad (3.9)$$

where $\mathcal{N}_k^{\text{same}}(x)$ and $\mathcal{N}_k^{\text{other}}(x)$ denote the k nearest neighbors of x from the same and different classes, respectively. $d_{\text{avg}}^{\text{other}}(x)$ can be interpreted as a proxy for distance to the decision boundary. Hence, the lower it is the harder the sample. On the other hand, $d_{\text{avg}}^{\text{same}}(x)$ serves as a proxy for sample rarity - rare samples tend to have few nearby same-class neighbors, making them more difficult. It is important to remember that in high-dimensional settings, the density of points per dimension becomes so sparse that such local distance-based metrics

often lose their discriminatory power, making such simplistic neighborhood-based hardness estimators unreliable.

Another commonly used neighborhood-based hardness estimator is *class impurity*, which quantifies the proportion of a sample’s neighbors that belong to different classes. Formally:

$$\text{Class Impurity}(x) = \frac{|\mathcal{N}_k^{\text{other}}(x)|}{|\mathcal{N}_k^{\text{same}}(x)| + |\mathcal{N}_k^{\text{other}}(x)|}. \quad (3.10)$$

While class impurity draws on similar signals as $d_{\text{avg}}^{\text{same}}(x)$ and $d_{\text{avg}}^{\text{other}}(x)$, it combines them into a single, normalized score, making it more interpretable in practice - ADASYN (He et al. 2008), a very popular oversampling method, selects minority samples with high class impurity for oversampling. In an idealized setting, high class impurity would indicate strong entanglement between class manifolds, reflecting intrinsic geometric ambiguity and closeness to other class manifolds. However, in real-world datasets, elevated class impurity is more often a consequence of sampling effects. Specifically, the local sampling density from the same class may be significantly lower than that from other classes, leading to high impurity despite well-separated manifolds. In this context, class impurity functions as a proxy for relative sample rareness, which may arise either locally or globally. The latter case corresponds to well-known problem of between-class data imbalance (Johnson & Khoshgoftaar 2019).

3.4 Providing Geometry-Based Taxonomy of Hard Samples

In the previous section, we reviewed several key families of data-based hardness estimators, uncovering a wealth of insights into what makes samples hard to learn and classify. Here, we summarise these findings and identify three core sources of sample-level hardness, which we outline below.

3.4.1 Geometry- and Topology-Related Hardness

The geometry and topology of latent class manifolds, particularly those in the final (output) layer representations of trained neural networks, have been shown to correlate with classification accuracy. Specifically, lower intrinsic dimension, volume, and curvature of latent class manifolds correlate with higher model accuracy. In addition, several studies have shown that both intrinsic dimension and topological complexity decrease as data flows through the layers of a trained neural network, backing the geometric perspective on inference as a process of disentangling class manifolds and collapsing them onto corresponding standard basis vectors. As far as we know, no analogous study currently exist for volume, whereas for curvature the only known study examining evolution of this property over layers estimates it at the dataset level, rather than the class level. Since curvature is a class-specific property, dataset-level aggregation conducted by Kaufman & Azencot (2023) is hard to interpret and provides limited practical insights. Thus, while there is no empirical evidence behind reduction of the curvature of class manifolds as they pass through layers, this assumption aligns with the geometric perspective.

This pattern of simplifying geometric and topological properties of class manifolds through forward propagation implies that high geometric or topological complexity in the raw (input-level) manifolds may be an underlying cause of poor performance, as the higher the input

complexity the more difficult it is for the network to simplify these structures through forward propagation. In other words, we believe that it is natural to assume that complexity of class manifolds in the raw and latent spaces should strongly correlate - a class that is the most complex in the input space is unlikely to become the simplest one in the latent space. However, empirical studies show little to no correlation between these raw manifold properties and final model accuracy. We hypothesize that this discrepancy is largely due to the curse of dimensionality. Most methods for estimating manifold properties rely on Euclidean assumptions that hold only locally and become unreliable in high-dimensional input spaces with limited data density. Our hypothesis is supported by two key observations. Firstly, Pope et al. (2021) found that intrinsic dimension measured at the dataset level allows ranking datasets in a way that aligns with SOTA performance, suggesting that a similar relationship may hold at the class level. Secondly, Narayanan & Niyogi (2009) proved that the sample complexity of binary classification depends on the intrinsic dimension and curvature of class manifolds, suggesting that the absence of empirical correlation is an artifact of estimation, rather than a true absence of effect.

A parallel conceptual argument can be made for topology. The geometric perspective on deep learning suggests that forward propagation progressively collapses class manifolds onto the standard basis vectors, effectively simplifying their topological structure. It is therefore plausible that a class manifold with high Betti numbers—meaning it possesses numerous or high-dimensional holes—should be inherently harder to classify. We hypothesize that this hardness stems from two interconnected reasons. First, the presence of non-trivial topology (loops, voids, etc.) imposes additional constraints on the transformations needed to collapse the manifold to a topologically trivial point, making the optimization problem more complex. Second, a manifold with holes can become entangled with other class manifolds in ways that a simply connected manifold cannot: while overlapping manifolds merely share a region, a manifold with a hole can have another manifold passing through it, creating a form of topological entanglement that is more difficult for a network to disentangle. Although this hypothesized link between Betti numbers and class-level hardness has not yet been empirically or theoretically confirmed, it offers a compelling motivation for including persistent homology among the candidate complexity properties in the categorization below.

In addition to individual manifold properties, a second key contributor to hardness arises from interactions between class manifolds. Class manifolds do not exist in isolation; they occupy the same Euclidean space, and as the number of classes grows, the complexity of their interactions increases. Three forms of interaction have been linked to increased classification difficulty: closeness, overlap, and entanglement. Closeness refers to manifolds that lie near one another, making them easily confusable in the presence of noise. Overlap occurs when manifolds share a region of space, forcing the network to resolve ambiguous points (e.g., heterogeneous outcomes in medicine). Entanglement, in this context, describes a situation where manifolds are interwoven in a way that prevents them from being separated by a linear or low-complexity decision boundary—for example, when one manifold winds around another without intersecting it. Empirical studies have shown that these forms of interaction correlate with higher classification difficulty, underscoring that hardness is not solely a function of each class manifold’s intrinsic properties, but also of how manifolds relate to one another. Together, these insights motivate the following categorization:

1. Geometry-related issues causing hardness
 - (a) Issues emerging due to properties of class manifolds themselves
 - i. Global properties of class manifolds
 - A. Volume
 - B. Intrinsic dimension
 - C. Persistent Homology
 - ii. Local properties of class manifolds
 - A. Curvature
 - (b) Issues emerging due to interaction between class manifolds
 - i. Closeness between class manifolds
 - ii. Entanglement of class manifolds
 - iii. Overlap between class manifolds

3.4.2 Sampling-Related Hardness

Another core source of hardness stems from sampling related issues. These issues can be viewed from two perspectives. When considering individual class manifolds, we can notice that in practice the data is never uniformly sampled from it. This naturally leads to emergence of within-class data imbalance, characterized by existence of small disjuncts—isolated subclusters of a class that are sparsely represented in the data (Jo & Japkowicz 2004). Such regions pose a challenge because the network must learn to correctly collapse these sparse submanifolds while having limited evidence to guide the transformation. Despite its intuitive importance, within-class sampling imbalance remains severely under-researched compared to its between-class counterpart (Leevy et al. 2018).

Just as with geometric hardness, where difficulties arise both from properties of individual class manifolds and from interactions between them, sampling-related hardness similarly stems from two sources. The second source concerns sampling inconsistencies across class manifolds. At the global scale, this is the well-known problem of between-class imbalance, where one class overall has many more samples than another. This form has received extensive attention in the literature (Makki et al. 2019, Fotouhi et al. 2019). However, sampling inconsistencies can also manifest locally: even when the overall dataset is balanced, there can be neighborhoods where the sampling density of one class dominates another. We conjecture that such local relative imbalance should also affect classification hardness. For instance, a small disjunct of class A surrounded by a densely sampled region of class B is likely more challenging to learn than the same disjunct isolated from other classes, because the decision boundary must carve out the sparse region while being pulled by the dense neighboring class. While the concept of local label mixing is captured by measures such as class impurity, systematic empirical investigations that isolate the effect of local relative imbalance—separate from global imbalance or other confounding factors—remain scarce. The practical severity of this phenomenon is therefore not yet well understood, and we raise it here as a plausible but under-explored contributor to hardness. Between the global and local extremes lie intermediate scales, where measures such as class impurity can capture the degree of local label mixing.

As discussed in Section 3.3.6, the hardness of a data sample can often be inferred from its distance to nearby samples from the same class, with larger distances indicating increased rareness. However, it is important to distinguish between two sources of such rarity. The first arises from non-uniform sampling within the data-generating distribution itself, leading to underrepresented regions on the class manifold. Samples from such regions are typically referred to as anomalies or rare in-distribution (ID) examples. The second arises when the sample is drawn from a different sampling distribution altogether, such that its support does not fully overlap with the support of the training data. These out-of-distribution (OOD) samples may still belong to the same semantic class (e.g., cat images from ImageNet versus CIFAR-10), but differ in visual style, resolution, or context. From a geometric perspective, this implies that neither sampling distribution fully covers the underlying class manifold. While both types of samples may appear rare under nearest-neighbor analysis, we argue that their origins and implications are fundamentally distinct, and thus merit separate treatment in our taxonomy. Together, these insights lead to the following taxonomy:

1. Sampling related issues causing hardness
 - (a) Issues due to sampling inconsistencies within class manifold
 - i. Within-class data-based imbalance (small disjunct problem)
 - (b) Issues due to across class sampling inconsistencies
 - i. Relative data-based imbalance
 - A. Local (measured by class impurity)
 - B. Global (between-class data-based imbalance measured by imbalance ratio between classes)
 - (c) Issues due to insufficient support of the main sampling distribution
 - i. Out-Of-Distribution samples

3.4.3 Error-Related Hardness

The previous two taxonomies fail to take into consideration various practical issues that can emerge in the real world such as noise, which constitutes a pervasive and often inescapable challenge in real-world applications. The reasons behind noise emergence can be numerous, ranging from sensor errors, and bias introduced through data imputation (when dealing with missing data), to simple human error. It can affect both the data as well as the labels. In this work, we do not provide further sub-categorization of noisy samples.

Summary Taken together, these insights support the construction of a geometry-based taxonomy of hard samples presented on the left side of Figure 3.1. Importantly, a data sample can be hard due to multiple reasons, and it is likely that as our understanding of the problem of hardness will expand so will this taxonomy.

In an idealized future, hardness-aware learning could evolve toward a more quantitative formulation—one in which each factor (such as intrinsic dimension, curvature, or local volume) constitutes a variable in a predictive model of sample hardness. With sufficiently precise estimators, such a framework could in principle resemble a regression-like formulation, where each factor is assigned an appropriate scaling coefficient, or even a more nuanced non-linear

model that accounts for interdependencies among factors—for instance, the dependence of the impact of intrinsic dimension on entanglement reported by Kienitz et al. (2022). Such a formulation would enable us not only to diagnose hardness more accurately but also to identify which interventions are most effective for mitigating hardness imbalance. While we are still far from this level of understanding, we believe that this is the direction in which a mature theory of hardness-aware learning should ultimately progress.

3.5 Model-Based Hardness Estimators

3.5.1 Confidence-Based Estimators

Confidence-based hardness estimators are among the earliest and most intuitive approaches proposed by the research community. Their central assumption is simple: easy samples should be classified with high confidence, whereas hard samples tend to result in low confidence predictions. In this context, confidence can be defined in several ways, depending on the available information and problem setting. In the supervised setting, where ground-truth labels are accessible, confidence is often measured as the logit assigned to the true class $\hat{y}$:

$$C = z_{\hat{y}}(x) \quad (3.11)$$

Alternatively, one may use the predicted probability of the correct class $p_{\hat{y}}$, obtained by applying the softmax function to the logits. In the unsupervised or label-agnostic setting, confidence is typically defined as the maximum logit or maximum predicted probability across all classes:

$$C = \max_j z_j(x). \quad (3.12)$$

This formulation allows the use of confidence-based estimators even when true labels are unavailable.

A canonical example of a confidence-based hardness estimator is the **Focal Loss**, first introduced to address class imbalance in object detection tasks (Lin et al. 2017). Focal Loss modifies the standard cross-entropy loss by dynamically scaling its contribution based on the model’s confidence. In the multi-class setting, the standard cross-entropy loss is defined as:

$$CE(p, \hat{y}) = -\log p_{\hat{y}}, \quad (3.13)$$

The Focal Loss introduces a modulating factor $(1 - p_{\hat{y}})^\gamma$, leading to:

$$FL(p, \hat{y}) = -(1 - p_{\hat{y}})^\gamma \log p_{\hat{y}}, \quad (3.14)$$

where $\gamma \in [0, 5]$ is a tunable focusing parameter. This formulation reduces the contribution of easy (i.e., high-confidence) samples to the loss, allowing the model to focus more heavily on harder, low-confidence samples during training. In other words, Focal Loss performs hardness-aware weighting based on model confidence. Focal Loss can be generalized by replacing its confidence-based scaling with alternative hardness measures. For example, Tao et al. (2023) proposed a margin-based variant termed **Dual Focal Loss**, which substitutes the confidence term with margin-based hardness estimates, which we will talk about in the next section.

Limitations of Confidence-Based Estimators

While confidence-based estimators are conceptually simple and computationally efficient, they exhibit inherent limitations stemming from their narrow focus on the logit (or predicted probability) assigned to the ground-truth class. By relying solely on this single value, they discard information about the distribution of scores assigned to competing classes, effectively defining the model’s uncertainty using only one scalar. As a consequence, these estimators may fail to distinguish between predictions that are genuinely confident and those that are only marginally correct. To illustrate this issue, consider two hypothetical six-class predictions:

Sample A: $\mathbf{p} = (0.5, 0.1, 0.1, 0.1, 0.1, 0.1)$,

Sample B: $\mathbf{p} = (0.6, 0.39, 0.005, 0.002, 0.002, 0.001)$.

Both samples assign the highest probability to the true class (class 1), but Sample B — despite having higher p_1 — exhibits much higher ambiguity, as a significant portion of the probability mass is assigned to class 2. Confidence-based estimators would incorrectly classify Sample B as easier, despite the model being clearly more uncertain about its decision. Such misrepresentations arise precisely because the hardness estimate fails to account for competing logits.

3.5.2 Margin-Based Estimators

Margin-based hardness estimators address the limitations of confidence-based approaches by explicitly incorporating information about both the correct class logit and the strongest competing class. Intuitively, the margin captures the gap between the predicted class and its closest competitor—a larger gap indicates a more confident, decisive prediction, while a small or negative gap signals ambiguity or error. Formally, in supervised classification, the margin M is defined as:

$$M(x) = z_{\hat{y}}(x) - \max_{j \neq \hat{y}} z_j(x), \quad (3.15)$$

In unsupervised settings the margin can be approximated using the top two logits:

$$M(x) = \max_j z_j(x) - \max_j^{(2)} z_j(x), \quad (3.16)$$

where $\max^{(2)}$ denotes the second-largest logit value. Returning to the illustrative example from the previous section, applying margin-based estimators to Samples A and B yields more intuitive hardness estimates. Despite Sample B having a higher confidence score $p_{\hat{y}}$, its margin is smaller due to the high probability assigned to the second class. Thus, margin-based estimators correctly identify Sample A as easier than Sample B, as they better capture the underlying uncertainty.

Involving training dynamics

While confidence-based estimators were limited by relying on a single logit, margin-based estimators improve upon this by incorporating additional information from the competing logit. However, both approaches share a common limitation: they only utilize information

available at a single time point, and ignore the evolution of predictions throughout training. Incorporating the training dynamics allows for much more precise and detailed categorization of samples based on hardness.

One method that leverages training dynamics within a margin-based framework is the **Area Under the Margin (AUM)** estimator (Pleiss et al. 2020). AUM extends the margin definition by tracking the margin throughout the entire training process. Specifically, at each epoch t , the margin is computed as:

$$M^{(t)}(x, \hat{y}) = z_{\hat{y}}^{(t)}(x) - \max_{i \neq \hat{y}} z_i^{(t)}(x), \quad (3.17)$$

and then averaged over all T training epochs:

$$\text{AUM}(x, \hat{y}) = \frac{1}{T} \sum_{t=1}^T M^{(t)}(x, \hat{y}), \quad (3.18)$$

Samples with large AUM values are considered easy, as the model was consistently more confident in its prediction throughout training. In contrast, hard samples and mislabeled samples exhibit low or even negative AUM values, reflecting persistent ambiguity or incorrect predictions across epochs.

Hard vs mislabeled samples

Pleiss et al. have demonstrated that mislabeled samples frequently manifest as hard samples under hardness estimators such as AUM. This property allows the majority of hardness estimators to also function as potential tools for label noise detection. Since ground-truth information about label correctness is typically unavailable in real-world datasets, complicating the evaluation of noise detection methods, Pleiss et al. used the concept of *threshold samples*: a subset of intentionally corrupted training samples with known incorrect labels. The model is retrained with these threshold samples included, and AUM values are computed for all threshold samples. A threshold is then selected, typically such that 99% of the threshold samples are classified as mislabeled, which enables identification of potential label noise among the remaining training data.

AUM showcases the value of incorporating training dynamics into hardness estimation. While it remains one of the most widely used margin-based methods, many subsequent estimators have extended the incorporation of learning dynamics by utilizing different training signals such as forgetting events, impact on loss, uncertainty estimates, or gradient behavior.

3.5.3 Uncertainty-Based Estimators

When discussing confidence-based hardness estimators, we pointed out that their main limitation is the reliance on information contained in a single logit. This limitation was partially addressed by margin-based estimators, which incorporate information from two logits, thus offering richer estimates on model confidence and uncertainty. However, margins still do not fully capture model uncertainty. To address this, we now delve deeper into explicit uncertainty quantification, which is grounded in well-established theoretical foundations and has led to the development of multiple hardness estimators.

Uncertainty can be broadly divided into two types: epistemic and aleatoric. Epistemic uncertainty reflects the model’s lack of knowledge; it arises due to limited data, and can in principle be reduced by acquiring more training samples. For instance, poor performance on genuinely hard samples — either because they lie in complex regions or in regions with low sampling density — is primarily driven by epistemic uncertainty. In contrast, aleatoric uncertainty stems from the inherent ambiguity or noise in the data itself and is fundamentally irreducible. Misabeled samples and ambiguous inputs naturally give rise to aleatoric uncertainty.

Formally, the total predictive uncertainty can be decomposed using the law of total variance:

$$\mathbb{V}(Y|x) = \mathbb{V}_\theta(\mathbb{E}[Y|x, \theta]) + \mathbb{E}_\theta(\mathbb{V}[Y|x, \theta]). \quad (3.19)$$

The first term, representing epistemic uncertainty, can be estimated as:

$$\mathbb{V}_\theta(\mathbb{E}[Y|x, \theta]) = \frac{1}{T} \sum_{t=1}^T [\mathcal{P}(x; \theta_e) - \bar{\mathcal{P}}(x)]^2. \quad (3.20)$$

This captures the variation of predictive probabilities for a given data sample across an (in principle) infinite set of models trained on the same data.

The second term corresponds to aleatoric uncertainty and can be estimated as:

$$\mathbb{E}_\theta(\mathbb{V}[Y|x, \theta]) = \frac{1}{T} \sum_{t=1}^T \mathcal{P}(x; \theta_e)(1 - \mathcal{P}(x; \theta_e)), \quad (3.21)$$

which reflects the inherent variance in the data given fixed model parameters. This type of uncertainty is particularly prominent in domains such as medicine, where patients with nearly identical features may nonetheless experience different outcomes - same symptoms but different underlying diseases (Van den Goorbergh et al. 2022).

Estimating hardness through epistemic and aleatoric uncertainty

Swayamdipta et al. (2020) proposed to leverage epistemic uncertainty for hardness estimation in their method called **Data Maps**. Specifically, they use both the mean confidence over epochs $\bar{\mathcal{P}}$ and epistemic uncertainty $\mathbb{V}_\theta(\mathbb{E}[Y|x, \theta])$. In their framework, easy samples exhibit high confidence and low epistemic uncertainty; ambiguous samples have intermediate confidence and high epistemic uncertainty; and hard samples exhibit low confidence and low epistemic uncertainty.

Seedat et al. (2022) points out that a key limitation of estimating hardness via epistemic uncertainty is its inability to capture uncertainty arising from inherent data ambiguity. For instance, if $\mathcal{P}(x; \theta_e) \approx 0.5$ throughout training due to aleatoric uncertainty, the epistemic variance may remain low, incorrectly suggesting the sample is either easy or hard (depending solely on the confidence value), rather than ambiguous. To address this issue, they proposed **Data-IQ**, which substitutes epistemic uncertainty with aleatoric uncertainty $\mathbb{E}_\theta(\mathbb{V}[Y|x, \theta])$ for hardness estimation. This allows the Data Maps framework to be extended into medical and financial domains, where heterogeneity of outcomes is a major issue.

3.5.4 Forgetting-Based Estimators

The notion of forgetting was first introduced by McCloskey & Cohen (1989), who studied sequential learning in connectionist networks.¹ Inspired by the observation that humans learn sequentially (e.g., learning addition precedes multiplication), they argued that *if connectionist networks are to be used to model human learning, then the networks must be able to learn sequentially*. However, such models are affected by **catastrophic interference**, now known as **catastrophic forgetting**, wherein previously acquired knowledge is disrupted by subsequent learning.

Building on this, Toneva et al. (2019) systematically investigated the relationship between forgetting and hardness. They introduced the concept of a **forgetting event**, which occurs when a model transitions from correctly classifying a sample at epoch t to misclassifying it at epoch $t + 1$. Formally, a forgetting event occurs if:

$$\arg \max_j z_j^{(t)}(x) = \hat{y} \quad \text{and} \quad \arg \max_j z_j^{(t+1)}(x) \neq \hat{y}. \quad (3.22)$$

Their analysis revealed that the frequency of forgetting events is not uniformly distributed across samples: some are frequently forgotten throughout training, while others are consistently remembered, with some never being forgotten. Hence, hardness of a data sample would be proportional to the number of its forgetting events—the more often it is forgotten by the model during training the harder it is. Importantly, Toneva et al. observed that forgetting statistics are highly consistent across different model architectures. This finding provides empirical evidence that hardness is partially model agnostic and highlights the need for development of data-based hardness estimators.

Incorporating Forgetting Time into Hardness Estimation

The seminal work of Toneva et al. (2019) has prompted further development of forgetting-based methods. As noted by Maini et al. (2022), subsequent works have focused on: (1) the epoch at which examples are first learned; (2) the number of prediction flips during training; and (3) whether examples are forgotten when held out of training. However, these approaches do not fully capture the heterogeneity of hardness; specifically, they fail to distinguish between examples that are hard due to label noise, low-density regions, or complex decision boundaries. To address this limitation, Maini et al. proposed **Second-Split Forgetting Time (SSFT)**, which analyzes forgetting events during fine-tuning on a holdout dataset. Let $\mathcal{D}_A$ denote the initial training set, and $\mathcal{D}_B$ the holdout set used for fine-tuning. The second-split forgetting time for sample x_i is defined as:

$$SSFT(x) = \min \left\{ t^* \mid \forall t \geq t^* : \arg \max_j z_j^{(t)}(x) \neq \hat{y} \right\}, \quad (3.23)$$

where $z_j^t(x_i)$ denotes the logit after t fine-tuning epochs on $\mathcal{D}_B$, and $(x, \hat{y}) \in \mathcal{D}_A$. Maini et al. found that in this framework, mislabeled samples are typically forgotten rapidly during

¹McCloskey and Cohen define connectionist networks as systems in which *information is stored in weights on connections between simple processing units*—a general definition that extends to modern neural network models.

fine-tuning, rare samples from low-density regions are forgotten more slowly, and complex examples tend to remain stable.²

By focusing on the stability of predictions during fine-tuning, SSFT enables finer-grained categorization of sample hardness into easy, complex, rare, and mislabeled examples. While SSFT offers a more detailed hardness taxonomy, its practical deployment is computationally demanding due to the need for additional fine-tuning, which is particularly expensive for modern large-scale models.

3.5.5 Loss-Based Estimators

The loss function serves as the primary learning signal, guiding parameter updates during training via backpropagation. It is designed to produce low values for samples that the model classifies correctly with high confidence, and high values for misclassified or uncertain samples. Thus, from a hardness perspective, easy samples correspond to low loss, while hard samples incur higher loss values. This naturally allows us to reuse the template introduced in Equation 3.18, simply substituting the margin-based hardness estimate with loss values.

A related formulation was proposed by Arriaga et al. (2023), who define hardness using what they call **action scores**, which are based on cumulative loss:

$$\mathcal{A} = \sum_{t=0}^T L_t \times \Delta t. \quad (3.24)$$

The inclusion of the Δt term is purely to allow drawing connections with continuous-time formulations in information theory, as in experiments they set Δt to correspond to epochs, similar to other works.

Hardness-driven curricula

In the works discussed so far, hardness has primarily been used post-hoc to guide data pruning or noise removal. However, hardness can also be incorporated directly into the training process through curriculum learning, where the sequence or selection of training samples is dynamically adjusted based on their estimated hardness. For detailed reviews on curriculum learning, we refer the reader to the surveys by Wang et al. (2021), Soviany et al. (2022). In this work we will only detail the **RHO-loss** curriculum (Mindermann et al. 2022), which, like other loss-based hardness estimators, prioritizes the selection of samples that are expected to most effectively reduce generalization loss.

The core idea behind RHO-loss is to select training samples that are expected to maximize generalization improvement, rather than merely minimizing training loss, which, according to the authors, is the focus of many other loss-based methods. To formalize this, the authors define a sample selection objective that estimates how much adding a given sample $(x, \hat{y})$ to

²To simulate complex examples, they trained on a union of MNIST (Deng 2012) and CIFAR-10, where CIFAR-10 samples were categorized as complex. To simulate rare examples, they modified CIFAR-100 (Krizhevsky et al. 2009), which contains 100 classes grouped into 20 superclasses (e.g., the superclass *aquatic mammals* contains subclasses such as dolphin, otter, seal, whale, and beaver), into a long-tailed variant. The class imbalance was created by changing the number of examples in each subgroup to $\{500, 250, 125, 64, 32\}$ from the original $\{500, 500, 500, 500, 500\}$.

the training data would reduce the loss on a holdout set:

$$\arg \min_{(x, \hat{y}) \in \mathcal{B}_t} -\log \mathcal{P}_{\hat{\mathbf{y}}^{ho}}(\mathbf{x}^{ho}; \theta^+), \quad (3.25)$$

where θ^+ was obtained by training on $\mathcal{D}_t \cup \{(x, \hat{y})\}$, $\mathcal{D}_t \subseteq \mathcal{D}$ is a subset of training samples seen up to time t , and $\mathcal{B}_{t+1}$ is a new batch of training samples. This expression can be rewritten as:

$$\mathcal{L}(x, \hat{y}; \theta^{\mathcal{D}_t}) - \mathcal{L}(x, \hat{y}; \theta^{\mathcal{D}_{ho} \cup \mathcal{D}_t}), \quad (3.26)$$

which represents the expected reduction in holdout loss obtained by including the sample.

To simplify the computation of the holdout loss, the authors replace the model trained on $\mathcal{D} \cup \{(x, \hat{y})\}$ with one trained only on the holdout set $\mathcal{D}_{ho}$:

$$\mathcal{L}(x, \hat{y}; \theta^{\mathcal{D}_{ho} \cup \mathcal{D}_t}) \approx \mathcal{L}(x, \hat{y}; \theta^{\mathcal{D}_{ho}}) \quad (3.27)$$

This term, referred to as the irreducible holdout loss (IL), represents an estimate of the loss that cannot be reduced by further training. To efficiently estimate this value during training, they introduce a lightweight auxiliary model (the IL model) with lower capacity than the target network. This model is trained on the holdout set alone, leveraging the assumption that sample hardness is at least partially model-agnostic. This formulation naturally leads to a sample taxonomy. Redundant (easy) samples have low training loss and low reducible holdout loss, as their inclusion yields little additional benefit. Noisy and ambiguous (hard) samples exhibit high training loss and high irreducible holdout loss, since their loss is difficult to reduce even with additional data. Outliers (hard) also exhibit low reducible loss, as neighboring samples are unlikely to help reduce their loss.

Potential Pitfalls of RHO-Loss

RHO-Loss selects for training only the hard samples that are non-noisy and task-relevant. While this approach improves test accuracy by reducing test loss, we believe that it raises potential concerns related to Goodhart’s Law: *when a measure becomes a target, it may cease to be a good measure* (Strathern 1997). Specifically, one may question whether excluding outliers and ambiguous samples truly improves generalization, or whether it merely overfits to the test set, which itself contains relatively few such examples. After all, the test set is only a proxy for estimating generalization error, and ambiguous or rare samples inevitably occur in real-world data. Thus, disregarding these harder cases may inadvertently reduce the model’s robustness on out-of-distribution or underrepresented scenarios, even while test accuracy improves.

3.5.6 Gradient-Based Estimators

As discussed earlier, learning is guided by the loss function, with each parameter updated via backpropagation according to the gradient of the loss. Since gradients directly encode how much the model adjusts its parameters for each sample, they provide an equally valuable measure of hardness, complementary to loss. This was utilized by Agarwal et al. (2022) who proposed **Variance of Gradients (VoG)** as a class-conditional hardness estimator,

designed to identify the most challenging samples for human-in-the-loop auditing. They first define S as the gradient matrix of the activation of the output neuron corresponding to the true or predicted class A_p^l with respect to the input pixels of x_i

$$S = \frac{\partial A_p^l}{\partial x_i}. \quad (3.28)$$

They later perform across-channel averaging to collapse S to a 2D spatial map and compute the mean across epochs

$$\mu = \frac{1}{T} \sum_{t=1}^T S_t. \quad (3.29)$$

The variance of gradients across epochs is then computed as:

$$\text{VoG}_p = \sqrt{\frac{1}{T} \sum_{t=1}^T (S_t - \mu)^2} \quad (3.30)$$

Finally, since VoG_p is a spatial map, the scalar VoG score is obtained by averaging across all N pixels

$$\text{VoG} = \frac{1}{N} \sum (\text{VoG}_p) \quad (3.31)$$

A key advantage of VoG is that, unlike many other hardness estimators, it does not require access to ground-truth labels, making it attractive for unsupervised or weakly supervised settings. Furthermore, the authors normalize VoG based on class-level hardness, addressing the non-uniform distribution of hardness across classes. Without this normalization, hard samples from easier classes would rarely be selected for human auditing, as their hardness scores would be low relative to samples from inherently more difficult classes. However, this also raises a broader question: should hardness be normalized across classes at all, or does doing so risk obscuring the very difficulty we seek to understand?

Two stages of learning

Agarwal et al. also point out one important phenomenon. Namely, at early stages of training (setting $T = 3$) the VoG is high for easy samples-the samples on which model has high accuracy after completed training. This switches in late-stage training (computing VoG over the last 3 epochs), where it is the samples on which model performs poorly that have high VoG.

Computationally cheap hardness estimation

Thanks to the incorporation of training dynamics information many of the previously described hardness estimators allow for very detailed and precise characterization of hardness. However, because of that they are also more expensive as they require pretraining a model to obtain the hardness estimates. To address this Paul et al. (2021) propose **Gradient**

Normed (GraNd), which approximates the impact of removing a data sample from the training batch on the training loss via the norm of the gradient.

$$\text{GraNd}(x_i) = \left\| \frac{\partial \mathcal{L}(x_i, \hat{y}_i; \theta)}{\partial \theta} \bigg|_{\theta=\theta^{(t)}} \right\|_2 \quad (3.32)$$

While the gradient norm can be computed for multiple layers, and averaged like it’s done with other hardness estimators using training dynamics, the authors demonstrate that evaluating GraNd scores at model initialization already provides a surprisingly strong approximation of sample hardness. They further observe that GraNd can be closely approximated by the **Error L2 Norm (EL2N)** score, which measures the Euclidean distance between the predicted probability vector and the one-hot encoded true label at an intermediate epoch.

$$\text{EL2N}(x_i) = \left\| \mathcal{P}(x_i; \theta^{(t)}) - \mathbf{e}_{\hat{y}_i} \right\|_2 \quad (3.33)$$

This requires some training, however, the authors show that training on barely 20 epochs can greatly improve the accuracy of EL2N as a hardness estimator, matching the Forget score computed at epoch 200 (experiments conducted on CIFAR-10, CINIC-10 (Darlow et al. 2018), and CIFAR-100). This provides an excellent, computationally cheaper alternative to previously mentioned hardness estimators.

3.5.7 Other Estimators

While many hardness estimators can be generally categorized into confidence-, margin-, uncertainty-, forgetting-, loss-, and gradient-based approaches, there also exist model-based hardness estimators that do not fit neatly into these categories. One such example is the **Noise Detector** introduced by Jia et al. (2023), which employs a separate auxiliary model to identify mislabeled or noisy samples based on training dynamics.

Specifically, they train an LSTM-based label noise detector that learns to distinguish between clean and mislabeled samples by observing training dynamics on synthetic noisy data. The LSTM architecture is a natural choice here, as the training dynamics form a time series — i.e., sequences of model predictions evolving over training epochs. In order to create training data for the detector, the authors inject label noise into existing datasets. The LSTM observes how predictive probabilities evolve across epochs for both clean and corrupted samples, learning patterns that differentiate the two. Importantly, because the detector operates purely on training dynamics rather than model parameters or architectures, it remains model-agnostic and can generalize across architectures and datasets. In their experiments, the detector is trained on CIFAR-100 and later successfully applied to Tiny ImageNet (Le & Yang 2015), CUB-200 (Welinder et al. 2010), Caltech-256 (Griffin et al. 2007), WebVision (Li et al. 2017), and Clothing1M (Xiao et al. 2015). This demonstrates that hardness estimation based on temporal training dynamics can generalize across domains.

3.6 Providing Model-Based Taxonomy of Factors Causing Hardness

Each model-based hardness estimator implicitly assumes certain categorization of hard samples. In Table 3.1 we summarize all of the reviewed hardness estimators, and in Table 3.2 we

Table 3.1: Summary of model-based hardness estimators reviewed in this chapter. Methods are listed in the order in which they were introduced in the section.

Estimator Name	Category	Hardness Estimation Method
Focal Loss	Confidence-based	Confidence in correct label
Dual Focal Loss	Margin-based	Margin between correct and highest incorrect logit
AUM	Margin-based	Average margin between correct and highest incorrect logit during training
Data Maps	Uncertainty-based	Epistemic uncertainty over training
Data-IQ	Uncertainty-based	Aleatoric uncertainty over training
Forgetting Score	Forgetting-based	Count of class flips (correct $\rightarrow$ incorrect) during training
SSFT	Forgetting-based	Monitor forgetting during fine-tuning
Action Scores	Loss-based	Averaged loss over epochs
RHO-Loss	Loss-based	Loss on a holdout set based on smaller auxiliary model
VoG	Gradient-based	Variance of input gradients across epochs
GraNd	Gradient-based	Norm of gradient of loss
EL2N	Gradient-based	L2 distance between predicted and one-hot labels
Noise Detector	Other	LSTM trained on training dynamics

have grouped them based on the taxonomy of the hard samples that each of them imposes on data. This leads to eight distinct sample categories: (1) easy samples; (2) hard samples; (3) mislabeled samples; (4) clean samples; (5) ambiguous samples; (6) complex samples; (7) rare samples; and (8) outliers. Since we are interested in taxonomy of hard samples we disregard easy and clean samples as they represent unproblematic cases. We also disregard hard samples (2) as it’s a supercategory. This leaves five meaningful subtypes, which we formalize using the terminology from Section 3.3, with sample complexity being the minimal number of samples that need to be sampled from a neighborhood $\mathcal{N}(x)$ for the model to correctly learn—be able to collapse it onto the standard basis vector—the given neighborhood with high probability.

- **Rare Samples.** Definition: A data point x is *rare* if its neighborhood $\mathcal{N}(x)$ contains few samples, regardless of its local sample complexity.

Literature:

- Maini et al. (2022) define rare samples as belonging to *underrepresented subpopulations* and identify them via a “second-split forgetting” signal—they are forgotten during the second split only after prolonged training (distinct from quickly-forgotten mislabeled samples or persistent complex ones). They simulate rarity by converting CIFAR-100 into a long-tailed variant.

Table 3.2: Categorization of model-based hardness estimators based on the categorization of hard samples that they assume.

Categorization	Methods
Easy vs. Hard	Focal Loss, Dual Focal Loss, Forgetting, Action scores, VoG, GraNd, EL2N
Easy vs. Hard vs. Mislabeled	AUM
Clean vs. Mislabeled	Detector
Easy vs. Ambiguous vs. Hard	Data Maps, Data-IQ
Easy vs. Complex vs. Rare vs. Mislabeled	SSFT
Easy vs. Outliers vs. Noisy vs. Ambiguous vs Hard	RHO-Loss

- **Complex Samples.** Definition: A data point x is *complex* if the sample complexity $SC(\mathcal{N}(x))$ of its neighborhood is high, even when the neighborhood is well-sampled.

Literature:

- Maini et al. (2022) define complex samples as belonging *to subpopulations for which the classification task is more difficult*. Within their two-split forgetting setup, such samples are never forgotten during the second split training. They simulate complexity by merging MNIST and CIFAR-10, treating CIFAR-10 as the complex component.

- **Out-Of-Distribution Samples.** Definition: A data point $x \in \mathcal{D}$ is out-of-distribution (OOD) if it lies outside of the support of the main sampling distribution $\mathbb{P}_{\mathcal{D}}$.

Literature:

- Mindermann et al. (2022) interpret outliers—effectively OOD samples—as points *far from most of the training data*. In their RHO-Loss formulation, these samples show high training and irreducible loss (IL; Eq. 3.27).

- **Ambiguous Samples.** Definition: A data point x is *ambiguous* if its neighborhood $\mathcal{N}(x)$ lies within a margin ϵ of a different class manifold, or contains samples from multiple class manifolds.

Literature:

- Mindermann et al. (2022) interpret ambiguous samples as points *that have more than one plausible label*. Within their RHO-Loss framework, these samples exhibit high training loss and IL, and are simulated using Ambiguous-MNIST (Mukhoti et al. 2021).
- Swayamdipta et al. (2020) associate ambiguity with high epistemic uncertainty, identifying such samples as those contributing most to out-of-distribution generalization.
- Seedat et al. (2022) define ambiguous samples as those that *appear where the current features are insufficient to distinguish the example correctly, regardless of the model used*. They associate such samples with high aleatoric uncertainty,

as observed in settings where patients share similar features but have different outcomes.

- **Mislabeled Samples.** Definition: A data point x is *mislabeled* if its observed label does not correspond to the class manifold from which it was sampled.

Literature:

- Pleiss et al. (2020) detect mislabeled samples as those with an AUM value below a fixed threshold. As in most works, mislabeled samples are synthetically introduced through random label flipping in ablation studies.
- Jia et al. (2023) train an LSTM-based label-noise detector to recognize the characteristic learning dynamics of mislabeled samples.
- In the two-split forgetting setup of Maini et al. (2022), mislabeled samples are identified as those forgotten earliest during the second phase of training.
- Within the RHO-Loss framework of Mindermann et al. (2022), mislabeled samples correspond to instances exhibiting high training loss and IL.

Hard Samples As shown in Table 3.2, many works treat *hard samples* as a single, generic category, making it difficult to map their findings onto our more granular taxonomy. For instance, Pleiss et al. (2020) define hard samples as those with low AUM values but above the threshold used to identify noisy samples, while Swayamdipta et al. (2020) and Seedat et al. (2022) associate hardness with low confidence or high uncertainty. Such broad definitions conflate multiple underlying factors—such as rareness, complexity, ambiguity, or mislabeling—each of which represents a distinct source of difficulty. This conflation complicates efforts to compare results across studies or to determine which specific hardness factors a given estimator is sensitive to. Addressing this ambiguity is essential for building a unified understanding of sample hardness.

3.7 Unifying taxonomies of hard samples

So far, we have introduced two complementary taxonomies of hard samples. The first, presented in Section 3.4, is grounded in data-based hardness estimators and relies on conceptual reasoning under the manifold assumption. While conceptually appealing, these methods are often difficult to apply in practice due to the curse of dimensionality. The second taxonomy, introduced in the previous section, is based on model-based estimators and synthesizes the categories used across empirical studies, and is therefore largely heuristic, reflecting observed empirical phenomena. By expressing the model-based categories in geometric terms, we can relate them to the data-based taxonomy, bridging empirical observations with plausible underlying mechanisms. This unified taxonomy (see Figure 3.1) allows us to connect observable model behavior to conceptual geometric and statistical causes, setting the stage for a more comprehensive conceptual understanding of hardness.

Ambiguous and complex samples Ambiguous samples are those for which more than one label is plausible. From a geometric viewpoint, ambiguity can be understood as occurring when two class manifolds overlap or approach each other closely in the input space; under

such conditions the posterior distribution would remain uncertain even under ideal sampling. Complex samples are more broadly defined as observations from subpopulations whose local sample complexity is high: they reside in regions of the class manifold that are difficult to collapse into compact latent centroids or to separate from other classes. In geometric terms, local sample complexity can be thought of as increasing with manifold attributes such as extrinsic volume, intrinsic dimension, nontrivial persistent homology, and degree of entanglement with neighboring manifolds, while high local curvature further increases relative difficulty within a class. Thus, ambiguous and complex samples can be understood as sharing a geometric origin: the former primarily reflect manifold closeness/overlap, the latter reflect intrinsic geometric/topological complexity and entanglement between manifolds (see Section 3.4.1). In practice, however, diagnosing these phenomena from finite, suboptimally sampled data is challenging in two distinct ways. For complex samples, it is difficult to determine which geometric factor contributes most to hardness, making it unclear which type of regularization proposed by Ma et al. would be most effective. For ambiguous samples, the difficulty lies in reliably establishing whether class manifolds truly overlap or merely appear close due to limited sampling, which prevents accurate identification of inherently ambiguous cases that might require human oversight or uncertainty-aware models. These diagnostic limitations make it difficult to apply targeted interventions, forcing most current approaches to rely on model-based heuristics or indirect signals of hardness rather than direct, factor-specific corrections.

OOD and rare samples OOD samples are those that lie far from the bulk of the training data. From a sampling perspective, this can be understood as occurring when the main sampling distribution fails to cover the entire class manifold, or when the dataset is constructed from multiple sampling distributions whose supports differ. Consequently, two samples may belong to the same semantic class yet originate from distinct distributions, with the distance between their supports being non-zero. Rare samples, on the other hand, arise from underrepresented subpopulations. Conceptually, this corresponds to non-uniform sampling either between classes—leading to globally rare classes—or within classes—leading to locally rare samples within otherwise well-represented classes. Hence, both OOD and rare samples are difficult not because of intrinsic geometric complexity, but because of inadequate sampling. Nevertheless, these factors may interact: a sample initially considered OOD or rare may, upon denser sampling, reveal underlying geometric complexity, thereby transitioning into the complex category. As with geometry-related hardness, distinguishing between these contributing factors remains a challenge. Detecting within-class imbalance, local rarity, or subtle distributional shifts is difficult under limited sampling and high-dimensionality. Yet, doing so is essential, as these cases determine where data collection, augmentation, or domain adaptation efforts should be targeted to most effectively mitigate hardness.

Mislabeled samples Mislabeled samples correspond to the final supercategory in our data-based taxonomy, representing hardness that arises from annotation errors rather than geometric or sampling-related factors. From a geometric perspective, these are points whose assigned label y does not correspond to the class manifold from which the input x was drawn, thereby violating the assumption that each sample lies on a correctly labeled manifold. While such samples may reside deep within well-sampled and geometrically simple regions,

their incorrect labeling makes them fundamentally misleading during training, introducing contradictory gradients that distort the learned decision boundaries. These samples are particularly harmful because they cannot be corrected through resampling, regularization, or increased data density. Instead, they would require explicit identification and either relabeling or removal. However, distinguishing mislabeled examples from genuinely hard but informative samples—such as ambiguous or complex points—remains a major open challenge.

3.7.1 Identifying research gaps

The unification of model-based and geometry-based taxonomies highlights several key gaps in the current literature—both conceptual and methodological. Below, we identify three central directions in which further research is critically needed.

Intractability of Geometric Estimators in Input Space

While our geometric taxonomy offers a principled understanding of sample hardness, estimating manifold-level properties (e.g., curvature, entanglement, or overlap) in input space remains infeasible due to the curse of dimensionality, as we discussed in Section 3.4.1. Existing data-based estimators fail to approximate these properties reliably when applied directly to the input space, leading to misleading results that diverge from theoretical predictions or intuitive expectations. As a consequence, it is difficult to estimate the relative impact of specific geometric and topological features on sample hardness. This poses a major challenge for targeted regularization: although individual geometric properties can in principle be addressed through specialized loss terms—e.g., for curvature (Ma, Jiao, Liu, Yang, Liu & Li 2023), intrinsic dimension (Ma et al. 2024), or volume (Ma, Jiao, Liu, Li, Yang & Liu 2023)—applying all of them indiscriminately may lead to optimization conflicts, over-regularization, and degraded generalization. Ideally, one would first estimate the contribution of each geometric factor to hardness and then use this information to guide the selection or weighting of appropriate regularization strategies—much like in regression, where knowing the relative influence of each variable enables more effective modeling. There is thus a pressing need for proxy estimators or approximation techniques that can selectively recover meaningful geometric structure despite the curse of dimensionality.

Under-Utilization of Training Dynamics for Hardness Subtype Identification

Training dynamics—such as per-sample loss, confidence, or forgetting patterns—hold rich, under-explored signals about sample hardness. While prior work has leveraged these dynamics to detect mislabeled samples (Jia et al. 2023), and SSFT demonstrated that rare samples can exhibit distinct forgetting behaviors (Maini et al. 2022), little effort has been devoted to linking such dynamic patterns to the specific subtypes in our geometric taxonomy. It remains an open question whether different subtypes produce distinguishable training dynamics; for example, samples whose hardness stems primarily from the intrinsic dimension of their class manifold might show different learning dynamics than those affected mainly by local curvature or neighborhood entanglement. Such distinctions might offer a promising alternative to data-based geometric estimators by potentially enabling the estimation of each geometric factor’s contribution to sample hardness through training dynamics alone.

Conflation of Hardness Subtypes in Model-Based Estimators

Each model-based hardness estimator imposes its own categorization of hard samples. As summarized in Table 3.2, some estimators—like SSFT or RHO-Loss—provide relatively fine-grained distinctions among subtypes, but many conflate several sources of hardness into broader categories. This conflation leads to a broader issue in the literature: downstream methods are frequently evaluated on the aggregated set of “hard” samples without performing ablation studies to assess how each subtype is affected. A similar concern was recently raised by Seedat et al. (2024), who show that different estimators target different notions of hardness, making it difficult to compare or generalize findings. As a result, we risk drawing misleading conclusions—some interventions may benefit one subtype while harming others, yet this remains invisible under coarse evaluation. Worse still, these methods are often applied uniformly across datasets and classes, despite each having distinct hardness profiles shaped by geometry, sampling, topology, or label noise. This undermines the interpretability and reliability of hardness-aware learning. In principle, interpretable learning would begin with estimating the hardness profile of a dataset—using either data-based estimators or training dynamics—and then tailoring the intervention to the dominant factors. Fully resolving this issue will require a systematic effort to disentangle and evaluate the impact of methods on each subtype individually—an effort far beyond the scope of any single study, yet essential for future progress and for achieving principled, interpretable hardness-aware learning.

3.8 Conclusion

In this chapter, we investigated data-based hardness estimators through the lens of the geometric perspective on learning. This analysis revealed three fundamental sources of sample difficulty: geometry-related, sampling-related, and error-related issues. These informed the construction of a novel, geometry-based taxonomy of hard samples.

We then turned to model-based estimators, examining their core assumptions and limitations. While these estimators are highly practical, their reliance on behavioral heuristics offers limited insight into the underlying causes of hardness. From this analysis, we distilled the distinct categorizations each estimator imposes on hard samples and unified them into a single model-based taxonomy of hard samples.

To understand how these two perspectives relate, we mapped each category from the model-based taxonomy onto components of the geometry-based one. This revealed that many model-based estimators act as coarse proxies for multiple subtypes of hardness, often conflating distinct sources of difficulty. As a result, current practice makes it difficult to identify the true reasons behind sample hardness.

We conclude by identifying three core directions for future research: (1) overcoming the intractability of geometric estimators in input space; (2) leveraging training dynamics to more precisely characterize hardness subtypes; and (3) avoiding the conflation of subtypes in model-based estimators. We posit that until hardness subtypes are explicitly accounted for in both evaluation and design, our understanding of what makes samples hard — and what makes methods effective — will remain fundamentally incomplete.

Chapter 4

Which Hardness Estimator to Use? Verifying Estimate Robustness and Benchmark of Data- and Model-Based Estimators

4.1 Introduction

In the previous chapter, we reviewed hardness estimators drawn from both model- and data-based families, synthesizing them into a unified taxonomy of hard samples and identifying future research directions. However, we deliberately left open one crucial question—*which hardness estimator should one use in practice?* This question is particularly challenging due to the lack of robust benchmarking procedures, and is the central focus of this chapter.

Benchmarking plays a central role in methodological research. Once a new method is proposed, it is not sufficient to evaluate it based on theoretical guarantees or intuitive appeal; robust empirical validation is essential to assess its practical applicability. This is especially important in machine learning, where many widely used methods are guided by heuristic insight rather than formal derivation, and even theoretically grounded approaches often rely on assumptions that fail in practice (Dundar et al. 2007, He et al. 2021). These discrepancies make empirical benchmarking not just complementary to theory, but indispensable.

However, evaluating hardness estimators presents a unique challenge: the property being estimated is not directly observable. In many settings, empirical validation is straightforward because the target quantity—such as a class label or regression target—is explicitly defined and provided. In contrast, there is no ground truth for how “hard” a sample truly is, nor a definitive list of factors contributing to hardness, as we highlighted in the previous chapter. As a result, it is difficult to determine which estimator performs better, making traditional benchmarking approaches inapplicable. To work around this, prior research has adopted one of three main evaluation strategies:

- **Downstream-task-based evaluation**, which assesses whether a hardness estimator improves performance on downstream tasks like data pruning, curriculum learning, or label noise detection.

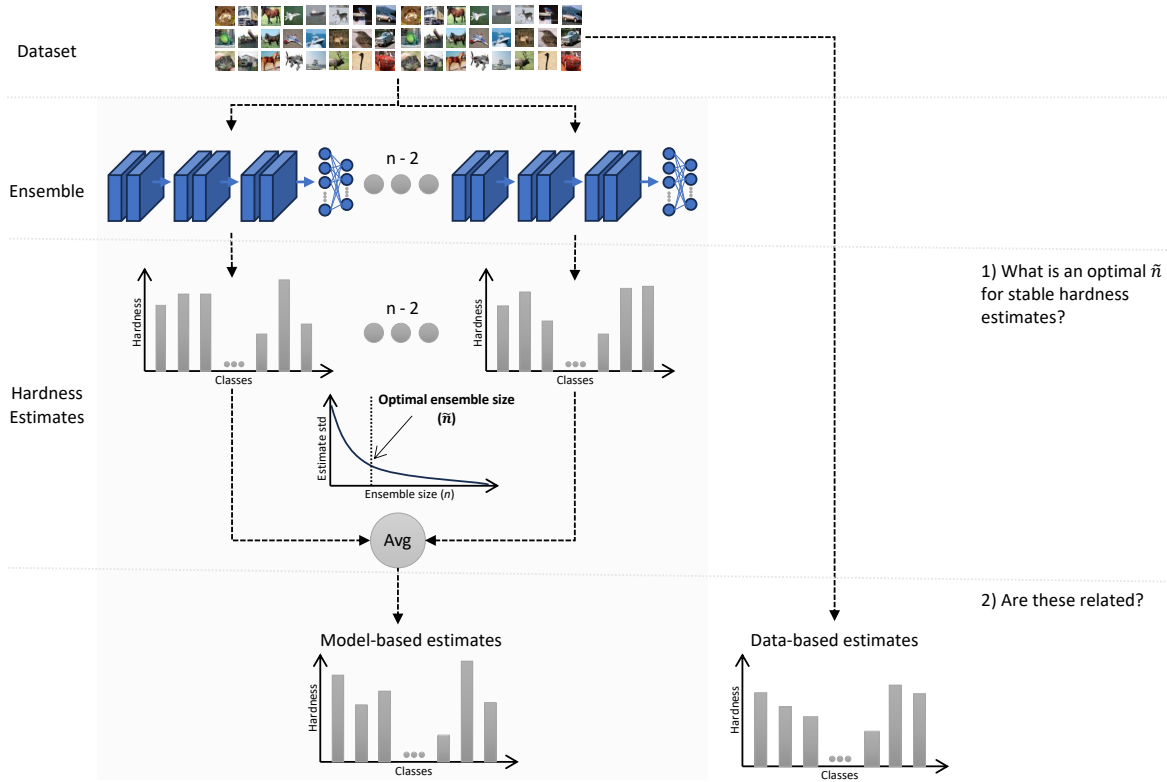


Figure 4.1: Our literature review revealed that many works use hardness estimates obtained over a single run. This raises concerns regarding stability and reliability of such hardness estimates. To address this we investigate the impact of the ensemble size on the hardness estimates, revealing that for some downstream tasks single-run-based hardness estimates are extremely unreliable. After establishing that we perform the first comparative benchmarking between model- and data-based hardness estimators, finding that current data-based estimators are not able to achieve good performance and identify curse of dimensionality as the main cause.

- **Injection-based evaluation**, which introduces synthetically hard samples—e.g., mis-labeled, out-of-distribution, or augmented examples—and evaluates whether the estimator can identify them.
- **Proxy-based evaluation**, which compares hardness predictions to an external variable assumed to correlate with hardness, such as accuracy.

Unfortunately, there is no standardization in how these strategies are used. As a result, papers proposing different hardness estimators often rely on different evaluation protocols, making it difficult to compare them directly or identify the state of the art. The work of Seedat et al. (2024) is, to our knowledge, the only one to perform a thorough benchmarking study of model-based hardness estimators under a fixed strategy (injection-based), highlighting the broader need for more systematic evaluation in this field. Beyond the issue of standardization in evaluation protocols, we identify two additional shortcomings in the current literature on hardness estimators that we address in this chapter (see Figure 4.1 for visualization).

Benchmarking Model-Based Estimators Against Data-Based Ones

First overlooked issue concerns the lack of comparative evaluation between model-based and data-based hardness estimators. While both aim to quantify sample difficulty, they have not yet been benchmarked side by side as to our knowledge. This gap obscures important practical tradeoffs between interpretability, stability, and performance.

We address this by conducting a proxy-based evaluation of selected data- and model-based estimators. Our results, discussed in Sec. 4.5.3, show that intrinsic dimension (ID), volume, curvature, and class impurity exhibit weak or negligible correlations with class-level accuracy, suggesting poor estimation performance. We attribute this largely to the curse of dimensionality, as argued in Chapter 3. Supporting this interpretation, we observe that ID and volume, which are tightly linked in theory, are in fact poorly correlated when computed in the raw space. We therefore apply these estimators to PCA-reduced representations, expecting lower intrinsic dimensionality to improve estimate reliability, yet finding no consistent change. Addressing this likely requires more sophisticated dimensionality reduction techniques—a promising direction for future work—but in this thesis we treat PCA as a simple, low-cost probe.

Robustness Analysis of Hardness Estimates

A second critical shortcoming is the lack of robustness analysis in hardness estimation. Most works compute hardness using a single model, without reporting variance or consistency across seeds. While some studies justify this choice using correlation metrics such as Spearman or Pearson across repeated runs, we find this practice insufficient. High correlation does not necessarily imply stability in the top- k hardest or easiest samples — a requirement that is critical for computing resampling ratios in our hardness-based resampling method, as well as for other downstream tasks such as noise removal and curriculum learning.

To rigorously assess robustness, we introduce three complementary metrics. First, the pruning change rate (PCR) quantifies the fraction of samples differing in the top- k hardest or easiest sets as ensemble size grows, indicating stability in extreme sample rankings. Second, we compute the Jaccard overlap between top- k sets generated by different hardness estimators to measure agreement and the risk of suboptimal estimator choice. Third, we analyze absolute differences in class-wise sample counts under hardness-based resampling as ensemble size increases, revealing the impact of estimator variability on our task of class-level resampling ratios.

Our experiments, discussed in Sec. 4.5.1 and 4.5.2, demonstrate that hardness estimates from a single model can be very unstable depending on the task. Namely, while single-run estimates appear stable enough for certain hardness estimators when identifying top- t most extreme samples for high t (50%), the same is not the case for lower t (10%). Furthermore, we find that hardness estimators that do not utilize information on learning dynamics are less stable than their counterparts that use that information. We also notice low agreement across hardness estimators when identifying a set of top- t most extreme samples. Combined, this tells us that the choice of hardness estimator is very important, due to low agreement, and that single-run estimates are often unstable, despite high correlation values reported in the literature. Furthermore, our findings show the issues of estimating hardness using a single model run, highlighting the need to use ensemble approaches where hardness is

estimated over multiple runs. While further research is needed to quantify how this instability affects downstream outcomes, in Chapter 4 we rely on this robustness analysis to inform our experimental design and henceforth use hardness estimates averaged over ensembles of 10 models to improve stability.

Contributions

To conclude, this chapter makes the following contributions:

- We perform the first direct benchmark comparing model-based and data-based hardness estimators, exposing large gap between the two.
- We find that applying PCA does not reliably enhance data-based estimators, indicating the need for more sophisticated dimensionality reduction techniques to make data-based estimators competitive.
- We highlight the volatility of hardness measures from single networks, recommending ensemble approaches for robust downstream use.

4.2 Reviewing Benchmarking Strategies of Hardness Estimators

4.2.1 Benchmarking in Works Involving Data-Based Hardness Estimators

Most works involving data-based hardness estimators do not attempt to benchmark them in a comparative or systematic manner. In many cases, this is entirely appropriate: several studies focus on analyzing (1) the evolution of data properties during training or inference (Naitzat et al. 2020, Magai 2023), or (2) the relationship between such properties and sample complexity (Kienitz et al. 2022), neither of which naturally lends itself to benchmarking.

However, other works do attempt to evaluate their proposed estimators, albeit in fragmented or indirect ways. Broadly, we can identify two recurring strategies. First, several studies introduce geometry-inspired regularization strategies and assess their utility via downstream-task-based evaluation - comparing class-level performance gaps with and without regularization (Ma, Jiao, Liu, Yang, Liu & Li 2023, Ma, Jiao, Liu, Li, Yang & Liu 2023, Ma et al. 2024). While this reveals the potential benefits of geometric constraints, these works do not compare their regularizers against each other, making it difficult to determine which geometric property most effectively mitigates class-level hardness. Second, some works use proxy-based evaluation, exploring the relationship between data-centric metrics and classification accuracy (Ansuini et al. 2019, Pope et al. 2021, Kaufman & Azencot 2023). While this is rarely framed in terms of hardness explicitly, the core idea is to demonstrate that a given data property tracks or predicts class-level performance. In effect, these studies treat accuracy as a proxy ground truth for hardness and use correlation with accuracy as an implicit benchmark.

4.2.2 Benchmarking in Works Involving Model-Based Hardness Estimators

Model-based hardness estimators are typically introduced alongside a specific downstream task, with their evaluation tailored to that particular application. In this section, we review how such estimators have been benchmarked in prior work, organized around three primary use cases: detection of problematic samples, data pruning, and training modulation.

Detection of Problematic Samples

Hardness estimators are frequently used to identify and, potentially, remove problematic samples. The most prominent use case is label noise detection, where mislabeled samples in the training data are removed. Several model-based estimators, such as AUM (Pleiss et al. 2020), SSFT (Maini et al. 2022), noise detector (Jia et al. 2023), and action scores (Arriaga et al. 2023), have been evaluated on this downstream task. Another related task is OOD detection, with the goal of identifying inputs that deviate significantly from the distribution of the training data. Among the reviewed estimators, only VoG (Agarwal et al. 2022) has been applied to this task, although other hardness estimators, such as confidence-based ones, are commonly used across the literature (Yang et al. 2024).

Benchmarking for these tasks typically follows one of two experimental protocols. The first evaluates how accurately the estimator identifies problematic samples that were synthetically introduced into the dataset, using metrics such as AUC, AUROC, or average precision (injection-based evaluation). The second evaluates the quality of the estimator indirectly by removing samples with high estimated hardness from the synthetically corrupted training set and measuring the resulting change in accuracy (downstream-task-based evaluation). In this setup, better noise detection corresponds to smaller accuracy degradation—or even accuracy improvement—after data cleaning. Hence, the main objective of benchmarking is to estimate how accurate hardness estimators are at identifying the problematic samples.

Data Pruning

Data pruning refers to the process of removing training samples based on their estimated utility, with the goal of improving generalization or reducing computational cost. Typically, benchmarking involves pruning increasing fractions of the training set according to the hardness estimator, retraining models on the remaining data, and measuring test accuracy. Performance is then compared against baseline methods, including training on the full dataset and pruning based on alternative heuristics, including random pruning, with results averaged across multiple seeds (downstream-task-based evaluation).

Two main variants of hardness-based pruning exist in the literature, each associated with distinct objectives. The first aims to reduce computational burden by removing samples that contribute least to learning—those deemed easy or redundant. Estimators such as Forgetting, EL2 and GraNd (Toneva et al. 2019, Paul et al. 2021) were evaluated in this setting. The second variant seeks to improve robustness and generalization by pruning both easy and hard samples, i.e., those that may be either uninformative or noisy. This strategy has been explored in works like Data Maps (Swayamdipta et al. 2020), Data-IQ (Seedat et al. 2022), and SSFT (Maini et al. 2022). Hence, the main objective of benchmarking is to indirectly

estimate how accurate hardness estimators are at identifying the least informative and, in some cases, the most harmful samples.

Hardness-Aware Training Modulation

Beyond post hoc interventions, sample hardness estimators have also been used to dynamically guide the training process itself. While this has historically taken different forms across loosely connected subfields, the unifying idea is to modulate the training dynamics in response to the estimated informativeness or difficulty of individual samples.

A central example is curriculum learning, in which the training process follows a curriculum—an ordering of training samples from easiest to hardest—under the hypothesis that learning simpler examples first improves convergence and generalization. This curriculum can be predefined based on heuristics (Bengio et al. 2009), or constructed dynamically by the model itself in what is known as self-paced learning (Kumar et al. 2010). From our reviewed methods, only Action Scores (Arriaga et al. 2023) has been applied to this task, however, confidence-based methods are also commonly employed (Wang et al. 2021, Soviany et al. 2022).

Other methods similarly adjust training emphasis based on instance-level hardness. For example, Focal Loss (Lin et al. 2017) and its extensions (Tao et al. 2023) introduce scaling factors that downweight high-confidence or high-margin samples. Hence, rather than explicitly ordering samples, these approaches implicitly prioritize harder ones by shaping the objective function. Rather than adjusting how much a sample matters, one can instead decide whether it should remain part of the learning process at all. This intuition underlies instance-dependent early stopping (Yuan et al. 2025), where samples are not used in the current epoch if their second-order loss falls below a certain threshold. A related idea drives RHO-Loss (Mindermann et al. 2022), which curates batches by estimating which samples are expected to yield the largest reduction in hold-out loss.

Because all of these methods—whether based on ordering, weighting, or removal—aim to modulate the training process in response to instance-level signals, their benchmarking often follows a downstream-task-based philosophy: does the method improve performance, efficiency, or fairness compared to standard training or competing techniques? Although specific metrics vary—ranging from accuracy and F1 score to convergence speed—the focus on the whole spectrum of hardness remains relatively consistent. In other words, the main objective of benchmarking is to indirectly estimate how accurate hardness estimators are at ranking samples based on hardness.

4.2.3 Benchmark papers

A critical contribution to the study of sample hardness comes from the recent work of Seedat et al. (2024), who introduced the first systematic benchmarking framework for hardness estimators, called H-CAT. They propose a series of three experiments. The first assesses how well hardness estimators identify hard samples when datasets are corrupted with increasing proportions of synthetically injected instances from five predefined hardness subcategories involving mislabeled, OOD, and augmented samples. They evaluate each estimator’s performance using the Area Under the Precision–Recall Curve (AUPRC), treating these injected samples as positive cases. The second experiment ranks the estimators based on their AUPRC

Table 4.1: *Overview of works by category. Model-based and data-based hardness studies differ in their benchmarking strategies, with the latter relying primarily on proxy-based evaluation.*

Downstream-task-based	Injection-based	Proxy-based
Lin et al. (2017), Toneva et al. (2019), Pleiss et al. (2020), Swayamdipta et al. (2020), Paul et al. (2021), Maini et al. (2022), Mindermann et al. (2022), Seedat et al. (2022), Tao et al. (2023), Jia et al. (2023), Arriaga et al. (2023), Yuan et al. (2025)	Pleiss et al. (2020), Maini et al. (2022), Agarwal et al. (2022), Mindermann et al. (2022), Jia et al. (2023), Seedat et al. (2024)	Lorena et al. (2018), Ansuini et al. (2019), Pope et al. (2021), Ma, Jiao, Liu, Yang, Liu & Li (2023), Ma, Jiao, Liu, Li, Yang & Liu (2023), Kaufman & Azenkot (2023), Ma et al. (2024)

scores from the first task and visualizes these rankings using critical difference diagrams to assess statistical significance. The third experiment shifts focus to ranking consistency by computing the average Spearman correlation between hardness scores obtained across three independently trained model instances. As far as we are aware, this remains the first and only benchmarking paper focused on model-based estimators.

Another important work is that of Lorena et al. (2018), which is, to our knowledge, the only systematic benchmarking study of data-based hardness estimators, although their taxonomy of measures departs substantially from ours. Their experiments yield two main contributions. First, they demonstrate that data-based hardness estimates can be used not only to assess dataset complexity but also to guide model design, for instance by recommending Support Vector Regression hyperparameters that control model capacity. In the context of the broader vision outlined in the previous chapter, this represents a concrete step toward hardness-aware learning—an approach where data-derived estimators reveal the underlying sources of complexity and inform appropriate interventions. In this case, the intervention concerns architectural choice rather than sample-level regularization mentioned in the previous chapter, highlighting how hardness estimates can improve interpretability by explaining why specific hyperparameters are chosen instead of relying on blind grid search. Second, they evaluate the relationship between hardness estimates and learner performance by measuring the Pearson correlation between these estimates and the normalized mean squared error (NMSE) of base learners across 39 regression datasets. Their correlations are generally stronger than those we observe, but this difference is likely due to two factors: the datasets in their study are substantially lower-dimensional than CIFAR-10/CIFAR-100 (reducing the impact of the curse of dimensionality), and their estimators are comparatively simple (e.g., feature–target correlation, mean absolute error of a linear regressor, average examples per feature dimension) compared to the estimators we use. Importantly, neither benchmark compares data-based estimators against model-based ones; to date, the two families have only been evaluated in isolation.

4.2.4 Summary

As became apparent through our review and as summarized in Table 4.1, the choice of benchmarking strategy depends strongly on the intended purpose of benchmarking. Downstream-task-based benchmarking is the most direct and interpretable approach: it simply evaluates how a hardness-aware method affects performance on the downstream task. This strategy

does not require an explicit definition of hardness and can accommodate analyses across the entire hardness spectrum or within selected subsets (e.g., the hardest or easiest samples). Its simplicity, however, comes at a cost—it requires retraining or re-evaluating the downstream model for each method under comparison and does not provide detailed insights into performance on distinct hardness subtypes.

Injection-based benchmarking is conceptually more demanding. By design, it focuses on a specific subtype of hard samples (such as mislabeled or out-of-distribution examples) that must be explicitly defined and artificially injected into the dataset. When applicable, it provides highly informative results, as it directly tests whether an estimator correctly identifies a known type of hard sample. Importantly, injection- and downstream-task-based benchmarking are often used in tandem: since the downstream-task-based protocol is conceptually simple and provides complementary information, many works that perform injection-based benchmarking also repeat the downstream task to quantify its impact. The drawback is the additional computational cost of retraining models. Conversely, the reverse rarely occurs—most works relying on downstream-task-based benchmarking do not adopt injection-based evaluation, either because it is conceptually more demanding (requiring a strict definition of the hardness subtype of interest) or because they aim to study hardness across the full spectrum, where injection-based protocols are not applicable.

Proxy-based benchmarking is the most common strategy among data-based estimators. It is inexpensive and easy to implement, as it evaluates estimators using a proxy metric that approximates ground-truth hardness (for instance, class accuracy or loss). While convenient, it relies heavily on the assumption that the chosen proxy meaningfully reflects true hardness and—like downstream-task-based benchmarking—does not provide detailed information about performance on different hardness subtypes. Each strategy therefore offers a distinct balance between practicality, specificity, and generality.

In our benchmarking experiments—where we compare data- and model-based estimators—we adopt a proxy-based evaluation following common practice in the literature. Specifically, we treat class-level accuracy as a surrogate for ground-truth hardness and assess estimator quality via the Pearson and Spearman correlations between estimated class hardness and accuracy values. Higher correlations indicate better alignment with this proxy notion of hardness. We acknowledge that this design inherently favours model-based estimators, since accuracy reflects model-dependent biases. Nonetheless, in the absence of ground-truth hardness, all existing evaluation paradigms suffer from comparable limitations: proxy-based and downstream-task-based evaluations both hinge on model behaviour, while injection-based approaches only test narrow subsets of hard samples. Because our goal—hardness-based resampling—requires evaluating estimators across the full hardness spectrum rather than isolated subtypes, proxy-based benchmarking remains the most practical and conceptually aligned choice for our setting.

4.3 Background on Stability of Hardness Estimates

No matter the evaluation strategy or the downstream task involved, the majority of works on hardness estimation lack a systematic robustness analysis. From the reviewed methods, the authors of Focal Loss (Lin et al. 2017), Dual Focal Loss (Tao et al. 2023), RHO-Loss (Mindermann et al. 2022), VoG (Agarwal et al. 2022), and Noise Detector (Jia et al. 2023)

compute hardness from a single model and report no robustness analysis. Other works, such as AUM (Pleiss et al. 2020) and Data-IQ (Swayamdipta et al. 2020), also use single-seed estimates but measure correlations across architectures, with both reporting relatively stable estimates (98% for AUM and 80–98% for Data-IQ depending on the dataset). Pleiss et al. further find that omitting training dynamics leads to noticeably less stable hardness estimates.

This is not to say that robustness has been entirely neglected. The authors of GraNd and EL2N (Paul et al. 2021) use ten seeds, although they do not explicitly report robustness metrics. Data Maps (Seedat et al. 2022) and Forgetting (Toneva et al. 2019) go further, estimating hardness over five and one hundred seeds respectively, and explicitly measuring Pearson correlation between runs. Seedat et al. find that the average correlation between five random seeds is 0.75 or higher, while Toneva et al. report 89.2% correlation on average between pairs of runs and 95% when aggregating over sets of five—clear evidence that averaging across multiple runs improves stability. Both works also note that correlations plateau early in training, implying that training to full convergence is unnecessary, as model-based hardness estimates stabilize after a certain point. This explains how EL2N, which has only access to a probe network obtained after 20 epochs (10% overall training time), is competitive with hardness estimators that have access to the full range of learning dynamics. Finally, the authors of SSFT (Maini et al. 2022) (five seeds) and Action Scores (Arriaga et al. 2023) (two seeds) perform robustness analyses both across seeds and across architectures, consistently finding greater variability across architectures than within seeds.

Summary

Despite their growing use, model-based hardness estimators are almost exclusively benchmarked within the context of their target downstream task. This makes cross-comparison between estimators—and especially with data-based alternatives—very limited in existing literature. Moreover, due to task-specific design choices, many estimators are evaluated only on their ability to identify either easy or hard samples, rarely both. For example, benchmarks for label noise detection focus narrowly on the hardest examples, while data pruning often targets the easiest. As a result, current evaluation practices provide only a fragmented view of estimator utility. Most importantly, there is little attention being put to the robustness of hardness estimates, with many works using estimates produced from only a single run.

4.4 Methodology

This section details the design and implementation of our benchmarking experiments. We begin with class-level analyses, where model- and data-based hardness estimators are compared against accuracy-based ground-truth. We then shift to instance-level experiments, investigating the stability of model-based estimators as a function of ensemble size to identify a suitable ensemble size for stable downstream performance of hardness-related tasks. Finally, we describe the training setup and implementation-specific considerations related to curvature estimation.

4.4.1 Comparing Data- and Model-Based Hardness Estimators

We begin by training a modified ResNet-18 architecture on the dataset of interest using an ensemble of twenty independently initialized models. During training, we track several model-based instance-level hardness estimators: confidence, AUM, aleatoric uncertainty (denoted as Data-IQ), loss, and the number of forgetting events. Since Data-IQ incorporates both confidence and aleatoric uncertainty, we exclude the confidence term to avoid redundancy. All estimators are averaged across both training epochs and ensemble members to yield a single scalar per data sample, reducing the impact of model biases on hardness estimates. After training, we recompute all estimators—excluding Forgetting—using only information from the final epoch. These variants, denoted iConfidence, iAUM, iData-IQ, and iLoss (with the prefix “i” indicating “instant”), allow us to assess the importance of learning dynamics in hardness estimation. In parallel, we compute a suite of data-based estimators including intrinsic dimension (ID), volume, Gaussian curvature, and class impurity. For ID, we use TwoNN due to its lack of hyperparameters (Figure 4.2 shows these ID estimates on CIFAR-10 and CIFAR-100), and for curvature, we apply CAML.

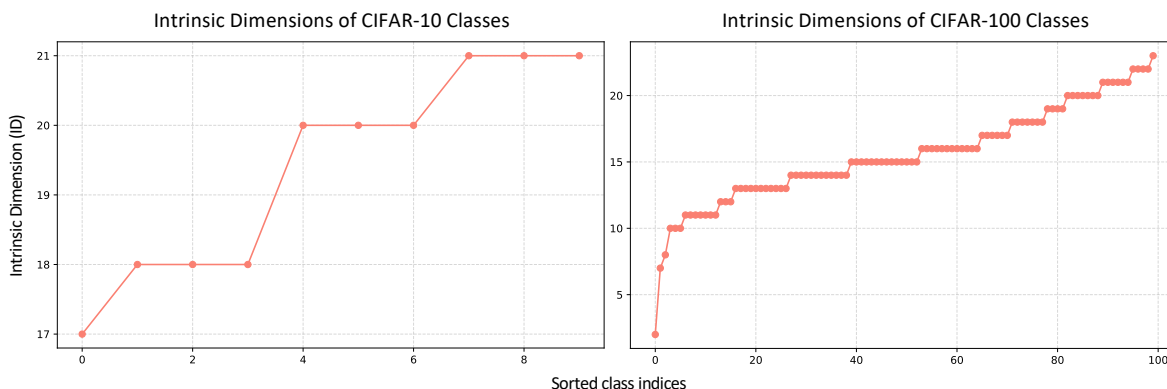


Figure 4.2: Estimated class-level intrinsic dimensions (IDs) using TwoNN. We provide this Figure to visualize the difference between the extrinsic dimensions—3072—and IDs for classes in CIFAR-10 and CIFAR-100. It is clear that the class manifolds are significantly lower dimensional than the space that they occupy.

We then aggregate instance-level estimates to the class level by averaging over all training examples per class. We also compute two complementary class-level accuracy metrics. The first, which we term *Subjective Accuracy*, is derived from the ensemble used for hardness estimation. The second, *Objective Accuracy*, is computed using a diverse set of pretrained classifiers—including ResNet (He et al. 2016), VGG (Simonyan & Zisserman 2015), MobileNetV2 (Sandler et al. 2018), ShuffleNetV2 (Ma et al. 2018), and RepVGG (Ding et al. 2021)—sourced from Chen (n.d.) and yielding a total of twenty networks, with four variants of increasing complexity per architecture (e.g., ResNet20, ResNet32, ResNet44, and ResNet56). Because we have no access to the true sample hardness, we follow standard practice and treat accuracy as a proxy ground truth. From this perspective, estimators that better correlate with accuracy are assumed to better capture true class-level difficulty. Finally, we quantify the alignment between the obtained class-level estimates by computing pairwise Pearson and Spearman correlations across all class-level measures.

As discussed in Chapter 3, data-based estimators may yield poor estimates due to the curse of dimensionality when applied directly in input space. To investigate whether this issue can be mitigated, we project each class manifold onto lower-dimensional subspaces using class-conditional PCA—i.e., performing PCA separately for each class to better preserve local geometric structure. We then recompute the data-based hardness estimators on these projected manifolds and evaluate how their correlations with both *Subjective Accuracy* and *Objective Accuracy* vary as a function of the projection dimensionality. For Curvature, we follow a slightly different procedure: because CAML internally performs PCA to define local tangent spaces, we bypass external PCA and instead vary the internal projection dimensionality within CAML itself, applying it directly to the raw input data. This ensures consistency with the estimator’s design while still enabling a controlled study of dimensionality effects. Overall, this experimental setup enables us to comprehensively benchmark hardness estimators and examine whether PCA improves the reliability of data-based measures.

To summarize, all of these experiments aim at answering the following research questions:

- Can data-based hardness estimators compete with model-based estimators in predicting class difficulty, using class-level accuracy as a ground-truth proxy?
- Can projecting data into class-conditionally PCA-transformed spaces mitigate issues of unreliability due to the curse of dimensionality in data-based estimators?

4.4.2 Instance-Level Stability and Agreement of Model-Based Hardness Estimators

While the previous section focused on class-level benchmarking, we now shift to evaluating hardness estimators at the instance level. We reuse the instance-level model-based hardness estimates obtained in earlier experiments. Since data-based estimators are model agnostic, and our objective is to measure the stability of hardness estimates with respect to the number of seed-measurements, we omit them from this experiment. To assess the behavior of each estimator, we consider a pruning-based setup in which samples are ranked by hardness and progressively removed. For each estimator, we identify the subset of samples that would be removed at pruning thresholds of 10%, 20%, 30%, 40%, and 50%. This is done under two scenarios:

1. **Easy-sample pruning**, where we remove the *lowest*-scoring (i.e., easiest) samples;
2. **Hard-sample pruning**, where we remove the *highest*-scoring (i.e., hardest) samples.

This allows us to measure the stability of identifying the most extreme samples by different hardness estimators. Specifically, we vary the number of models M used to compute ensemble-averaged hardness scores, ranging from $M = 1$ to $M = 10$. For each M , we compute the set of pruned indices $S_A^{(M)}$ at a given pruning threshold, with hardness estimator A guiding the choice of pruned indices. To ensure robustness of our results we repeat the computation five times each time randomly choosing the models for the ensemble from the twenty models we trained for class-level benchmarking. To quantify how the pruning sets change as we increase ensemble size we introduce *Pruning Change Rate (PCR)*, which measures the percentage of samples whose inclusion in the pruned set changes:

$$\text{Pruning Change Rate}(A, M) = \frac{|S_A^{(M)} \setminus S_A^{(M+1)}|}{|S_A^{(M)}|} \times 100\% \quad (4.1)$$

Since the pruning threshold is fixed—both sets have the same cardinality and no duplicate samples—this difference cleanly captures the percentage of samples that would be replaced if the ensemble was expanded from M to $M + 1$ models. The earlier-mentioned five-fold repetition of computation allows us to obtain five PCRs which we later use to compute mean and standard deviation. This experiment closely parallels the third benchmark proposed by Seedat et al. (2024). However, our approach differs in two key ways. First, we use a more interpretable metric—PCR—which directly quantifies changes in pruned subsets rather than relying on correlation of full sample orderings. Second, we explicitly distinguish between stability in identifying easy and hard samples, while their analysis focuses on global ranking stability. We believe these design choices yield a complementary perspective and contribute a distinct, application-oriented benchmark. Moreover, by distinguishing between performance on easy and hard sample identification we address the methodological gap identified in Section 4.2.2, where prior evaluations often focused on only one side of hardness spectrum.

Ideally, assessing the impact of ensemble size on downstream tasks would involve repeating the entire pipeline—recomputing hardness scores for each ensemble size, pruning accordingly, and retraining models on the resulting datasets—until performance converges. However, this exhaustive approach is computationally prohibitive. We hope that the introduced PCR will act as a more computationally feasible surrogate metric of stability on pruning-based downstream tasks. The key intuition is that if PCR remains high, then adding models to the ensemble will significantly change which samples are identified as most extreme, likely resulting in differences in downstream task performance. Conversely, once PCR stabilizes, we expect downstream behavior to stabilize as well. While this proxy is not a formal guarantee, we argue it provides a practical and interpretable estimate of estimator robustness to the downstream tasks with respect to ensemble size.

The main objective of this thesis is to verify practical applicability of hardness-based resampling as a method to reduce performance gap across classes on balanced datasets. As such, we are also interested in the stability of the resampling ratios—more specifically the per-class sample counts after resampling—with respect to the size of the ensemble. To estimate that, we measure the averaged changes to the per-class sample count after resampling after increasing the ensemble size used to estimate hardness by one. More formally, we define

$$\text{Absolute Difference}_c^{(M)} = |b_c^{(M+1)} - b_c^{(M)}|, \quad (4.2)$$

where $b_c^{(M)}$ is the number of samples in class c after resampling with the hardness estimates being obtained using ensemble of M models, with the algorithm for computing $b_c^{(j)}$ being detailed in the next chapter. For clarity we report class-averaged Absolute Difference. Similarly to PCR computation, here we also repeat experiments five times, each time using a new ensemble of M and $M + 1$ randomly chosen pre-trained models.

While our stability analysis captures how consistent a given estimator is in ranking the easiest/hardest samples under model variation, it says nothing about consistency of this

ranking across hardness estimators. To investigate this, we compute pairwise Jaccard overlaps between the pruned sets obtained by different estimators :

$$\text{Jaccard}(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \quad (4.3)$$

This reveals how much the outcome of hardness-based part of the pipeline—i.e., the actual data selected or removed—would change if one switches from one estimator to another. In doing so, we assess the functional similarity between estimators, independent of their individual stabilities. Just like in the previous cases, we repeat the experiments 5 times with pruned indices being obtained from estimates produced by distinct ensembles of 10 randomly chosen pre-trained models.

To summarize, this set of instance-level benchmarking experiments addresses the following research questions:

- How stable are model-based hardness estimates under varying ensemble sizes, and how can we reliably determine an appropriate ensemble size without retraining downstream models?
- To what extent do different model-based hardness estimators agree on which samples are easy or hard?

4.4.3 Experimental Setup

We evaluate all hardness estimators using CIFAR-10 and CIFAR-100 datasets, using their standard training/test splits with no additional validation set. Each image is normalized using the dataset-specific mean and standard deviation. We apply basic data augmentations during training, specifically random horizontal flips and 32×32 random crops with 4-pixel padding. While performing experiments on larger and more complex datasets like ImageNet would better verify the generality of our conclusions we note that the usage of simple datasets is more established in relevant literature, with very few works on hardness-aware learning involving ImageNet or other, more complex, datasets.

We focus on CIFAR-10 and CIFAR-100, as their use has been well established in the relevant literature. While performing our experiments on ImageNet would provide better generalization information

Our classifier architecture is a variant of ResNet-18, modified for low-resolution inputs. Specifically, the initial 7×7 convolution and max-pooling layers are replaced with a single 3×3 convolution. All models are trained using stochastic gradient descent with momentum (0.9), an initial learning rate of 0.1, and weight decay of 5×10^{-4} . The learning rate is reduced by a factor of 0.2 at epochs 60, 120, and 160. Each model is trained for 200 epochs with a batch size of 128.

To account for randomness in training and evaluate the robustness of hardness estimators, we train ensembles of 20 independently initialized models per dataset. Seeds are assigned deterministically to ensure reproducibility: the i -th model uses seed $42 + 42i$. We store model checkpoints at epoch 20 (for EL2N) and at the end of training for iConfidence, iAUM, iData-IQ, and iLoss. The hardness estimates for the comparison between model- and data-based

hardness estimators (Section 4.5.3) and overlap experiments (Section 4.5.2) were obtained by averaging over all 20 models.

Learning-dynamics-based estimators are computed using models trained with data augmentation specified before. Although this introduces slight noise into per-sample behavior, we believe this effect is minimal due to averaging across many epochs. Importantly, this choice allows us to reuse the same ensemble for future downstream tasks, which require training on augmented data to replicate real-world settings. All other estimators (e.g., iAUM, iLoss, data-based methods) are computed on non-augmented inputs. For fairness sake, all of the hardness estimates were obtained on training data. Finally, all experiments are implemented in PyTorch and run on GPUs.

Implementation Considerations for Class-Level Curvature and Volume Estimations

Estimating the Gaussian curvature of a class manifold and its volume introduces several methodological challenges. For curvature, a key challenge lies in selecting the local neighborhood size k , which determines the manifold patch used for local PCA in CAML. A constant k across classes is appealing for consistency but is expected to lead to unreliable estimates for classes with vastly different intrinsic dimensions (ID). Small values of k can destabilize estimates in high-ID classes, while large k values flatten curvature in low-ID classes, as neighborhoods span larger, less curved regions of the manifold. Furthermore, k must be at least as large as the target number of principal components to make PCA feasible, and should generally be higher to ensure numerical stability.

To strike a balance, we adopt a class-conditional strategy: for each class, we set $k = 2 \times \text{ID}_{\text{class}}$, where ID_{class} is the class-level intrinsic dimension. This proportionality ensures that k scales with manifold dimensionality while avoiding manual tuning. Our choice of the scalar factor $\alpha = 2$ is motivated by prior work by (Kaufman & Azencot 2023) who used it when estimating curvature of different datasets. On the other hand, for our experiments involving applying data-based estimators on PCA-transformed spaces we opt for uniform k across all classes to ensure fair comparison across volume, curvature, and ID estimates. Specifically, we set $k = 2d$, where d is the number of retained principal components (fixed across all classes). Although our approach follows intuition and draws from prior recommendations, the impact of choosing class-specific versus global neighborhood sizes k , as well as the optimal setting of the scaling factor α , remains largely unexplored in the literature. Addressing these questions would require a systematic theoretical and empirical analysis, which we consider a valuable but separate research effort outside the scope of this thesis.

When it comes to estimating the volume, the challenge comes from computing the determinant from Equation 3.6. This is because Σ is high-dimensional and badly conditioned, making $\det(I + \Sigma)$ numerically unstable. As a result, the computation may return infinity rather than a valid estimate, rendering any correlation analysis meaningless. We fix this by employing the log-det trick via eigenvalues:

$$V_c = \frac{1}{2} \log_2 \det(I + \Sigma) = \frac{1}{2} \log_2 \det \left(\prod_i (1 + \lambda_i) \right) = \frac{1}{2} \sum_i \log_2 (1 + \lambda_i). \quad (4.4)$$

This allows us to avoid computing the determinant directly, making the formulation significantly more numerically stable.

4.5 Results and Discussion

4.5.1 Stability on Top- t Most Extreme Samples

We begin by evaluating the stability of data pruning under different hardness estimators using the Pruning Change Rate (PCR) defined in Equation 4.1. Figure 4.3 reports PCR values for various ensemble sizes and pruning thresholds, distinguishing between easy and hard sample pruning on CIFAR-10 and CIFAR-100. We also report the changes to Absolute Difference as a function of ensemble sizes in Figure 4.4.

A number of consistent trends emerge. First, class-level averaged Absolute Differences are lower by one degree of magnitude on CIFAR-100 than on CIFAR-10, likely stemming from lower number of samples per class. Second, stability consistently increases with the pruning threshold—regardless of whether we target easy or hard samples. This observation aligns with intuition: selecting the top- t most extreme samples requires less precision as t increases, leading to higher stability. This is a crucial, as when identifying problematic samples, which represent a small minority of the dataset, it is common to rely on hardness estimates. Lastly, we observe that when Forgetting is excluded, estimators that leverage learning dynamics (column 1) are significantly more stable in both identifying the most extreme samples and producing resampling ratios than their instant counterparts (column 2), clearly highlighting the benefit of involving learning dynamics in hardness estimation. This happens because instant estimators are inherently limited by their reliance on a snapshot of a fully trained model, whereas their dynamics-based counterparts can exploit patterns that evolved throughout training.

A notable outlier among estimators that leverage training dynamics (column 1 of Figure 4.3) is Forgetting, which exhibits the highest instability across all pruning settings. This is especially pronounced on CIFAR-10 when identifying the easiest samples, where Forgetting performs similarly bad to instant estimators that lack access to learning dynamics altogether. The root cause of this large instability when identifying the easiest samples on CIFAR-10 is that a large number of samples exhibit identical or near-identical forgetting rates. Specifically, in an ensemble of eight models, 4,163 samples have a forgetting rate of 0.0, and 3,430 samples have a forgetting rate of 0.125. Consequently, when pruning 10% of the dataset (5,000 samples), Forgetting provides no meaningful way to rank these 3,430 samples, explaining the poor stability of Forgetting observed by Seedat et al. (2024). Due to this reason, the lack of stability of Forgetting becomes more pronounced for simpler datasets, which is why we do not notice similar trend on CIFAR-100.

Crucially, our results diverge from those of Toneva et al. (2019), who reported correlations of 89.2% between single-seed Forgetting estimates and up to 95% with five seeds. In contrast, our results reveal substantially higher variability: transitioning from single- to double-seed estimates changes the composition of the top-10% easiest samples by over 60% on average, and of the top-10% hardest samples by more than 30%. Even when increasing from five to six seeds, we still observe deviations exceeding 40% and 15%, respectively. These findings back our concerns regarding unreliability of correlation as a measure of stability of hardness estimates. We argue that, if one aims to perform downstream-task-based benchmarking, it is only natural to evaluate stability with respect to the specific downstream objective — for instance, stability of top- k identification or deviations in resampling ratios, as done here. Our findings also indicate that hardness estimator stability is not uniform but varies depending

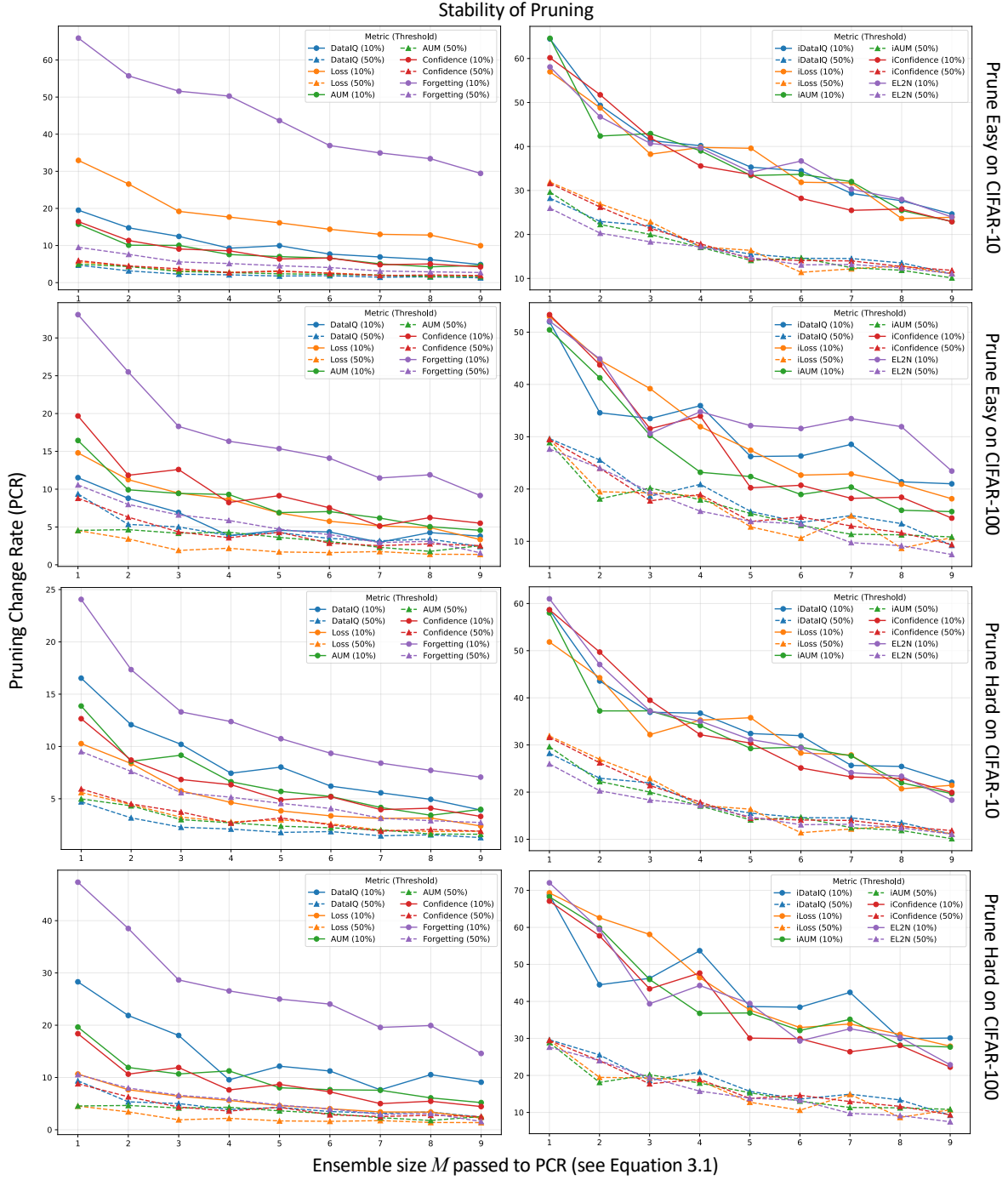


Figure 4.3: Percentage change in the pruned indices (Eq. 4.1) when increasing the ensemble size by one, across two pruning thresholds (10%, 50%) and two scenarios (easy vs. hard pruning). High PCR indicates low stability of hardness estimates. We find that: (i) Forgetting is very unstable at identifying the easiest samples on simple datasets; and (ii) instant estimators are generally less stable than those using training dynamics. Results were averaged over five ensembles of randomly chosen models and std is not reported for clarity.

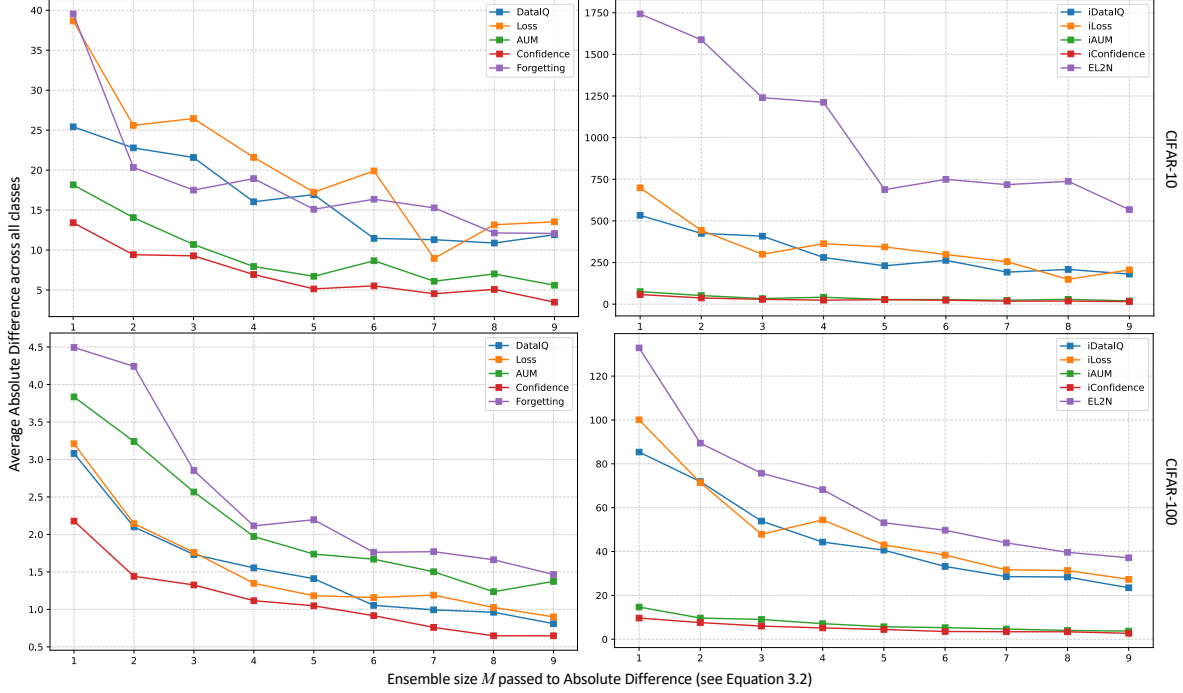


Figure 4.4: We report changes to the sample count of a class after hardness-based resampling (Absolute Difference) averaged across all classes when increasing the ensemble size used to estimate hardness by one. High Absolute Difference indicates low stability in resampling ratios and class-level hardness estimates. Unlike in Figure 4.3, we find Forgetting to be competitive with other dynamics-based hardness estimators indicating that struggling at identifying top- t easiest samples does not translate to difficulty in estimating class-level hardness. At the same time we notice large instability of EL2N, despite this estimator yielding similar stability to other hardness estimators at the instance level. The values are averaged over five ensembles of randomly chosen models and std is not reported for clarity.

on which part of the hardness spectrum is being targeted—whether the focus lies on the hardest, easiest, or all samples. Forgetting shows the greatest instability when identifying top- k extreme samples, while EL2N is the least stable when estimating resampling ratios. This extends the insights of Seedat et al. (2024), who demonstrated that estimator performance (rather than stability) varies across downstream tasks. Whether these two properties — stability and task performance — are correlated remains an open question.

Another important point is that while stability analysis itself varies depending on the part of the hardness spectrum being targeted, its practical relevance ultimately depends on the downstream task in question. For tasks that focus on hard-sample identification—such as detecting mislabeled or out-of-distribution instances—stability is likely to be particularly valuable, as inconsistent identification of these rare and influential samples could undermine the reliability of any corrective intervention. In contrast, for tasks centered on easy-sample selection, stability plays a much smaller role. Prior work on data pruning has shown that model performance remains largely unaffected by which specific easy samples are removed, provided they are indeed easy (Sorscher et al. 2022). Thus, while stability currently appears

most consequential for tasks emphasizing the hard end of the spectrum, this observation should not be mistaken for a universal rule. Future downstream applications may well emerge where stability among easy samples proves equally critical.

Taken together, these findings motivate our choice of AUM as the hardness estimator in the following chapter. Among all evaluated estimators, it exhibited consistently high stability across both downstream tasks, aligning with the high precision reported by Seedat et al. (2024). Its robustness to both task and seed variation makes it a reliable foundation for the resampling and reweighting experiments that follow. To summarize, the main insights from this experiment are as follows:

- Due to experimental design, Forgetting is one of the most unstable model-based hardness estimator, especially when applied to simple datasets.
- Overall, estimators that utilize learning dynamics are significantly more consistent at identifying the most extreme samples than instant alternatives, with Forgetting being the only major exception.
- Just because hardness estimator is stable at instance level doesn't imply it will also be stable at the class level, and vice versa.

4.5.2 Verifying Estimator Agreement Through Jaccard Overlap

We now turn to analyzing the overlap between pruned indices produced by different hardness estimators, as shown in Figure 4.5. Each plot visualizes the pairwise overlap between pruning sets obtained from distinct estimators, across a range of thresholds, and separately for easy and hard sample pruning on CIFAR-10 and CIFAR-100.

These results reveal substantial variability in estimator agreement. At low thresholds (e.g., top 10%), overlap between estimators spans a wide range: while some pairs (e.g., Loss and AUM on hard samples) achieve up to 70% overlap on CIFAR-10, others share less than 10% of pruned indices on CIFAR-100. This sharp divergence underscores that different estimators often disagree on which samples are the most extreme. Agreement improves with increasing threshold, indicating that rankings converge somewhat when more samples are included. Importantly, no universal pattern emerges: estimators that agree in one setting (e.g., Loss and AUM on hard pruning) may diverge in another (e.g., the same pair under easy pruning). These inconsistencies are also dataset-dependent, suggesting that estimator agreement is sensitive not only to the hardness subtype but also to task complexity. These results make it clear that choosing appropriate hardness estimator can potentially have a profound impact on downstream performance as they rarely tend to agree with each other when identifying the most extreme samples.

We also note that overlap between instant estimators and those that incorporate learning dynamics remains consistently low, especially at smaller thresholds (column 3). Together with the observations from our pruning stability experiment this emphasizes a key trade-off: while instant estimators such as iConfidence or iData-IQ are computationally cheap and require only a pretrained model, or ensemble thereof, they fail to identify the same top- t hardest or easiest samples as estimators that utilize the learning behavior. In contrast, estimators like AUM or Forgetting yield fundamentally different (and arguably more meaningful) rank-

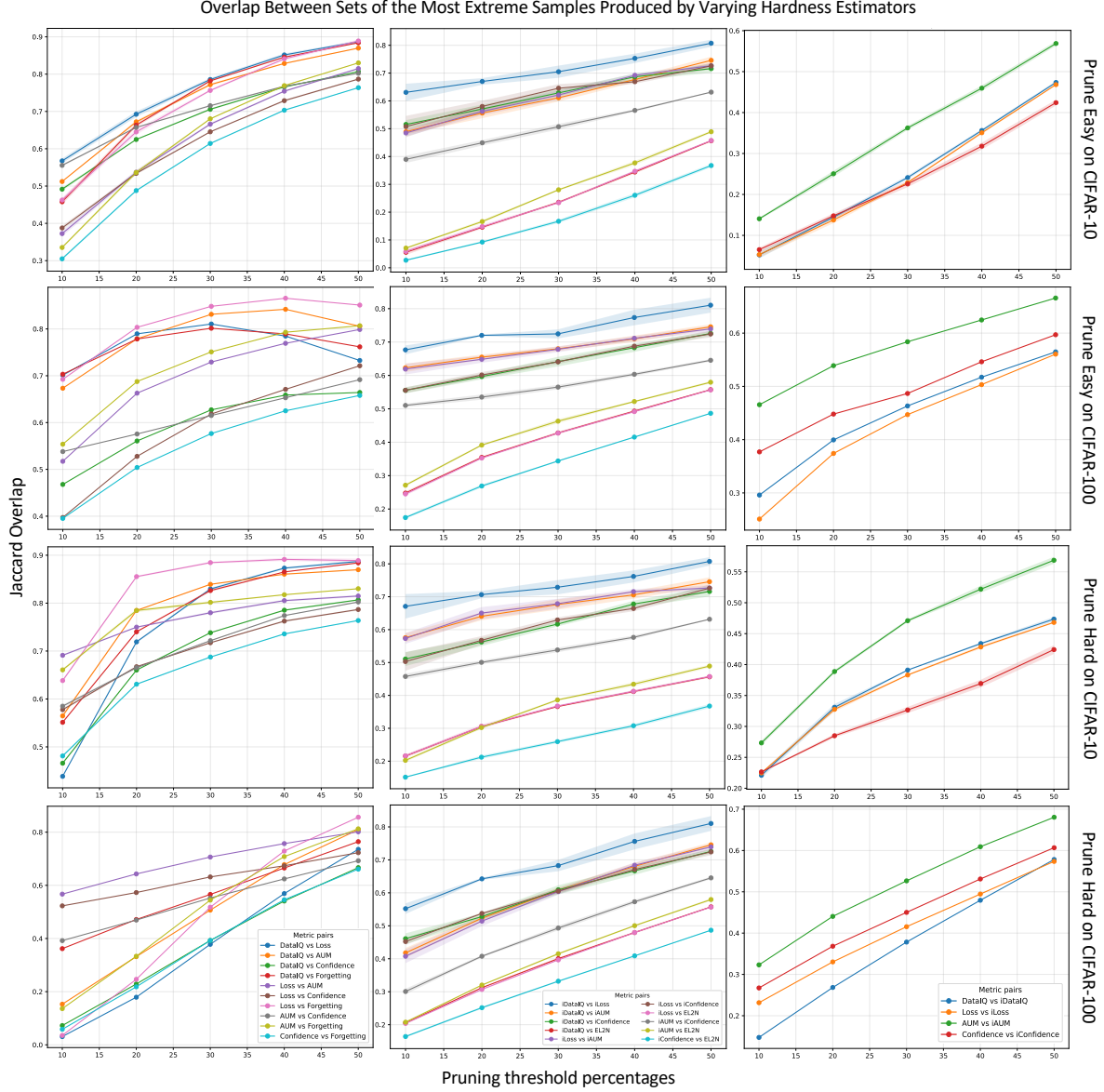


Figure 4.5: We report Jaccard overlap (Eq. 4.3) between the set of pruned samples selected by various hardness estimators across two scenarios (easy vs. hard pruning) and five pruning thresholds. Low overlap indicates strong disagreement on which samples are considered most extreme, which can significantly affect downstream tasks. Notably, (i) estimator choice heavily influences pruning outcomes (columns 1 and 2), and (ii) incorporating learning dynamics (e.g., AUM vs. iAUM) leads to marked shifts in selected samples (column 3).

ings—but at the cost of retraining the model. This trade-off between computational efficiency and estimator fidelity remains a central challenge in practical hardness-based interventions.

To summarize, the main insights from this experiment are as follows:

- The choice of hardness estimator is very important, as the estimators rarely agree in

terms of identifying the most extreme samples.

- Modifying the hardness estimation by incorporating learning dynamics into the formula (e.g., iAUM $\rightarrow$ AUM) significantly alters the set of samples identified as the most extreme.

4.5.3 Benchmarking Model- and Data-Based Estimators

We finish by measuring the correlation between class-level hardness estimates, obtained by both data- and model-based estimators, using Pearson and Spearman Rank Correlations. Figure 4.6 visualizes these correlation values for CIFAR-10 and CIFAR-100.

Across most settings, we observe that Pearson and Spearman rank correlations yield highly consistent conclusions regarding the similarity between class-level hardness estimates. When one correlation is high, the other tends to be similarly strong; when one is weak, so is the other. This co-movement suggests that estimator pairs which yield similar rank orderings also tend to preserve their relative magnitudes, and vice versa.

One of the most striking findings is that while estimator pairs exhibit low overlap in the specific samples they rank as easiest or hardest—as per Figure 4.5—their class-level hardness estimates tend to be much more strongly correlated. A likely explanation is that class estimates are often obtained by averaging over instance-level estimates, which greatly reduced the impact of various noises and biases.

In contrast to model-based estimators, data-based hardness measures—such as local intrinsic dimension, curvature, and volume—exhibit little to no correlation with both *Subjective Accuracy* and *Objective Accuracy*. This is true across both CIFAR-10 and CIFAR-100, and holds regardless of whether Pearson or Spearman correlations are used. Moreover, we observe that class-level ID and Volume estimates themselves are uncorrelated, despite being theoretically related through manifold scaling laws. This discrepancy lends further support to our hypothesis from previous chapter that data-based estimators break down when applied to high-dimensional raw data, likely due to the curse of dimensionality.

In an attempt to mitigate the curse of dimensionality, we applied data-based estimators to PCA-transformed class manifolds. Specifically, we first performed PCA for each class separately reducing its dimensionality to d ranging from 10 to 100, and then recomputed the ID, volume, and curvature. We then recomputed the correlations between the updated data-based estimates and the accuracy measures, as well as the correlation between ID and volume, as a function of d (see Figure 4.7). We use different d range for curvature than ID and volume due to higher computational complexity of CAML.

We expected that correlations between data-based estimates and class-level accuracy would be insignificant for large d and would gradually become more significant as d decreased, reflecting improved reliability in lower-dimensional spaces. This expectation holds only for volume, and even then the trend is weak. For intrinsic dimension (ID), both its correlation with accuracy and with volume change in a highly erratic fashion, with no consistent upward or downward trend. On CIFAR-100, ID–volume correlation shows a slight movement from ≈ 0.1 to ≈ -0.1 , but this change is so small and in the wrong direction that it cannot reasonably be called a trend. Curvature–accuracy correlations are even more static, varying by less than 0.1 across the entire range of d .

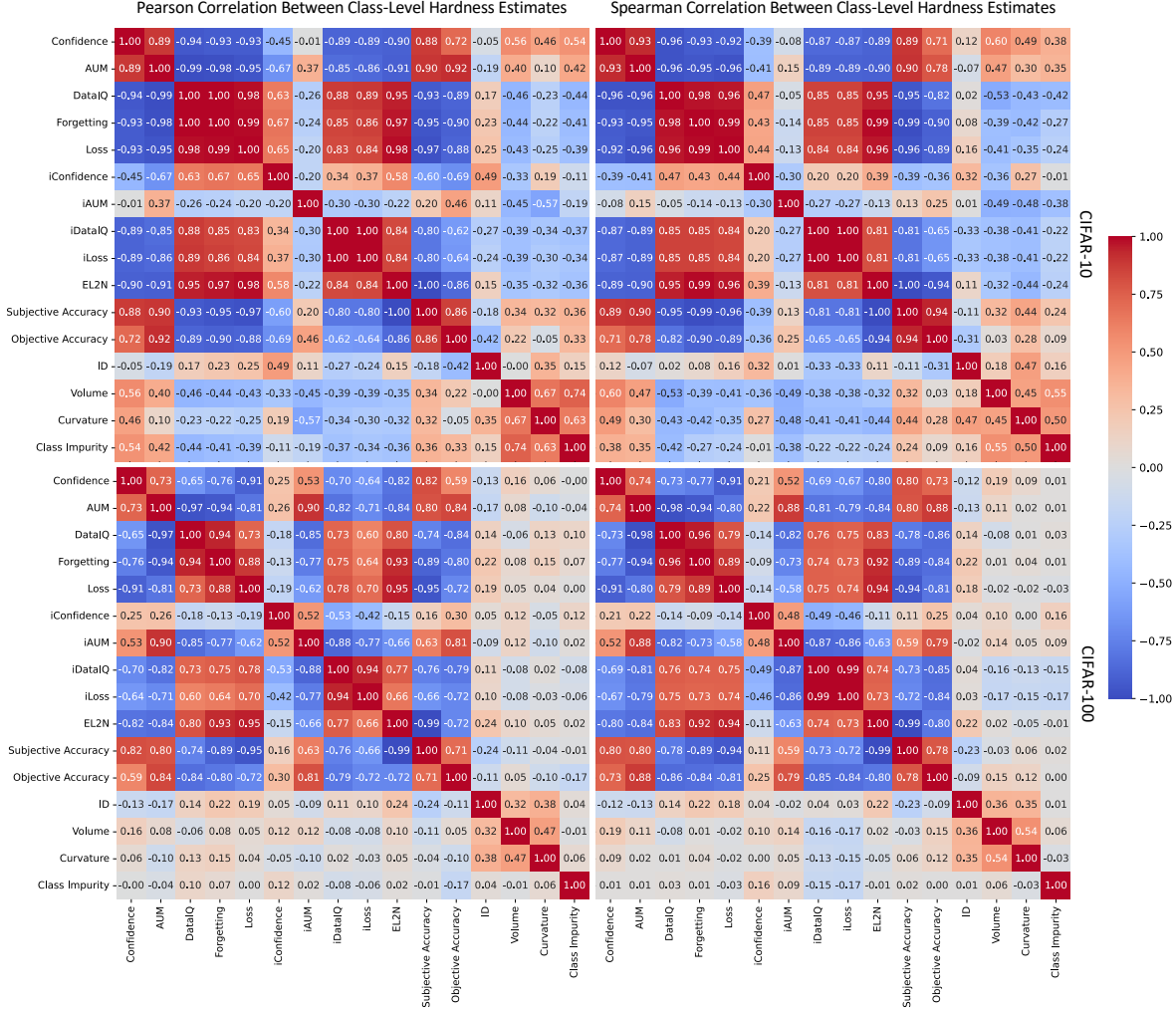


Figure 4.6: We report the Pearson and Spearman rank correlations between all pairs of class-level hardness estimates obtained from raw input data, including their correlations with Subjective Accuracy and Objective Accuracy, which serve as proxies for ground truth. Compared to Fig. 4.5, the stronger agreement between estimators suggests that class-level hardness is easier to estimate than instance-level hardness. On top of that, the weak correlation between volume and ID — despite theoretical expectations of alignment — further supports the view that data-based estimates on raw input are unreliable.

Overall, it appears that PCA reduction does not address the fundamental issue: correlations between data-based estimates and accuracy remain weak, and ID remains largely uncorrelated with volume. While we do observe occasional spikes (e.g., ID–accuracy below -0.6 and ID–volume above 0.7 for specific d on CIFAR-10), these are isolated and embedded in otherwise chaotic behavior, making it likely that they are statistical artifacts rather than meaningful effects. This highlights that addressing the issue of poor performance of data-based estimators requires more sophisticated dimensionality reduction techniques that cause less alternations to the manifold geometry.

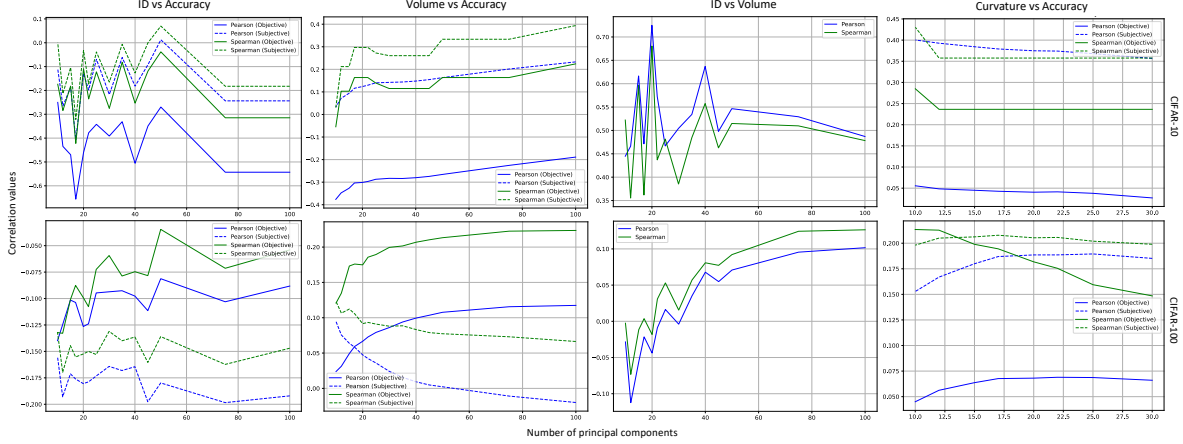


Figure 4.7: We report correlation values between data-based hardness metrics (ID, volume, curvature) and class-level accuracy, as well as between ID and volume, for different PCA projection dimensions. on CIFAR-10 and CIFAR-100. Higher correlation values imply greater performance of data-based estimators. Our key insights are: (i) expected monotonic improvement with decreasing d is absent, suggesting that occasional high-correlation spikes are likely noise rather than a trend, and (ii) the same is the case for ID vs Volume, which are theoretically related, indicating that PCA reduction does not consistently improve data-based estimators.

To summarize, the main insights from this experiment are as follows:

- Data-based estimators correlate poorly with accuracy, likely due to unreliable estimates on raw input data, as evidenced by lack of correlation between estimated IDs and Volumes.
- Projecting class manifolds via PCA fails to robustly improve the performance of data-based estimators.

4.6 Conclusion

In this chapter, we conducted two key benchmarking experiments. The first introduced the first direct comparison between data-based and model-based hardness estimators. Our results show that, in their current form, data-based estimators are not competitive with model-based ones—likely due to the curse of dimensionality, as discussed in the previous chapter. To test a simple mitigation, we applied data-based estimators to PCA-reduced spaces, expecting lower intrinsic dimensionality to improve reliability. However, we found no consistent performance gains. Moreover, even theoretically related metrics such as intrinsic dimension and volume, which should correlate strongly, showed negligible or inconsistent correlation in practice—further evidence of estimator unreliability and a sign that deeper methodological changes are needed.

The second experiment examined the robustness of hardness estimates. We found that stability of single-seed estimates is highly dependable on the downstream task and the choice of the estimator. While prior work suggests that these instabilities do not hinder pruning

performance, its effects on other downstream applications—such as noise detection or curriculum learning—remain unclear. Given this uncertainty and the severity of the instability, we recommend using ensembles of models to obtain more robust hardness estimates, as well as more careful stability and robustness analysis of obtained hardness estimates.

Chapter 5

Can Hardness-Based Resampling Improve Fairness in Balanced Setting?

5.1 Introduction

Class bias, where class-level hardness is not uniform across classes resulting in discrepancies in accuracy across classes, is a crucial problem in machine learning. This issue has traditionally been explored in the context of imbalanced datasets, where classes are represented by unequal numbers of samples (Kubat 1997). These imbalances naturally lead to models favoring classes with more samples, which becomes particularly evident once the imbalance ratio—defined as the number of instances from the most populous class divided by the number of samples from the least populous one—crosses a certain threshold. However, recent studies challenge the assumption that the imbalance ratio is the primary cause of class bias (Sinha et al. 2020, 2022), revealing that such bias persists even in balanced datasets (Ma, Jiao, Liu, Yang, Liu & Li 2023), as illustrated in Figure 5.1.

To better understand the phenomenon of class bias, we return to the geometric perspective introduced in Chapter 3. From this standpoint, hardness differs across classes due to geometric and topological differences between them. Consequently, for each class, a desired performance level corresponds to a specific sample complexity—the number of samples required to attain that performance level with high probability. If the available data falls below this threshold, generalization beyond the specified generalization error is not theoretically guaranteed, which can contribute to poor performance (Balcan et al. 2010). Conversely, exceeding the sample complexity may lead to data redundancy (Friedman & Yakhini 2013, Hanneke 2009). This implies that simply maintaining balanced class sizes does not necessarily yield fair or optimal performance: some classes may still be underrepresented relative to their intrinsic complexity.

From this perspective, fairness may be improved not by enforcing equal sample counts, but by resampling according to class hardness—oversampling harder classes and undersampling easier ones. It should be noted that this reasoning directly challenges the common belief that maintaining data balance is always beneficial for performance. To operationalize such hardness-based resampling, we need to approximate the sample complexity of each class.

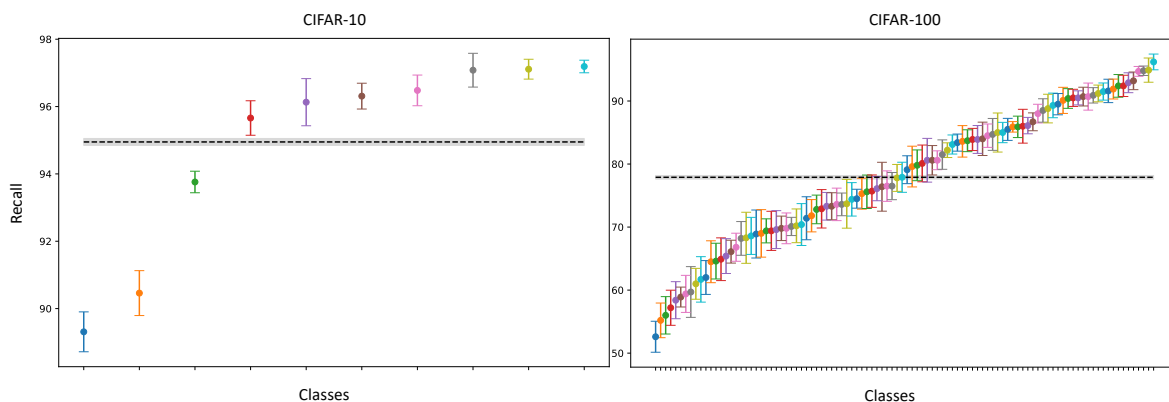


Figure 5.1: Training an ensemble of ten ResNet18 networks on CIFAR-10 (left) and CIFAR-100 (right) reveals large recall gaps across classes, despite the balanced nature of these datasets. Paired with significantly larger recall gaps across classes for CIFAR-100 than CIFAR-10, this shows class- and dataset-level hardness discrepancies, which we call hardness imbalance. We highlight the (very clearly) visible negative effects of this imbalance, even on datasets that are data-balanced, and believe that it can be addressed by hardness-based resampling—oversampling hard classes and undersampling easy ones.

While existing work provides methods for estimating generalization error given sample size and probability (Neyshabur et al. 2018, Arora et al. 2018, Bisla et al. 2021), estimating the required sample size for a target accuracy (i.e., sample complexity) remains intractable. This is because theoretical estimates rely on strong distributional assumptions, such as i.i.d. sampling (Vapnik 1999), which rarely hold in real-world settings due to data biases (Dominguez-Catena et al. 2024, Sarriidis et al. 2024), sample correlations (Dundar et al. 2007, Siddiqi et al. 2023), and distributional shifts (He et al. 2021). Moreover, such estimates depend on model capacity, also known as expressivity, but defining capacity for deep neural networks is notoriously difficult (Gamba et al. 2022). Unlike traditional models, deep networks exhibit overparameterization (Garg et al. 2022) and implicit regularization (Neyshabur 2017), rendering classical capacity measures (such as VC-dimension (Blumer et al. 1989) and Rademacher complexity (Neyshabur 2017)) either intractable or poorly predictive of real-world behavior (Neyshabur 2017, Zhou 2021). Given these difficulties, we turn to empirical hardness estimators, reviewed in Chapter 3, as practical proxies for sample complexity: classes with a higher proportion of hard samples intuitively require more data for reliable generalization.

This reasoning leads us to one of the main research questions of this thesis: *can hardness-based resampling improve fairness on balanced datasets?* To investigate this, we design two complementary case studies. The first uses a controlled setup in which we emulate access to the sampling distribution by treating pruned samples as a holdout set of unseen data. The second simulates a realistic scenario with no access to the underlying sampling distribution, reflecting the common constraint of real-world datasets. Together, these two settings allow us to disentangle the effects of resampling methodology from those of data availability and quality.

We evaluate the resulting models using five fairness metrics and test for statistical significance using paired t-tests. Our results show that hardness-based resampling reduces recall

gaps across classes while increasing precision gaps. Improvements in F1-based fairness are possible but hinge on the quality of data used for oversampling—simple SMOTE or random oversampling do not meet these conditions, and even EDM, a state-of-the-art diffusion model, underperforms compared to the holdout set of real samples. Crucially, the improvements to fairness are not restricted to the most extreme classes but are distributed across all classes. Finally, our results suggest that hardness-based resampling can also improve average recall, precision, and F1, but these gains seem to depend on the use of reliable hardness estimators—to guide both data pruning and the computation of resampling ratios—and high quality data for oversampling. Altogether, these observations highlight that addressing hardness imbalance is a more complex task than addressing data imbalance. Not only does the outcome depend more strongly on the quality of the data being oversampled, but it also introduces an additional dependency on the accuracy of the hardness estimates themselves.

Contributions

This chapter makes the following contributions:

- We introduce hardness imbalance as a complementary perspective to the classical data imbalance to understand and address class bias.
- We design a *controlled experimental framework* via two case studies that enables systematic analysis of hardness-based resampling under varying degrees of induced imbalance and sample quality, effectively simulating partial access to the underlying data distribution.
- Through empirical analysis, we provide evidence that *hardness-based resampling can substantially reduce class bias even in data-balanced datasets*, with its effectiveness governed by the degree of induced imbalance, the quality of oversampled data, and the reliability of the hardness estimator.

5.2 Background: From Class Imbalance to Hardness Imbalance

5.2.1 Types of Imbalance

Data imbalance is a long-standing and well-studied challenge in machine learning, in which the distribution of data across classes often follows a long-tailed pattern, similar to a Pareto distribution (Li et al. 2022). This between-class data imbalance often arises from various factors, such as the natural scarcity of certain classes, the time and cost involved in acquiring them, or biases introduced during sampling (Johnson & Khoshgoftaar 2019). As a result, data imbalance is a common obstacle in various practical applications, including healthcare diagnostics (Fotouhi et al. 2019), fraud detection (Makki et al. 2019), and computer vision (Kubat et al. 1998), underscoring the importance of continued research in this area.

Although the crucial problem, the between-class data imbalance is not the only type of imbalance cursing real-world datasets, as we earlier mentioned in Chapter 3. Another important imbalance is disjunct-based within-class data imbalance, which was first reported by Holte et al. (1989). This data imbalance is characterized by nonuniform sampling from

class submanifolds, resulting in data clusters (disjuncts) of various sizes within each class, with small disjuncts being harder to learn. Japkowicz (2001) reported that, at the time, *no works took into consideration the fact that both between-class and within-class imbalances may occur*. In fact, multiple works report lack of standardization (Guo et al. 2008) or research (Leevy et al. 2018, Ren et al. 2023) on disjunct-based imbalance.

While data imbalances have received substantial attention in the literature, the field’s focus on this specific type of imbalance has arguably diverted attention from the deeper underlying goal. That is because the primary objective of rebalancing strategies should not be to produce datasets with equal numbers of examples per class, but rather to mitigate disparities in model performance across classes. Indeed, our taxonomy from Chapter 3 highlights that both of the above-mentioned data imbalances represent just two of many factors that contribute to sample—and by extension class—hardness. In many cases, addressing these data imbalances leads to improved performance, particularly when they are the primary driver of hardness. However, this is not always the case. Recent works have shown that performance gaps can persist even in perfectly balanced datasets (Sinha et al. 2020, 2022), and that underrepresented classes may sometimes achieve higher accuracy than majority ones (Ma, Jiao, Liu, Yang, Liu & Li 2023). These findings evidently show that class imbalance is not the only factor affecting class difficulty, aligning with insights from our taxonomy.

5.2.2 Standard Techniques for Imbalance Correction

The strategies that mitigate the effects of data imbalances can generally be divided into two categories. One major category includes specialized loss functions, such as Focal Loss (Lin et al. 2017), Dual Focal Loss (Tao et al. 2023), and Equilibrium Loss (Feng et al. 2021), all of which are explicitly motivated by the need to emphasize underrepresented classes. However, in practice, these losses focus on instances that are difficult to classify—Focal Loss penalizes based on confidence, Dual Focal Loss based on margin, and Equilibrium Loss based on mean accuracy—thus effectively targeting hardness imbalance. The problem is that, to the best of our knowledge, all evaluations of these methods are performed on imbalanced datasets, confounding the effects of sample size with intrinsic difficulty. This lack of an ablation study makes it unclear whether their observed gains are due to correcting class imbalance or mitigating hardness disparities. A closely related example is the Class-Wise Difficulty-Balanced Loss (Sinha et al. 2020, 2022), which also uses class-level accuracy to estimate hardness and, unlike other works, acknowledges that sample counts may not reflect true class difficulty. However, it ultimately falls into the same empirical trap: all evaluations are conducted on imbalanced datasets.

Another class of strategies for mitigating data imbalance is resampling, which modifies the number of samples per class to create a more balanced dataset. This can be achieved through undersampling, by removing samples from majority classes, oversampling, by increasing the number of samples for minority classes, or both (Sasada et al. 2020). Undersampling methods are typically simple and follow the same principle as data pruning: many examples are redundant and can be removed without degrading performance, making even random undersampling a popular approach (Prusa et al. 2015, Hasanin & Khoshgoftaar 2018). However, undersampling risks discarding useful information (Liu et al. 2008), and therefore only addresses representation bias—neglecting the fact that minority classes may lack sufficient data to learn meaningful patterns. Oversampling addresses this limitation and is therefore more

commonly used. This preference is further supported by empirical studies, which show that oversampling generally leads to better performance gains than undersampling (Mohammed et al. 2020).

Oversampling can take several forms. One can simply replicate existing data (random oversampling), apply data augmentation to create synthetic variants (Ahn et al. 2023, Vu et al. 2024), or interpolate between minority class samples (e.g., SMOTE (Chawla et al. 2002) and its variants such as Borderline-SMOTE (Han et al. 2005), ADASYN (He et al. 2008), or DeepSMOTE (Dablain et al. 2022)). However, due to the usage of duplication or interpolation these methods typically do not generate genuinely new examples and may lead to overfitting. To overcome this limitation, recent works have explored generative models (e.g., Generative Adversarial Networks (Goodfellow et al. 2020), or diffusion models (Ho et al. 2020)) as a way to synthesize novel samples that better capture the true class distribution. These generative approaches have increasingly become a focal point for addressing data scarcity (Ali-Gombe & Elyan 2019, Zheng et al. 2020, Marchesi et al. 2023). Unfortunately, as with cost-sensitive learning, most resampling methods have been evaluated exclusively on imbalanced datasets.

5.2.3 Measuring Quality of Generative Models

Generative models aim to produce novel synthetic samples, often with the goal of filling gaps in the data manifold caused by non-uniform sampling distributions. Their architectures have developed over the years, moving from Variational Autoencoders (Doersch 2016) and Generative Adversarial Networks (Goodfellow et al. 2020) to Diffusion Models (Ho et al. 2020), which are the current state of the art in this field. Considering their purpose, it is imperative to use metrics that measure the quality of the generated samples. The two most commonly used ones are the Inception Score (IS) (Salimans et al. 2016) and the Fréchet Inception Distance (FID) (Heusel et al. 2017). The former is a proxy-based metric that measures the entropy of a batch of synthetic images. A higher IS is interpreted as higher-quality synthetic samples and indicates that the Inception V3 model (Szegedy et al. 2016), used as the proxy, was able to correctly and confidently classify the majority of the synthetic samples. The latter works by comparing the distribution of latent representations produced by Inception V3 for real images from a reference set and for the synthetic samples, computing the squared Wasserstein distance between the two multivariate Gaussian distributions. A lower FID means that the distribution of the synthetic samples is very similar to that of the real data.

Despite their popularity, these metrics exhibit important shortcomings. IS implicitly favors generative models that produce samples which the Inception network can classify with high confidence—typically the easier regions of the data manifold. As a result, models optimized for IS may underrepresent rare or ambiguous cases that are crucial for downstream robustness. FID, while conceptually broader, also has fundamental limitations. It measures only global alignment between real and synthetic feature distributions, assuming they can each be represented as a single multivariate Gaussian. This assumption collapses the complex structure of data manifolds, erasing distinctions between classes and overlooking whether harder regions are faithfully captured. Moreover, because FID depends on representations learned by a specific pretrained model, it inherits that model’s inductive biases. For example, if the reference network was exposed to the evaluation dataset or closely related samples during training or fine-tuning, its embedding space may overfit to certain patterns—treating

some complex regions of the real data as deceptively “easy.” In practice, this suggests that FID can overestimate the fidelity of generative models whose samples resemble familiar, low-hardness regions, while failing to penalize insufficient coverage of harder ones.

Consequently, the hardness spectrum of synthetic data is often skewed toward easier samples compared to real data, as demonstrated by Wang et al. (2024). This observation has motivated the development of architectures that explicitly encourage the generation of harder samples by modifying the training process (Vandenhende et al. 2019, Pennisi et al. 2021, Ferracci et al. 2024, Wang et al. 2024). Yet research in this direction remains extremely limited: only a handful of studies have examined how the hardness of synthetic data affects model performance or perceived quality. It therefore remains unclear whether harder synthetic samples are inherently more valuable, whether the optimal hardness should mirror that of real data, or whether pushing generative models toward increasingly hard samples merely risks producing noisy or unrealistic outputs.

5.3 Formal Experimental Design

Let’s define a balanced dataset $\mathcal{D}$ containing a samples in each class. We are interested in addressing class bias—a phenomenon characterized by inequalities in recall observed across classes. More formally, let $r_c := \text{Rec}(M, \mathcal{D}_c^{\text{test}})$ denote the recall of model M on a test set containing only samples from class c . Class bias occurs when some class c performs better or worse than the mean recall across all classes, $\bar{r}_c$. This section outlines the formal framework underlying our experiments, while implementation details, estimator choices, and statistical testing procedures are presented in Section 5.4.

To quantify the degree of class bias, we employ a set of fairness metrics that capture complementary aspects of per-class recall disparities: (i) *gap-based* metrics, which compare the easiest and hardest classes, and (ii) *dispersion-based* metrics, which measure variability across all classes.

- **Gap-based metrics:**

- **Maximum gap:**

$$\Delta_m = \max_c r_c - \min_c r_c, \quad (5.1)$$

measures the gap between the highest and lowest class recalls. It captures extreme inequality, making it highly sensitive to outliers.

- **Quantile gap:**

$$\Delta_q = \frac{1}{k} \sum_{i=1}^k r_{(K+1-i)} - \frac{1}{k} \sum_{i=1}^k r_{(i)}, \quad (5.2)$$

where $r_{(i)}$ denotes the i -th smallest element of $\mathbf{r} = (r_1, \dots, r_C)$. This measures the average gap between the k easiest and k hardest classes. By averaging across multiple classes, it reduces sensitivity to single outlier classes while still emphasizing extremes like Δ_m . We set $k = 2$ for CIFAR-10 and $k = 10$ for CIFAR-100.

- **Hardness-based equality gap:**

$$\Delta_{he} = \frac{1}{|\mathcal{E}|} \sum_{c \in \mathcal{E}} r_c - \frac{1}{|\mathcal{H}|} \sum_{c \in \mathcal{H}} r_c, \quad (5.3)$$

where $\mathcal{E}$ and $\mathcal{H}$ denote sets of easy and hard classes identified through hardness-based resampling ratios. This measure directly reflects our experimental design by comparing the mean performance of classes targeted for under- vs. oversampling. It provides a task-specific notion of fairness that is more robust to individual outlier classes than Δ_m or Δ_q .

- **Dispersion-based metrics:**

- **Standard deviation:**

$$\Delta_\sigma = \text{Std}(\mathbf{r}), \quad (5.4)$$

quantifies the overall dispersion of class recalls around their mean. Unlike the gap-based metrics, it provides a hardness-agnostic view of fairness that accounts for all classes simultaneously. However, since it squares deviations, it tends to overweight large differences in skewed distributions, making it more sensitive to outliers than Δ_{he} .

- **Median absolute deviation:**

$$\Delta_{\text{MAD}} = \text{Median}(|r_c - \text{Median}(\mathbf{r})|), \quad (5.5)$$

is a robust alternative to Δ_σ that downweights classes with extreme recall values. It offers a stable measure of overall fairness even in the presence of outlier classes.

While Δ_m and Δ_q are more sensitive to outliers, comparing them to other metrics reveals whether fairness changes are localized to a few extreme classes or spread more uniformly across all classes. In addition, we track the average performance (Recall, Precision, and F1) to contextualize the trade-off between fairness and accuracy. Our objective is therefore to minimize the Δ metrics while maintaining the highest possible class-averaged performance.

We define hardness-based resampling as a data preprocessing operation that breaks the data balance by resampling classes based on their hardness, changing the number of samples in class c to some b_c that is proportional to the true hardness of class c , $\hat{h}_c$. It is composed of two steps: 1) undersampling; and 2) oversampling. Undersampling works by removing the $a' - b_c$ easiest samples from classes where $b_c < a'$. Meanwhile, oversampling is performed by expanding the training set by $b_c - a'$ samples. Naturally, ground-truth hardness $\hat{h}_c$ is inaccessible, forcing us to rely on hardness estimates.

Intuitively, the resampling ratios b_c should be based on the true hardness of a class as follows

$$b_c = \left\lfloor f \left(a' \frac{\hat{h}_c}{\sum_{c'} \hat{h}_{c'}} \right) \right\rfloor, \quad (5.6)$$

with $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ being a function that determines the rate of progression. For instance, an identity function yields linear progression, while logarithmic progression puts more emphasis on easy classes than hard ones, as the differences in hardness across classes would be more amplified for the easiest classes. Since Equation 5.6 does not allow for modifying the degree of the introduced data imbalance we update the number of samples after resampling as follows

$$b'_c = \bar{\mathbf{b}} + \alpha (b_c - \bar{\mathbf{b}}) \quad (5.7)$$

with b'_c being the update sample count for class c and α a scalar that controls the degree of the introduced data imbalance.

To reiterate, we investigate whether the fairness objective can be met by performing hardness-based resampling. This objective has the following key challenges: 1) oversampling techniques struggle with generating genuinely new, i.i.d. data samples; and 2) estimating resampling ratio relies on class-level hardness estimates, the quality of which is hard to ascertain. While the second point is very challenging, and ultimately remains a limitation of hardness-based resampling approach, the first one can be addressed through a carefully designed case study. We therefore design two complementary case studies: Case Study 1, which introduces a holdout set to alleviate the questionable quality of synthetically generated samples, and Case Study 2, which more closely reflects realistic training conditions, where no access to holdout set is provided.

Case Study 1: Access to holdout set. We assume that CIFAR-10 and CIFAR-100 were obtained through i.i.d. sampling from an underlying distribution $\mathcal{X}$. More specifically, for each class c , $\mathcal{D}_c$ is a set of i.i.d. samples drawn from $\mathcal{X}_c$. Now consider a sub-dataset $\mathcal{D}' = \bigcup_{c=1}^C \mathcal{D}'_c$, where $\mathcal{D}'_c \subset \mathcal{D}_c$ and $|\mathcal{D}'_c| = a'$ for each class c , with $a' < a$. Next, define a holdout set $\mathcal{D}_c^{\text{ho}} := \mathcal{D}_c \setminus \mathcal{D}'_c$. We assume that $\mathcal{D}'_c \sim \mathcal{X}_c$ and that $\mathcal{D}_c^{\text{ho}} \sim \mathcal{X}_c$, making $\mathcal{D}_c^{\text{ho}}$ a simulated oracle for new draws from $\mathcal{X}$. The $\mathcal{D}_c^{\text{ho}} \sim \mathcal{X}_c$ assumption allows us to alleviate issue (1). We call this hypothetical situation in which the dataset construction process was stopped early *Scenario A*. In order to simulate it we construct $\mathcal{D}'_c$ by randomly selecting a' samples from each class without replacement. The remaining samples form a disjoint holdout set $\mathcal{D}^{\text{ho}}$. Although subsampling without replacement technically induces weak dependence between samples, we treat the resulting subsets $\mathcal{D}'_c$ and $\mathcal{D}_c^{\text{ho}}$ as approximately i.i.d. in practice. Ultimately, these assumptions play a formal role and do not impact the validity of our conclusions, especially given that CIFAR-10 and CIFAR-100 are themselves not strictly i.i.d. in any meaningful sense. Hence, in this case study we apply hardness-based resampling to $\mathcal{D}'$, which we will call $\mathcal{D}^{\text{CLP}}$ as it was obtained through class-level pruning.

Case Study 2: No access to holdout set. Here we apply hardness-based resampling to the original dataset $\mathcal{D}$. Hence, the only difference from Case Study 1 is that we do not perform CLP before applying resampling. This means that we do not gain access to the holdout set, but is also more realistic.

5.4 Methodology

5.4.1 Detailed Experimental Design

Case Study 1. Design. We prune the dataset of interest using thresholds of 33%, 42%, 50%, 58%, 66%, and 75%. Smaller pruning thresholds are omitted because, for these values, the maximum number of samples after resampling (b_c) exceeds the available sample count a_c in the holdout set for the hardest classes. We set $\alpha = 1$. After pruning via class-level pruning (**CLP**), which removes a fixed number of random samples from each class, we perform hardness-based resampling on $\mathcal{D}^{\text{CLP}}$.

Hardness estimation. Hardness-based resampling relies on hardness estimates at two stages: (i) to compute resampling ratios (Equation 5.6), and (ii) to guide pruning of easy classes. Ideally, both stages—and both case studies—would rely on identical hardness estimates, such as those computed in Chapter 4. However, this is problematic in *Scenario A*, where the subsampling alters the relative hardness of many samples: an originally easy sample may become hard simply due to sampling effects. Conversely, recomputing hardness estimates for each scenario would reduce comparability across case studies and incur significant computational overhead (especially considering the instability of single seed estimates reported in Chapter 4, and cost of producing multiple-seed estimates). We therefore use a single, consistent set of hardness estimates (AUM) across all settings, while acknowledging that this introduces some noise in *Scenario A*.

AUM values are obtained by averaging results from ten trained models from Chapter 4. We adopt AUM as our default estimator, as Chapter 4 demonstrated its high stability, and the benchmark by Seedat et al. (2024) identified it as one of the best-performing hardness estimators.

Oversampling strategies. Because the results of models trained on $\mathcal{D}^{\text{CLP}}$ depend on both initialization seed and random pruning, we train multiple ensembles of modified ResNet-18 models on $\mathcal{D}^{\text{CLP}}$ and on three resampled datasets derived from it. These resampled variants employ: (1) random oversampling, (2) SMOTE, and (3) the holdout set.

Case Study 2. Design. In this case study, we compute resampling ratios using AUM as the proxy for $\hat{h}_c$. Since higher AUM values correspond to easier samples, we use h_c^{-1} instead of h_c to ensure that harder classes receive higher resampling weights (b_c). We set $\alpha \in \{1, 3\}$ for CIFAR-10 and $\alpha \in \{1, 2\}$ for CIFAR-100. Larger α values are used for CIFAR-10 to compensate for its weaker hardness imbalance relative to CIFAR-100.

Hardness estimation. We again employ AUM as the hardness estimator, as argued before.

Oversampling strategies. We employ three main oversampling strategies: (1) random oversampling, (2) SMOTE, and (3) synthetic samples generated using EDM (Karras et al. 2022). Since training a diffusion model is computationally prohibitive, we use a million of synthetic samples made publicly available by Wang et al. (2023)¹. Because diffusion and generative models are known to preferentially generate easy samples (see Section 5.2.3), we further define three EDM sub-strategies:

1. rEDM, where synthetic samples are drawn randomly from the specified classes
2. aEDM, where samples are drawn from the set of $2b_c$ synthetic images whose hardness is closest to the class average (based on AUM)
3. hEDM, where samples are drawn from the set of $2b_c$ hardest synthetic images of class c

5.4.2 Experimental Setup

Datasets. All experiments are performed on CIFAR-10 and CIFAR-100 using their standard train/test splits. No additional validation set is used. Images are normalized using dataset-specific channel means and standard deviations. During training we apply standard augmentations: random horizontal flips and 32×32 random crops with 4-pixel padding.

¹This is also why we do not use diffusion models in Case Study 1, as retraining them for each *Scenario A* is significantly beyond this thesis.

Model and optimization. We use a ResNet-18 variant adapted for low-resolution inputs: the original 7×7 convolution and initial max-pooling are replaced with a single 3×3 convolution. All models are trained with stochastic gradient descent (SGD) with momentum 0.9, initial learning rate 0.1, and weight decay 5×10^{-4} . The learning rate is multiplied by 0.2 at epochs 60, 120, and 160. Training runs for 200 epochs with batch size 128.

Experimental replicates and seeds (deterministic scheme). Model performance in both case studies depends on stochastic components arising from data pruning and resampling. To ensure robustness and enable paired statistical testing, we adopt a deterministic seed scheme that guarantees reproducibility and one-to-one correspondence between models across datasets.

In Case Study 1, the source of randomness differs between the base and resampled datasets. For models trained on the base pruned datasets ($\mathcal{D}^{CLP}$), randomness originates solely from the pruning process. For models trained on the corresponding resampled datasets, randomness arises from both pruning and resampling. Hence, to ensure robust results we generate four independently pruned datasets for each pruning threshold, each using a distinct pruning seed, and train four independently initialized models on each of them (i.e., four pruned datasets $\times$ four models each). Model initialization seeds are computed deterministically as

$$\text{seed}_{\text{model}}(i, j) = 42 + 420,000 \cdot i + 42 \cdot j,$$

where $i \in \{0, 1, 2, 3\}$ indexes the dataset replicate and $j \in \{0, 1, 2, 3\}$ indexes the model within that replicate. These model seeds are independent of both the pruning and resampling seeds. Meanwhile, the pruning seed used to obtain $\mathcal{D}^{CLP}$ is set as

$$\text{seed}_{\text{prune}}(i) = 42 \cdot i.$$

In Case Study 2, randomness stems exclusively from resampling. Most oversampling and undersampling methods (except deterministic SMOTE) yield different datasets when initialized with different seeds. Accordingly, we again generate four independently resampled datasets, each controlled by its own resampling seed, and train four independent models per dataset using the same deterministic model-seed scheme as above. This design yields a total of $4 \times 4 = 16$ model-dataset combinations in both case studies.

Implementation notes. All experiments are implemented in PyTorch.

Controlling data imbalance. For the choice of f in Equation 5.6, we use the identity function, as it is intuitive that a class twice as hard should receive twice as many samples to address the imbalance. While this choice is guided by intuition, we acknowledge that other functional forms may be more appropriate, possibly in a dataset-dependent manner. A systematic investigation of this question is left for future work.

Evaluation. For each trained model we compute class-level Recall, Precision, and F1 on the standard test set. From these per-class scores we compute the fairness metrics defined in Section 5.4.1 (Eqs. for Δ_m , Δ_q , Δ_{he} , Δ_σ , and Δ_{MAD}).

Paired t-test

To move beyond qualitative observations, and to enable consistent comparison across both case studies, we perform paired t-tests. Let X_i denote the fairness metric from model i trained on the base dataset ($\mathcal{D}^{CLP}$ in Case Study 1, and the full dataset in Case Study 2), and let

Y_i denote the same metric from model i trained on the corresponding resampled dataset. Because seeds are matched, we can form $n = 16$ pairs (X_i, Y_i) and define $D_i = Y_i - X_i$.

The null hypothesis is $H_0 : \mu_D = 0$, i.e., resampling has no effect. The one-sided alternative is $H_A : \mu_D > 0$, i.e., resampling improves fairness. We verify whether the null hypothesis can be rejected by applying Student’s t-test, defined as follows:

$$t = \frac{\bar{D}}{s_D / \sqrt{n}}, \quad (5.8)$$

with

$$s_D^2 = \frac{1}{n-1} \sum_{i=1}^n (D_i - \bar{D})^2. \quad (5.9)$$

A t value of 2 indicates that observed average improvement to a specific fairness metric is twice as large as what we’d expect from sampling noise. In other words, the higher the t value the larger and/or more consistent the improvements to fairness as per specific fairness metric. Hence, any negative values of t indicate that applying hardness-based resampling worsened fairness rather than improving it. After that, we compute the p-value, which gives the probability of observing a t statistic at least as extreme as the one obtained, under the null hypothesis. In general, a p-value above 0.05 indicates that null hypothesis cannot be rejected due to lack of statistical significance.

5.5 Results and analysis

In this section, we analyse the results of both case studies, propose possible explanations for the observed trends, and highlight several open questions that emerge from them. Figures 5.2–5.3 present the changes in the fairness metrics defined in Section 5.3 for the first case study on CIFAR-10 and CIFAR-100, while Figures 5.4–5.5 show the corresponding results for the second case study. Tables 5.1 and 5.2 report the outcomes of the Student’s t-test for the first case study on CIFAR-10 and CIFAR-100, and Tables 5.3 and 5.4 provide the corresponding results for the second case study.

5.5.1 Observations

Sample quality is crucial for hardness-based resampling. We first observe a consistent increase in recall-based fairness resulting from hardness-based resampling. In the first case study, we find statistically significant decreases in Δ_m , Δ_q , Δ_{he} , and Δ_σ when using the holdout set for oversampling. Specifically, across all pruning rates, hardness-based resampling improves the recall gap between the average hard and easy class by approximately 0.01 on CIFAR-10 and 0.04 on CIFAR-100. Similar trends appear in the second case study when only the hardest EDM samples are used for oversampling (hEDM): Δ_{he} decreases from 0.043 to 0.031 on CIFAR-10 and from 0.165 to 0.137 on CIFAR-100. Using random EDM samples (rEDM) or adaptive EDM samples (aEDM), whose hardness most closely matches the target class hardness, also frequently improves recall-based fairness, although less consistently. In contrast, random oversampling and SMOTE generally yield smaller or statistically

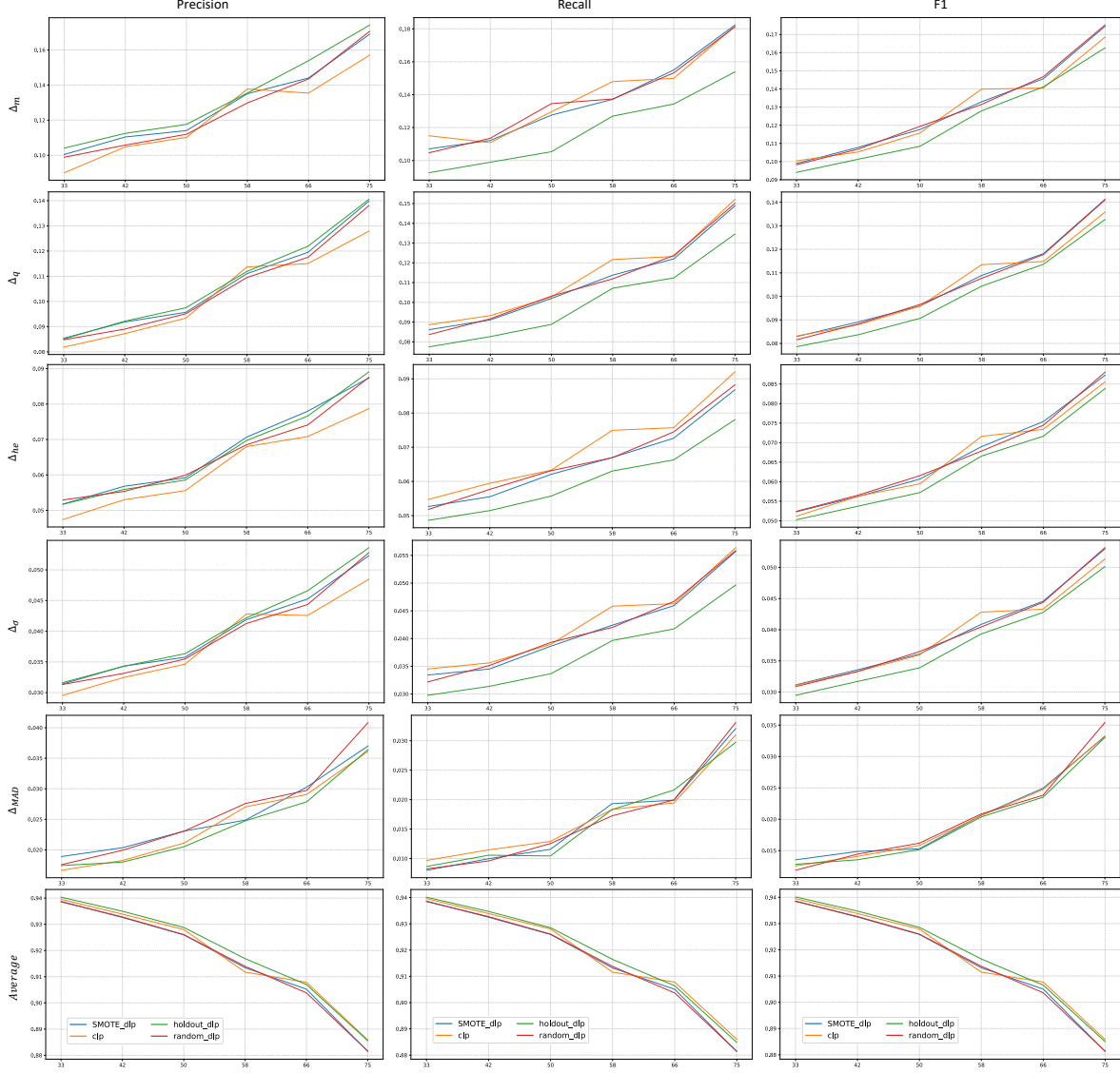


Figure 5.2: Case Study 1 (CIFAR-10): Analyzing changes in various fairness metrics on pruned versions of CIFAR-10 reveals that hardness-based resampling reduces the recall gap across classes at the cost of increasing the precision gap, with the improvements in recall overwhelming the drops in precision as indicated by F1 gap. We find that hardness-based resampling yields similar improvements no matter the pruning rate. Furthermore, it seems to be highly reliant on the quality of samples used for oversampling, as random- and SMOTE-based resampling was not able to meaningfully impact the performance gap. We report means over four ensembles of four models trained on 4 distinct $\mathcal{D}^{CLP}$ and do not report std for clarity.

insignificant changes, and occasionally worsen fairness metrics. The importance of high quality synthetic samples is particularly evident in Figure 5.3, where we see that changing from SMOTE or random oversampling to holdout set doubles the changes in recall-based metrics.

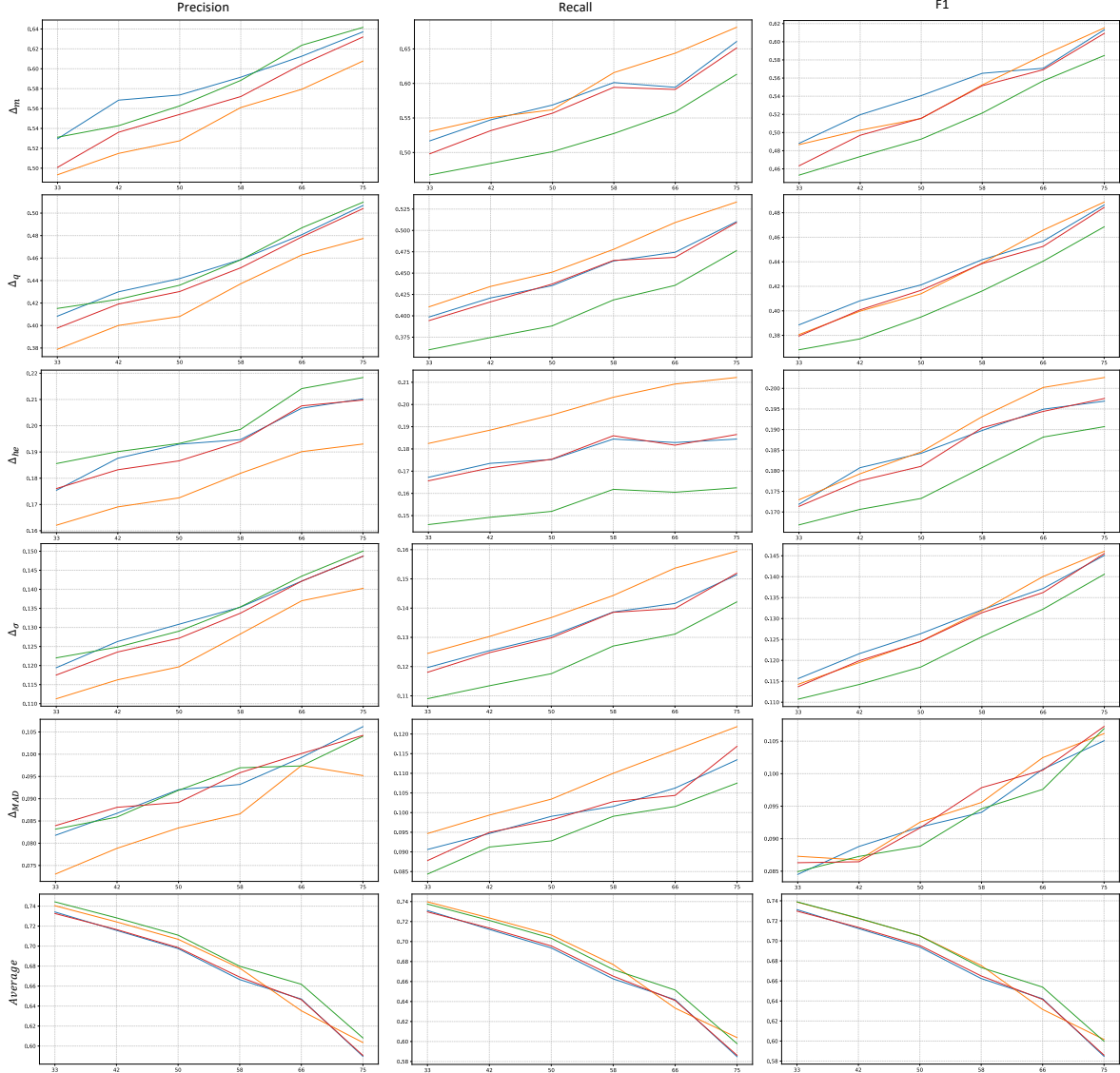


Figure 5.3: Case Study 1 (CIFAR-100): Just like on CIFAR-10 we notice reduction in recall- and F1-based fairness metrics indicating reduction in performance gap across classes. This time the changes are more pronounced, most likely due to more linear-like hardness spectrum on this dataset (see Figure 5.1). The importance of high quality samples also becomes more apparent on CIFAR-100, as resampling using holdout set yields twice the improvement achieved by using SMOTE or random oversampling.

These observations yield two important insights: (i) synthetic data quality is imperative for successful hardness-based resampling; and (ii) when generative models are involved, harder synthetic samples are more beneficial. A possible reason behind this is that harder synthetic samples (hEDM), just like real samples from the holdout set, are more likely to populate the low-density or high-complexity regions of class manifolds. From now on we will focus on recall-based results on holdout set or hEDM, unless specified otherwise.

Table 5.1: Case Study 1 (CIFAR-10): t -statistics (with p -values) for fairness metrics under different pruning strategies. Significant results ($p < 0.05$) are **bolded.**

Fairness metric	Pruning rate	random oversampling			SMOTE oversampling			holdout oversampling		
		Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
Δ_m	33	2.67 (1.75e-02)	-3.01 (8.84e-03)	1.33 (2.04e-01)	2.03 (6.10e-02)	-3.46 (3.47e-03)	0.73 (4.76e-01)	6.84 (5.59e-06)	-6.72 (6.92e-06)	2.79 (1.38e-02)
	42	-0.64 (5.30e-01)	-0.42 (6.83e-01)	-0.73 (4.76e-01)	-0.34 (7.37e-01)	-1.43 (1.75e-01)	-0.93 (3.68e-01)	4.49 (4.28e-04)	-2.20 (4.41e-02)	1.84 (8.58e-02)
	50	-1.88 (7.93e-02)	-0.56 (5.87e-01)	-2.12 (5.09e-02)	0.66 (5.20e-01)	-1.33 (2.04e-01)	-0.78 (4.49e-01)	8.17 (6.64e-07)	-2.53 (2.30e-02)	2.74 (1.53e-02)
	58	2.08 (5.52e-02)	1.56 (1.39e-01)	2.93 (1.04e-02)	1.59 (1.32e-01)	0.42 (6.78e-01)	2.29 (3.68e-02)	3.60 (2.64e-03)	0.38 (7.06e-01)	5.49 (6.27e-05)
	66	-0.84 (4.12e-01)	-1.69 (1.11e-01)	-2.07 (5.65e-02)	-1.18 (2.55e-01)	-2.57 (2.14e-02)	-1.89 (7.80e-02)	3.63 (2.49e-03)	-3.44 (3.63e-03)	-0.20 (8.47e-01)
	75	0.11 (9.13e-01)	-2.41 (2.92e-02)	-1.79 (9.32e-02)	-0.11 (9.14e-01)	-1.84 (8.64e-02)	-1.25 (2.31e-01)	4.90 (1.92e-04)	-3.06 (7.96e-03)	2.34 (3.37e-02)
Δ_q	33	2.41 (2.93e-02)	-1.42 (1.77e-01)	1.33 (2.05e-01)	1.32 (2.08e-01)	-1.50 (1.54e-01)	0.17 (8.68e-01)	4.55 (3.84e-04)	-1.38 (1.88e-01)	2.72 (1.59e-02)
	42	0.73 (4.78e-01)	-0.84 (4.14e-01)	-0.19 (8.50e-01)	1.03 (3.18e-01)	-1.81 (9.11e-02)	-0.60 (5.59e-01)	5.13 (1.23e-04)	-1.76 (9.95e-02)	2.43 (2.84e-02)
	50	-0.26 (7.95e-01)	-0.74 (4.73e-01)	-0.59 (5.63e-01)	0.40 (6.92e-01)	-1.04 (3.14e-01)	-0.06 (9.54e-01)	5.79 (3.57e-05)	-1.84 (8.53e-02)	2.66 (1.79e-02)
	58	3.20 (5.93e-03)	1.06 (3.06e-01)	2.47 (2.62e-02)	2.57 (2.15e-02)	0.79 (4.44e-01)	2.36 (3.21e-02)	5.34 (8.21e-05)	0.51 (6.16e-01)	5.11 (1.28e-04)
	66	-0.21 (8.33e-01)	-0.74 (4.69e-01)	-1.19 (2.54e-01)	0.35 (7.29e-01)	-1.74 (1.02e-01)	-1.53 (1.48e-01)	2.74 (1.53e-02)	-2.50 (2.44e-02)	0.46 (6.50e-01)
	75	0.73 (4.78e-01)	-3.14 (6.77e-03)	-1.92 (7.42e-02)	0.73 (4.74e-01)	-2.66 (1.77e-02)	-1.33 (2.02e-01)	5.75 (3.81e-05)	-4.90 (1.92e-04)	1.35 (1.97e-01)
Δ_{he}	33	2.32 (3.49e-02)	-4.82 (2.27e-04)	-2.22 (4.19e-02)	1.55 (1.43e-01)	-2.08 (5.50e-02)	-1.09 (2.94e-01)	3.84 (1.61e-03)	-2.61 (1.97e-02)	0.81 (4.31e-01)
	42	1.26 (2.29e-01)	-2.70 (1.65e-02)	-0.34 (7.35e-01)	2.99 (9.23e-03)	-2.88 (1.16e-02)	0.08 (9.37e-01)	4.98 (1.66e-04)	-1.54 (1.45e-01)	2.46 (2.64e-02)
	50	0.15 (8.84e-01)	-2.82 (1.30e-02)	-2.32 (3.50e-02)	0.92 (3.72e-01)	-2.30 (3.62e-02)	-1.40 (1.83e-01)	5.03 (1.49e-04)	-1.67 (1.15e-01)	1.79 (9.41e-02)
	58	3.30 (4.82e-03)	-0.22 (8.26e-01)	2.83 (1.27e-02)	4.13 (8.89e-04)	-1.04 (3.15e-01)	2.35 (3.26e-02)	4.71 (2.79e-04)	-0.70 (4.92e-01)	4.90 (1.93e-04)
	66	0.82 (4.26e-01)	-2.04 (5.95e-02)	-0.99 (3.38e-01)	2.42 (2.88e-02)	-3.68 (2.24e-03)	-1.69 (1.12e-01)	4.47 (4.52e-04)	-3.04 (8.28e-03)	1.08 (2.98e-01)
	75	2.78 (1.39e-02)	-4.13 (8.84e-04)	-1.60 (1.29e-01)	2.83 (1.26e-02)	-2.80 (1.34e-02)	-0.80 (4.38e-01)	7.65 (1.49e-06)	-4.71 (2.77e-04)	1.03 (3.18e-01)
Δ_s	33	2.57 (2.14e-02)	-2.57 (2.15e-02)	0.61 (5.50e-01)	1.22 (2.40e-01)	-1.99 (6.55e-02)	-0.14 (8.91e-01)	5.08 (1.36e-04)	-2.60 (1.99e-02)	2.60 (2.00e-02)
	42	0.50 (6.23e-01)	-1.00 (3.33e-01)	0.02 (9.84e-01)	1.45 (1.67e-01)	-1.89 (7.87e-02)	-0.39 (7.02e-01)	6.26 (1.53e-05)	-1.77 (9.65e-02)	2.54 (2.26e-02)
	50	-0.67 (5.14e-01)	-1.10 (2.87e-01)	-1.21 (2.43e-01)	0.47 (6.44e-01)	-1.59 (1.34e-01)	-0.23 (8.18e-01)	6.09 (2.06e-05)	-2.23 (4.13e-02)	2.92 (1.05e-02)
	58	3.86 (1.55e-03)	1.27 (2.23e-01)	2.98 (9.39e-03)	3.37 (4.20e-03)	0.69 (4.99e-01)	2.83 (1.27e-02)	5.61 (4.94e-05)	0.54 (5.97e-01)	6.38 (1.24e-05)
	66	-0.48 (6.35e-01)	-1.52 (1.50e-01)	-1.43 (1.74e-01)	0.35 (7.30e-01)	-2.70 (1.64e-02)	-1.59 (1.32e-01)	3.47 (3.40e-03)	-3.38 (4.11e-03)	0.54 (5.95e-01)
	75	0.50 (6.22e-01)	-3.34 (4.52e-03)	-1.78 (9.50e-02)	0.44 (6.68e-01)	-2.20 (4.40e-02)	-1.20 (2.50e-01)	5.83 (3.30e-05)	-4.31 (6.18e-04)	1.48 (1.59e-01)
Δ_{MAD}	33	1.46 (1.64e-01)	-1.04 (3.14e-01)	0.94 (3.61e-01)	1.93 (7.27e-02)	-2.49 (2.51e-02)	-1.25 (2.31e-01)	1.45 (1.69e-01)	-0.73 (4.76e-01)	-0.32 (7.53e-01)
	42	2.04 (5.96e-02)	-2.31 (3.59e-02)	-0.60 (5.57e-01)	1.89 (7.76e-02)	-1.83 (8.69e-02)	-2.09 (5.39e-02)	0.95 (3.56e-01)	0.26 (7.95e-01)	1.05 (3.09e-01)
	50	0.27 (7.88e-01)	-2.05 (5.80e-02)	-0.43 (6.72e-01)	1.14 (2.73e-01)	-1.70 (1.10e-01)	0.80 (4.35e-01)	1.77 (9.78e-02)	0.44 (6.70e-01)	1.25 (2.30e-01)
	58	0.43 (6.71e-01)	-0.33 (7.48e-01)	-0.22 (8.26e-01)	-0.43 (6.75e-01)	1.22 (2.40e-01)	-0.07 (9.48e-01)	0.01 (9.88e-01)	1.29 (2.17e-01)	0.23 (8.23e-01)
	66	-0.34 (7.35e-01)	-0.62 (5.46e-01)	0.76 (4.59e-01)	-0.23 (8.18e-01)	-0.73 (4.79e-01)	-0.16 (8.71e-01)	-1.29 (2.18e-01)	0.94 (3.64e-01)	1.34 (2.01e-01)
	75	-1.04 (3.14e-01)	-2.77 (1.43e-02)	-1.50 (1.55e-01)	-0.56 (5.84e-01)	-0.49 (6.34e-01)	0.07 (9.46e-01)	0.86 (4.03e-01)	-0.13 (8.97e-01)	0.18 (8.62e-01)
Δ_{avg}	33	-2.21 (4.31e-02)	-1.92 (7.43e-02)	-2.06 (5.75e-02)	-1.72 (1.06e-01)	-1.45 (1.69e-01)	-1.61 (1.28e-01)	0.63 (5.41e-01)	0.89 (3.88e-01)	0.76 (4.62e-01)
	42	-1.58 (1.35e-01)	-1.49 (1.58e-01)	-1.54 (1.44e-01)	-1.84 (8.54e-02)	-1.74 (1.02e-01)	-1.80 (9.25e-02)	1.04 (3.13e-01)	1.38 (1.87e-01)	1.18 (2.56e-01)
	50	-3.56 (2.88e-03)	-3.65 (2.35e-03)	-3.59 (2.67e-03)	-3.08 (7.62e-03)	-2.93 (1.04e-02)	-3.04 (8.20e-03)	0.64 (5.31e-01)	1.04 (3.13e-01)	0.82 (4.23e-01)
	58	1.50 (1.53e-01)	1.52 (1.48e-01)	1.54 (1.45e-01)	1.29 (2.15e-01)	1.34 (2.01e-01)	1.34 (2.00e-01)	4.47 (4.50e-04)	4.64 (3.18e-04)	4.58 (3.61e-04)
	66	-3.12 (7.03e-03)	-3.07 (7.74e-03)	-3.11 (7.19e-03)	-2.78 (1.39e-02)	-2.64 (1.85e-02)	-2.69 (1.67e-02)	-1.02 (3.24e-01)	-0.63 (5.38e-01)	-0.85 (4.08e-01)
	75	-3.39 (4.08e-03)	-3.22 (5.69e-03)	-3.30 (4.82e-03)	-2.04 (5.90e-02)	-1.95 (6.95e-02)	-1.99 (6.54e-02)	-0.76 (4.59e-01)	-0.22 (8.29e-01)	-0.52 (6.13e-01)

Fairness improvement is not restricted to the most extreme classes. Interestingly, the relative magnitudes of fairness improvements do not directly align with their statistical significance. While hardness-based resampling yields the largest absolute reductions in Δ_m , followed by Δ_q and Δ_{he} , the corresponding t -values show the reverse trend. This discrepancy suggests that improvements in the recall gap across most extreme classes (Δ_m and Δ_q) are more variable across runs, but also more pronounced. Meanwhile, improvements spread across all hard and easy classes (Δ_{he}) are smaller but more consistent. This can be explained by higher robustness of Δ_{he} as a fairness metric, as we discussed in Section 5.3. Paired with the fact that Δ_{MAD} gets reduced on CIFAR-100, this indicates that *the impact of hardness-based resampling is not confined to the most extreme classes but rather spread across the whole hardness spectrum*. We believe that Δ_{MAD} increasing on CIFAR-10 does not contradict these insights but rather is a consequence of the unique hardness spectrum characterizing this dataset. As is visible in Figure 5.1, while on CIFAR-100 recall decreases almost linearly across classes, CIFAR-10 recall follows a logistic-like trend, with only a few low-performing classes and most classes clustered around similar recall values. This means that on CIFAR-10 the effects of hardness-based resampling are more related to the hardest classes, and any changes to the Δ_{MAD} are bound to be insignificant and inconsistent—if most classes already have similar recall, there is little room for median dispersion to change.

Quality of EDM-generated samples is insufficient. Cross-dataset comparisons reveal further nuances. In the first case study, the impact of hardness-based resampling on CIFAR-

Table 5.2: Case Study 1 (CIFAR-100): t -statistics (with p -values) for fairness metrics under different pruning strategies. Significant results ($p < 0.05$) are **bolded.**

Fairness metric	Pruning rate	random oversampling			SMOTE oversampling			holdout oversampling		
		Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
Δ_m	33	3.19 (6.08e-03)	-0.74 (4.72e-01)	2.53 (2.31e-02)	0.98 (3.40e-01)	-3.99 (1.18e-03)	-0.14 (8.91e-01)	5.96 (2.63e-05)	-5.00 (1.59e-04)	4.14 (8.74e-04)
	42	1.96 (6.80e-02)	-2.89 (1.13e-02)	0.84 (4.16e-01)	0.23 (8.25e-01)	-7.96 (9.22e-07)	-2.14 (4.91e-02)	5.43 (7.01e-05)	-3.30 (4.90e-03)	4.23 (7.30e-04)
	50	0.42 (6.79e-01)	-2.42 (2.85e-02)	-0.03 (9.80e-01)	-0.60 (5.59e-01)	-3.80 (1.75e-03)	-2.62 (1.91e-02)	5.54 (5.62e-05)	-4.41 (5.07e-04)	2.78 (1.41e-02)
	58	2.03 (5.99e-02)	-1.05 (3.12e-01)	0.12 (9.08e-01)	1.02 (3.24e-01)	-3.09 (7.48e-03)	-1.29 (2.17e-01)	6.71 (7.03e-06)	-3.21 (5.86e-03)	4.35 (5.70e-04)
	66	6.15 (1.86e-05)	-2.33 (3.39e-02)	1.80 (9.22e-02)	4.55 (3.85e-04)	-4.12 (9.00e-04)	1.64 (1.22e-01)	8.30 (5.50e-07)	-3.93 (1.35e-03)	2.88 (1.16e-02)
Δ_q	75	2.70 (1.64e-02)	-2.68 (1.70e-02)	0.66 (5.18e-01)	1.85 (8.44e-02)	-2.78 (1.41e-02)	0.25 (8.03e-01)	5.72 (4.06e-05)	-2.42 (2.84e-02)	2.99 (9.20e-03)
	33	4.61 (3.40e-04)	-4.20 (7.68e-04)	0.37 (7.19e-01)	2.15 (4.82e-02)	-7.97 (9.03e-07)	-2.11 (5.22e-02)	10.52 (2.57e-08)	-9.13 (1.63e-07)	3.69 (2.19e-03)
	42	5.35 (8.12e-05)	-5.02 (1.51e-04)	-0.36 (7.21e-01)	3.42 (3.83e-03)	-8.61 (3.45e-07)	-2.76 (1.45e-02)	17.26 (2.63e-11)	-4.99 (1.60e-04)	6.05 (2.24e-05)
	50	3.13 (6.87e-03)	-5.27 (9.47e-05)	-0.62 (5.42e-01)	4.43 (4.83e-04)	-7.59 (1.64e-06)	-2.11 (5.22e-02)	14.27 (3.91e-10)	-8.78 (2.67e-07)	7.00 (4.31e-06)
	58	3.52 (3.11e-03)	-3.92 (1.35e-03)	0.10 (9.18e-01)	6.13 (1.92e-05)	-4.53 (4.00e-04)	-1.34 (2.00e-01)	13.05 (1.37e-09)	-5.78 (3.61e-05)	7.97 (9.08e-07)
Δ_{he}	66	8.74 (2.86e-07)	-3.64 (2.40e-03)	3.56 (2.85e-03)	8.78 (2.70e-07)	-3.13 (6.87e-03)	3.08 (7.63e-03)	21.10 (1.45e-12)	-5.67 (4.41e-05)	10.98 (1.44e-08)
	75	6.39 (1.22e-05)	-7.29 (2.67e-06)	1.44 (1.69e-01)	6.01 (2.38e-05)	-5.00 (1.57e-04)	0.59 (5.67e-01)	11.29 (9.85e-09)	-6.33 (1.36e-05)	5.69 (4.31e-05)
	33	7.60 (1.61e-06)	-6.27 (1.51e-05)	0.98 (3.41e-01)	5.84 (3.24e-05)	-7.80 (1.18e-06)	0.61 (5.51e-01)	24.63 (1.52e-13)	-12.67 (2.04e-09)	4.72 (2.75e-04)
	42	9.71 (7.31e-06)	-6.91 (4.96e-06)	1.14 (2.74e-01)	6.23 (1.60e-05)	-9.67 (7.72e-08)	-0.90 (3.82e-01)	22.57 (5.45e-13)	-8.60 (3.51e-07)	5.58 (5.28e-05)
	50	9.49 (9.84e-08)	-4.73 (2.68e-04)	1.79 (9.42e-02)	10.99 (1.43e-08)	-10.77 (1.88e-08)	0.22 (8.29e-01)	29.52 (1.06e-14)	-10.26 (3.58e-08)	7.34 (2.46e-06)
Δ_σ	58	8.69 (3.97e-07)	-5.93 (2.75e-05)	1.51 (1.52e-01)	7.30 (2.62e-06)	-5.44 (3.24e-05)	2.00 (6.40e-02)	13.70 (6.90e-10)	-5.60 (5.06e-05)	5.19 (1.11e-04)
	66	10.68 (2.10e-08)	-6.63 (8.00e-06)	3.09 (7.50e-03)	13.63 (7.45e-10)	-4.47 (4.47e-04)	2.23 (4.17e-02)	20.46 (2.26e-12)	-9.30 (1.28e-07)	5.96 (2.61e-05)
	75	22.05 (7.63e-13)	-8.76 (2.77e-07)	3.98 (1.21e-03)	13.13 (1.26e-09)	-5.43 (6.92e-05)	2.64 (1.85e-02)	26.70 (4.63e-14)	-9.95 (5.32e-08)	6.93 (4.82e-06)
	33	5.19 (1.11e-04)	-5.49 (6.20e-05)	0.59 (5.62e-01)	3.19 (6.12e-03)	-9.40 (1.12e-07)	-1.28 (2.18e-01)	11.86 (5.06e-09)	-10.92 (1.56e-08)	3.70 (2.13e-03)
	42	7.10 (3.61e-06)	-6.82 (5.81e-06)	-0.48 (6.41e-01)	4.42 (5.01e-04)	-11.20 (1.10e-08)	-2.53 (2.30e-02)	19.61 (4.18e-12)	-6.58 (8.70e-06)	5.70 (4.25e-05)
Δ_{MAD}	50	6.53 (9.46e-06)	-6.48 (1.04e-05)	0.04 (9.67e-01)	7.22 (2.98e-06)	-12.37 (2.84e-09)	-2.40 (2.98e-02)	21.86 (8.65e-13)	-13.91 (5.59e-10)	10.53 (2.52e-08)
	58	6.17 (1.79e-05)	-6.04 (2.26e-05)	0.53 (6.01e-01)	6.52 (9.71e-06)	-5.90 (2.92e-05)	-0.25 (8.09e-01)	13.71 (6.86e-10)	-6.66 (7.64e-06)	5.59 (5.20e-05)
	66	14.00 (5.10e-10)	-3.87 (1.50e-03)	4.40 (5.21e-04)	12.79 (1.80e-09)	-3.18 (6.24e-03)	3.01 (8.76e-03)	19.62 (4.17e-12)	-5.44 (6.83e-05)	9.52 (9.50e-08)
	75	12.35 (2.90e-09)	-8.83 (2.50e-07)	0.72 (4.85e-01)	8.53 (3.90e-07)	-5.47 (6.40e-05)	0.89 (3.90e-01)	14.88 (2.18e-10)	-8.00 (8.58e-07)	5.16 (1.17e-04)
	33	3.15 (6.62e-03)	-4.33 (6.00e-04)	0.49 (6.28e-01)	2.09 (5.45e-02)	-4.49 (4.29e-04)	1.39 (1.85e-01)	5.26 (9.66e-05)	-3.77 (1.85e-03)	1.19 (2.52e-01)
Δ_{avg}	42	2.05 (5.83e-02)	-8.22 (6.20e-07)	0.23 (8.25e-01)	2.22 (4.25e-02)	-4.04 (1.07e-03)	-1.12 (2.79e-01)	6.34 (1.32e-05)	-4.25 (7.01e-04)	-0.31 (7.59e-01)
	50	2.26 (3.90e-02)	-1.92 (7.43e-02)	0.46 (6.52e-01)	2.33 (3.40e-02)	-3.54 (2.95e-03)	0.70 (4.97e-01)	5.22 (1.03e-04)	-3.65 (2.39e-03)	2.40 (2.99e-02)
	58	2.74 (1.51e-02)	-2.86 (1.20e-02)	-1.21 (2.44e-01)	3.39 (4.03e-03)	-3.43 (3.74e-03)	1.07 (3.03e-01)	4.10 (9.53e-04)	-3.79 (1.79e-03)	0.40 (6.93e-01)
	66	7.10 (3.61e-06)	-0.89 (3.90e-01)	0.95 (3.59e-01)	5.23 (1.02e-04)	-0.73 (4.74e-01)	0.70 (4.93e-01)	5.41 (7.22e-05)	0.03 (9.74e-01)	2.09 (5.38e-02)
	75	2.28 (3.73e-02)	-4.04 (1.07e-03)	-0.54 (5.96e-01)	3.97 (1.24e-03)	-5.39 (7.55e-05)	0.55 (5.92e-01)	6.08 (2.12e-05)	-5.17 (1.15e-04)	-0.37 (7.18e-01)
Δ_{avg}	33	-10.90 (1.59e-08)	-8.01 (8.50e-07)	-9.60 (8.57e-08)	-10.68 (2.10e-08)	-7.05 (3.95e-06)	-8.91 (2.25e-07)	-2.71 (1.60e-02)	4.78 (2.45e-04)	0.31 (7.58e-01)
	42	-12.24 (3.28e-09)	-10.54 (2.49e-08)	-11.16 (1.16e-08)	-12.46 (2.58e-09)	-9.96 (5.27e-08)	-11.08 (1.27e-08)	-2.68 (1.72e-02)	4.80 (2.33e-04)	0.25 (8.06e-01)
	50	-8.64 (3.31e-07)	-6.51 (9.92e-06)	-7.51 (1.87e-06)	-16.14 (6.85e-11)	-10.93 (1.54e-08)	-13.46 (8.90e-10)	-3.03 (8.52e-03)	3.73 (2.01e-03)	-0.10 (9.19e-01)
	58	-8.57 (3.64e-07)	-6.50 (1.00e-05)	-7.84 (1.11e-06)	-8.88 (2.33e-07)	-7.23 (2.91e-06)	-8.20 (6.39e-07)	-3.25 (5.34e-03)	1.13 (2.74e-01)	-1.30 (2.13e-01)
	66	6.67 (7.53e-06)	9.91 (5.59e-08)	8.93 (2.18e-07)	5.48 (6.32e-05)	7.71 (1.35e-06)	6.87 (5.29e-06)	15.02 (1.91e-10)	21.12 (1.44e-12)	17.62 (1.97e-11)
	75	-9.15 (1.58e-07)	-7.00 (4.27e-06)	-8.08 (7.63e-07)	-12.47 (2.56e-09)	-8.14 (6.96e-07)	-10.56 (2.42e-08)	-3.36 (4.25e-03)	2.31 (3.58e-02)	-0.98 (3.45e-01)

100 is nearly five times larger than on CIFAR-10, whereas in the second case study the improvements are comparable across datasets. We analyze this further and notice that the changes coming from hardness-based resampling appear similar on CIFAR-10 for both case studies (≈ 0.01 on Δ_m and Δ_q , and ≈ 0.005 on Δ_{he} and Δ_σ), but on CIFAR-100 the gains diminish by roughly $0.01 - 0.05$ depending on the fairness metric, when considering the results for $\alpha = 1$ on the second case study as we used the same α in the first case study. Considering the differences in experimental setup between the two case studies, two explanations arise. Either the quality of EDM-generated samples was insufficient to match the improvements achieved with the holdout set, or the impact of hardness-based resampling diminishes as dataset size increases.

While we do notice the results following a slow plateauing trend on some fairness metrics around the pruning rate of 33% we believe that the former hypothesis is more likely. This is because we see that for pruning rate of 33% Δ_{he} is lower when holdout set is involved (first case study) than on the full dataset when using EDM-generated data (second case study). Considering the observed monotonic decrease of Δ_{he} with the reduction of the pruning rate in the first case study, the evidence strongly suggests that the observed difference primarily stems from the quality of data used for oversampling. Further experiments at lower pruning rates could provide clearer insights, but were not feasible because the holdout set lacked sufficient samples for oversampling in accordance with the computed resampling ratios.

More pronounced data-imbalance leads to higher fairness gains. Beyond dataset-specific variations, we also analyze how the severity of the induced imbalance (controlled

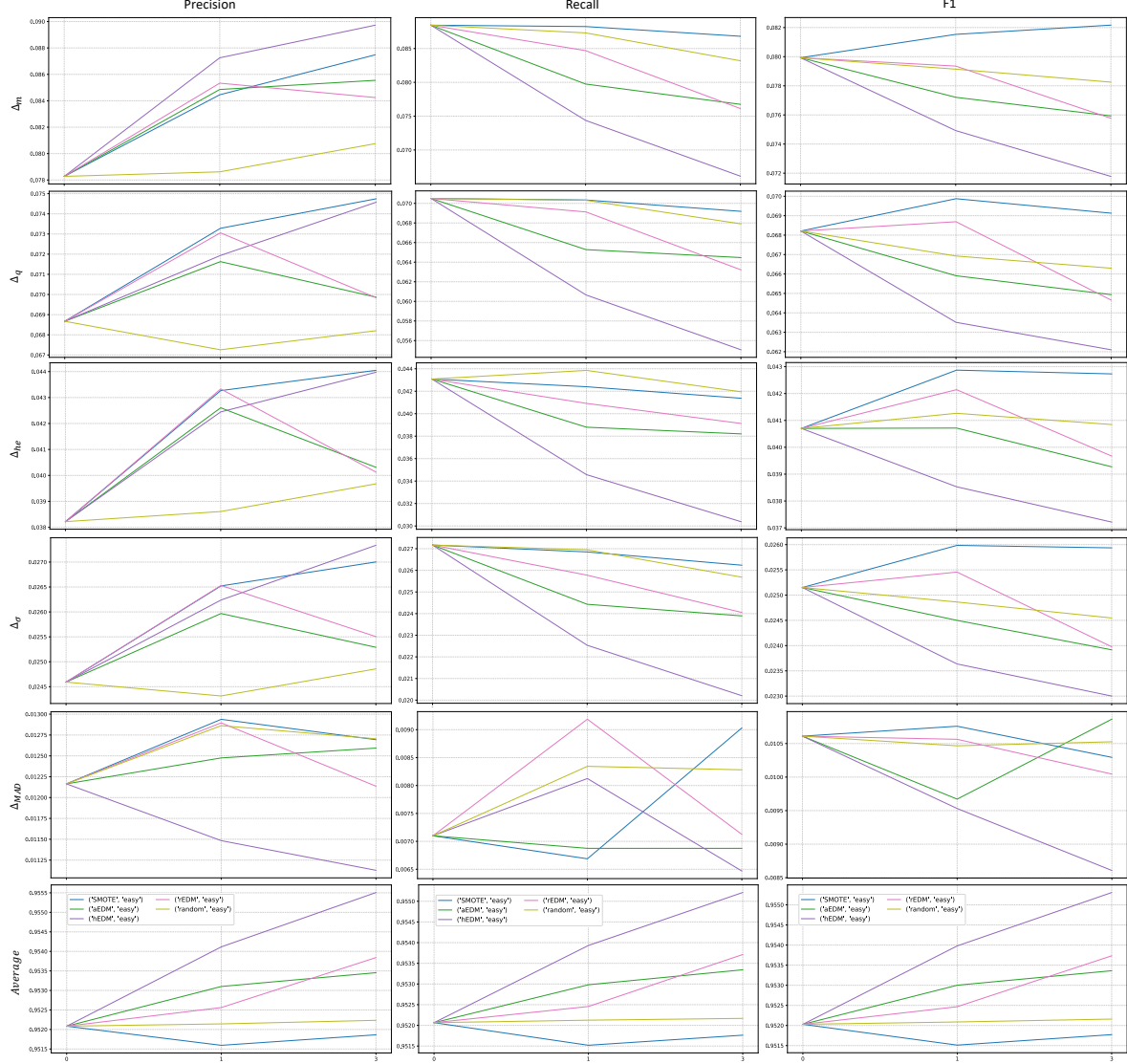


Figure 5.4: Case Study 2 (CIFAR-10): We notice that, in most cases, the higher the degree of the introduced data imbalance (x -axis) the larger the effect on fairness when high quality synthetic samples are available. We expect the existence of some threshold α beyond which the drawback from too heavy data imbalance starts to overwhelm the benefits of addressing hardness imbalance.

by α) modulates these trends. Consistent with expectations, we observe that in most cases greater imbalance yields stronger and more statistically significant gains from hardness-based resampling. Although our experiments do not explicitly identify a turning point, we hypothesize the existence of a threshold α_t beyond which the harm from data imbalance outweighs the benefits of mitigating hardness imbalance. Designing an algorithm to efficiently estimate this threshold remains an open research direction.

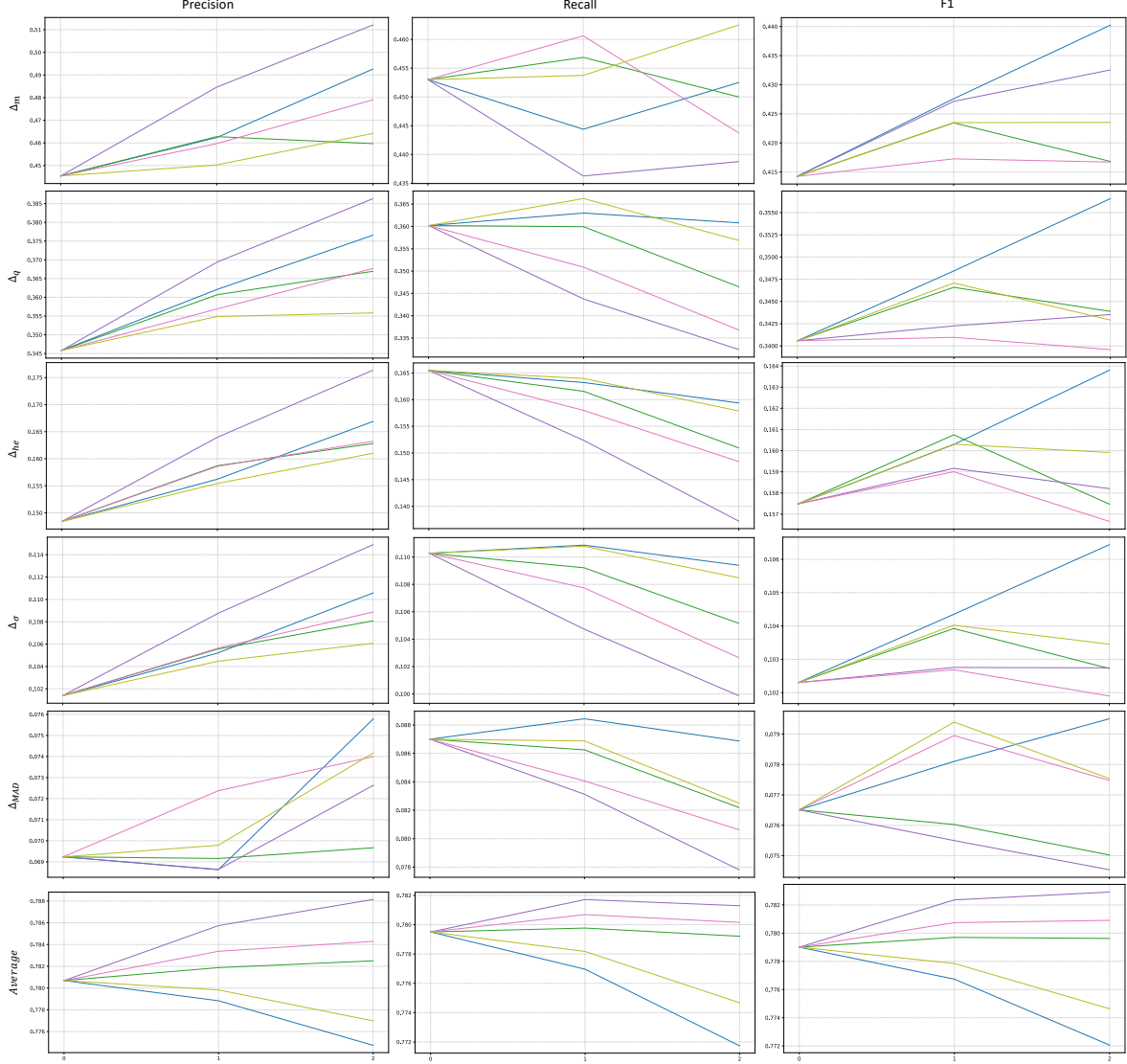


Figure 5.5: Case Study 2 (CIFAR-100): The effect of hardness-based imbalance is similar on both datasets here, unlike the fivefold more pronounced effect seen on CIFAR-100 in the first case study. We attribute this to the subpar quality of EDM-generated samples compared to the real samples from the holdout set. This is evidenced by the discrepancy between the forecasted fairness metrics from the first study and the higher values actually observed in this figure (e.g., Δ_{he} was forecasted to drop below 0.14 for $\alpha = 1$).

Applying only oversampling is problematic. Our ablation study demonstrates that combining over- and undersampling produces more stable fairness improvements than oversampling alone. Oversampling is applied to hard classes to alleviate their underperformance relative to easy ones. However, relying solely on oversampling would increase dataset size and conflate fairness improvements with gains from additional data, rather than redistributing the existing data more effectively. We therefore also applied undersampling to easy classes, both

Table 5.3: Case Study 2 (CIFAR-10): t -statistics (with p -values) for fairness metrics under different oversampling strategies. Significant results ($p < 0.05$) are **bolded.**

Metric	Strategy (oversampling, undersampling)	Δ_m		Δ_v		Δ_{hc}		Δ_σ		Δ_{MAD}		Average	
		$\alpha = 1$	$\alpha = 3$	$\alpha = 1$	$\alpha = 3$	$\alpha = 1$	$\alpha = 3$	$\alpha = 1$	$\alpha = 3$	$\alpha = 1$	$\alpha = 3$	$\alpha = 1$	$\alpha = 3$
Recall	(random, none)	-0.58 (0.57)	0.94 (0.36)	-0.06 (0.95)	-0.87 (0.40)	0.32 (0.75)	-0.72 (0.48)	0.26 (0.80)	-0.39 (0.70)	-2.02 (0.06)	-2.63 (0.02)	0.83 (0.42)	-3.38 (0.00)
	(SMOTE, none)	1.85 (0.08)	-2.20 (0.04)	1.85 (0.08)	-1.42 (0.18)	2.11 (0.05)	-0.34 (0.74)	1.90 (0.08)	-0.90 (0.38)	-0.55 (0.59)	-2.41 (0.03)	0.64 (0.53)	-6.89 (0.00)
	(rEDM, none)	0.71 (0.49)	0.28 (0.78)	0.55 (0.59)	-0.13 (0.90)	0.82 (0.42)	1.52 (0.15)	0.80 (0.44)	0.83 (0.42)	0.00 (1.00)	-3.28 (0.01)	1.07 (0.30)	-3.21 (0.01)
	(aEDM, none)	2.07 (0.06)	1.01 (0.33)	1.63 (0.12)	0.42 (0.68)	2.71 (0.02)	2.48 (0.03)	2.45 (0.03)	2.17 (0.05)	-0.24 (0.81)	-3.24 (0.01)	2.20 (0.04)	-2.89 (0.01)
	(hEDM, none)	5.88 (0.00)	6.19 (0.00)	5.37 (0.00)	6.80 (0.00)	4.86 (0.00)	8.09 (0.00)	7.10 (0.00)	8.20 (0.00)	-0.85 (0.41)	-3.07 (0.01)	3.31 (0.00)	1.98 (0.07)
	(random, easy)	0.14 (0.89)	1.42 (0.18)	0.24 (0.81)	1.41 (0.18)	-0.40 (0.69)	1.00 (0.34)	0.32 (0.75)	1.88 (0.08)	-1.35 (0.20)	-1.02 (0.33)	0.12 (0.90)	0.21 (0.84)
	(SMOTE, easy)	-0.24 (0.81)	0.32 (0.75)	0.20 (0.84)	0.74 (0.47)	0.62 (0.54)	1.28 (0.22)	0.41 (0.69)	1.27 (0.22)	-0.66 (0.52)	-2.67 (0.02)	-1.22 (0.24)	-0.42 (0.68)
	(rEDM, easy)	1.17 (0.26)	4.22 (0.00)	1.09 (0.29)	4.42 (0.00)	2.07 (0.06)	3.40 (0.00)	2.30 (0.04)	4.94 (0.00)	-3.10 (0.01)	0.11 (0.92)	0.73 (0.47)	3.24 (0.01)
	(aEDM, easy)	2.73 (0.02)	3.34 (0.00)	2.59 (0.02)	3.70 (0.00)	3.92 (0.00)	4.71 (0.00)	4.05 (0.00)	4.81 (0.00)	0.34 (0.74)	0.49 (0.63)	1.78 (0.09)	3.07 (0.01)
	(hEDM, easy)	4.83 (0.00)	7.59 (0.00)	4.84 (0.00)	8.80 (0.00)	7.29 (0.00)	14.85 (0.00)	7.27 (0.00)	12.20 (0.00)	-0.95 (0.36)	0.87 (0.40)	2.90 (0.01)	4.71 (0.00)
Precision	(random, none)	-0.14 (0.89)	-1.00 (0.33)	0.10 (0.92)	0.41 (0.69)	-0.83 (0.42)	-0.69 (0.50)	0.21 (0.84)	-0.33 (0.75)	-0.75 (0.47)	-2.09 (0.05)	0.86 (0.40)	-3.32 (0.00)
	(SMOTE, none)	-0.65 (0.52)	-1.67 (0.12)	-0.72 (0.48)	-1.51 (0.15)	-1.65 (0.12)	-3.03 (0.01)	-0.77 (0.45)	-2.08 (0.06)	-0.71 (0.49)	-2.09 (0.05)	0.79 (0.44)	-6.75 (0.00)
	(rEDM, none)	-0.14 (0.89)	-0.95 (0.36)	0.09 (0.93)	0.27 (0.79)	-0.83 (0.42)	-0.80 (0.43)	-0.17 (0.86)	-0.62 (0.54)	-1.27 (0.22)	-1.37 (0.19)	1.11 (0.29)	-3.05 (0.01)
	(aEDM, none)	-1.69 (0.11)	-1.24 (0.23)	0.15 (0.88)	0.17 (0.87)	-1.35 (0.20)	-1.49 (0.16)	-0.51 (0.62)	-0.67 (0.51)	-0.89 (0.39)	-1.11 (0.29)	2.25 (0.04)	-2.76 (0.01)
	(hEDM, none)	-2.14 (0.05)	-1.44 (0.17)	-0.18 (0.86)	-0.76 (0.46)	-1.98 (0.07)	-2.84 (0.01)	-0.84 (0.41)	-1.34 (0.20)	1.38 (0.19)	-1.66 (0.12)	3.50 (0.00)	2.57 (0.02)
	(random, easy)	0.55 (0.59)	-0.18 (0.86)	2.13 (0.05)	1.14 (0.27)	0.17 (0.87)	-0.68 (0.51)	1.50 (0.16)	0.34 (0.74)	-1.03 (0.32)	-0.72 (0.48)	0.10 (0.92)	0.28 (0.78)
	(SMOTE, easy)	-1.22 (0.24)	-3.22 (0.01)	-1.74 (0.10)	-2.33 (0.03)	-2.96 (0.01)	-3.69 (0.00)	-2.03 (0.06)	-2.64 (0.02)	-1.10 (0.29)	-0.63 (0.54)	-1.12 (0.28)	-0.33 (0.75)
	(rEDM, easy)	-1.59 (0.13)	-1.00 (0.33)	-1.24 (0.23)	0.20 (0.85)	-2.46 (0.03)	-1.06 (0.31)	-1.64 (0.12)	-0.48 (0.64)	-0.90 (0.38)	-0.09 (0.93)	0.87 (0.40)	3.39 (0.00)
	(aEDM, easy)	-1.49 (0.16)	-1.66 (0.12)	-0.70 (0.49)	0.19 (0.85)	-2.93 (0.01)	-1.21 (0.24)	-1.20 (0.25)	-0.26 (0.79)	-0.56 (0.59)	-0.52 (0.61)	1.95 (0.07)	3.20 (0.01)
	(hEDM, easy)	-2.65 (0.02)	-2.63 (0.02)	-1.13 (0.28)	-1.72 (0.11)	-2.45 (0.03)	-2.73 (0.02)	-1.60 (0.13)	-2.22 (0.04)	0.58 (0.57)	1.39 (0.19)	3.12 (0.01)	5.15 (0.00)
F1	(random, none)	0.78 (0.45)	0.13 (0.90)	1.06 (0.30)	0.60 (0.56)	-0.27 (0.79)	-1.30 (0.21)	0.71 (0.49)	0.19 (0.85)	0.25 (0.81)	0.35 (0.73)	0.83 (0.42)	-3.37 (0.00)
	(SMOTE, none)	1.89 (0.08)	-2.84 (0.01)	1.12 (0.28)	-1.41 (0.18)	-0.15 (0.88)	-2.55 (0.02)	0.89 (0.39)	-2.07 (0.06)	0.38 (0.71)	0.86 (0.40)	0.71 (0.49)	-6.84 (0.00)
	(rEDM, none)	1.23 (0.24)	1.41 (0.18)	0.99 (0.34)	1.84 (0.09)	-0.27 (0.79)	0.34 (0.74)	0.50 (0.62)	1.31 (0.21)	2.49 (0.02)	1.13 (0.27)	1.08 (0.30)	-3.16 (0.01)
	(aEDM, none)	1.33 (0.20)	1.17 (0.26)	2.03 (0.06)	2.10 (0.05)	1.48 (0.16)	0.03 (0.98)	1.71 (0.11)	1.35 (0.20)	2.11 (0.05)	1.13 (0.28)	2.23 (0.04)	-2.85 (0.01)
	(hEDM, none)	5.10 (0.00)	5.04 (0.00)	4.54 (0.00)	5.43 (0.00)	3.11 (0.01)	3.66 (0.00)	4.27 (0.00)	4.98 (0.00)	0.98 (0.34)	2.10 (0.05)	3.39 (0.00)	2.21 (0.04)
	(random, easy)	0.99 (0.34)	1.41 (0.18)	1.98 (0.07)	1.98 (0.07)	-0.25 (0.80)	0.30 (0.77)	1.30 (0.21)	1.70 (0.11)	0.71 (0.49)	0.56 (0.59)	0.11 (0.91)	0.25 (0.81)
	(SMOTE, easy)	-0.59 (0.57)	-0.98 (0.34)	-0.72 (0.48)	-0.02 (0.99)	-2.73 (0.02)	-1.83 (0.09)	-1.44 (0.17)	-0.95 (0.36)	0.24 (0.81)	1.09 (0.29)	-1.18 (0.26)	-0.37 (0.72)
	(rEDM, easy)	1.09 (0.29)	3.40 (0.00)	0.31 (0.76)	3.57 (0.00)	-1.22 (0.24)	1.94 (0.07)	-0.10 (0.92)	3.26 (0.01)	0.56 (0.59)	1.44 (0.17)	0.81 (0.43)	3.31 (0.00)
	(aEDM, easy)	2.08 (0.06)	3.47 (0.00)	2.04 (0.06)	4.68 (0.00)	0.53 (0.60)	2.66 (0.02)	1.73 (0.10)	4.45 (0.00)	2.25 (0.04)	0.03 (0.97)	1.87 (0.08)	3.16 (0.01)
	(hEDM, easy)	3.26 (0.01)	4.25 (0.00)	3.94 (0.00)	3.94 (0.00)	3.34 (0.00)	4.24 (0.00)	3.42 (0.00)	3.86 (0.00)	1.67 (0.12)	4.67 (0.00)	3.02 (0.01)	4.93 (0.00)

to preserve overall dataset size and to moderate their performance advantage, resulting in a more equitable class distribution. Intuitively, easy classes exceed their sample complexity and can therefore tolerate pruning, whereas hard classes fall below it and benefit from additional samples. Empirically, we observe that using only oversampling leads to more consistent gains in average recall but less consistent changes in fairness, while the combined approach yields less consistent increases in overall performance but noticeably more consistent improvements in fairness (see Tables 5.3, 5.4). This trade-off supports the view that fairness and performance need not move in tandem, and that balanced gains require both mechanisms working together.

Hardness-based resampling worsens precision. Hardness-based resampling proves effective for improving recall-based fairness, however, it introduces its own trade-offs. When the same fairness metrics are computed using precision instead of recall, we observe the opposite trend. As shown in Appendix Figures A.1-A.8, oversampling hard classes increases the number of positive predictions on those classes, leading to higher counts of both true positives (TP) and false positives (FP), while false negatives (FN) and true negatives (TN) decrease. The simultaneous rise in TP and FP explains the observed increase in recall and the corresponding drop in precision on hard classes. Changes on easy classes have the opposite effects but remain comparatively minor. Consequently, the overall change in fairness arises predominantly from the oversampling of hard classes rather than from undersampling effects, although both contribute to the observed dynamics discussed earlier. This behavior reflects the well-known precision–recall trade-off—desirable when fairness in recall is prioritized, but potentially detrimental in precision-critical settings.

Inconsistent changes to F1. A natural follow-up question is how the observed improvements in recall compare to the accompanying reductions in precision. To examine this, we

Table 5.4: Case Study 2 (CIFAR-100): t -statistics (with p -values) for fairness metrics under different oversampling strategies. Significant results ($p < 0.05$) are **bolded.**

Metric	Strategy (oversampling, undersampling)	Δ_m		Δ_q		Δ_{he}		Δ_σ		Δ_{MAD}		Average	
		$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$
Recall	(random, none)	-1.36 (0.19)	0.13 (0.90)	-1.86 (0.08)	-0.38 (0.71)	-0.69 (0.50)	1.05 (0.31)	-2.03 (0.06)	0.35 (0.73)	-1.10 (0.29)	0.52 (0.61)	0.02 (0.99)	-1.70 (0.11)
	(SMOTE, none)	2.00 (0.06)	0.29 (0.77)	-1.45 (0.17)	-0.10 (0.92)	-0.13 (0.90)	1.15 (0.27)	-0.56 (0.59)	-0.20 (0.84)	0.88 (0.39)	0.49 (0.63)	-2.13 (0.05)	-4.09 (0.00)
	(rEDM, none)	-0.09 (0.93)	-0.00 (1.00)	1.81 (0.09)	3.85 (0.00)	3.19 (0.01)	5.13 (0.00)	1.79 (0.09)	4.24 (0.00)	-0.37 (0.72)	2.33 (0.03)	3.10 (0.01)	4.04 (0.00)
	(aEDM, none)	-0.53 (0.61)	0.60 (0.56)	0.79 (0.44)	1.93 (0.07)	0.54 (0.60)	3.44 (0.00)	1.12 (0.28)	3.26 (0.01)	0.00 (1.00)	1.52 (0.15)	3.92 (0.00)	4.28 (0.00)
	(hEDM, none)	2.29 (0.04)	0.64 (0.53)	5.24 (0.00)	4.68 (0.00)	6.58 (0.00)	11.74 (0.00)	7.35 (0.00)	7.47 (0.00)	2.91 (0.01)	2.45 (0.03)	6.77 (0.00)	5.83 (0.00)
	(random, easy)	0.45 (0.66)	-0.59 (0.56)	-1.54 (0.14)	1.06 (0.31)	0.72 (0.48)	4.27 (0.00)	-0.62 (0.54)	2.70 (0.02)	0.00 (1.00)	3.05 (0.01)	-1.55 (0.14)	-4.31 (0.00)
	(SMOTE, easy)	1.67 (0.12)	0.51 (0.62)	-0.65 (0.53)	-0.20 (0.85)	1.67 (0.11)	3.10 (0.01)	-0.82 (0.42)	0.84 (0.42)	-0.77 (0.45)	0.00 (1.00)	-2.59 (0.02)	-6.06 (0.00)
	(rEDM, easy)	-0.41 (0.69)	2.02 (0.06)	3.04 (0.01)	6.73 (0.00)	4.44 (0.00)	10.64 (0.00)	3.44 (0.00)	9.67 (0.00)	1.45 (0.17)	4.44 (0.00)	1.37 (0.19)	0.72 (0.48)
	(aEDM, easy)	0.00 (1.00)	0.97 (0.35)	-0.00 (1.00)	3.84 (0.00)	2.11 (0.05)	8.10 (0.00)	1.09 (0.30)	7.16 (0.00)	0.38 (0.71)	2.53 (0.02)	0.43 (0.67)	-0.27 (0.79)
	(hEDM, easy)	2.47 (0.03)	2.65 (0.02)	4.99 (0.00)	8.01 (0.00)	8.10 (0.00)	15.82 (0.00)	7.93 (0.00)	11.35 (0.00)	2.09 (0.05)	5.46 (0.00)	2.16 (0.05)	2.28 (0.04)
Precision	(random, none)	0.41 (0.69)	0.23 (0.82)	-0.57 (0.58)	-0.90 (0.38)	-0.72 (0.48)	-0.42 (0.68)	-0.91 (0.38)	-0.64 (0.53)	1.05 (0.31)	1.98 (0.07)	-0.21 (0.84)	-1.46 (0.16)
	(SMOTE, none)	-0.39 (0.70)	-1.63 (0.12)	-1.31 (0.21)	-2.96 (0.01)	-1.50 (0.15)	-2.19 (0.04)	-1.29 (0.22)	-3.10 (0.01)	0.83 (0.42)	0.12 (0.91)	-2.07 (0.06)	-2.88 (0.01)
	(rEDM, none)	-1.23 (0.24)	-2.17 (0.05)	-0.51 (0.62)	-2.11 (0.05)	-0.71 (0.49)	-2.55 (0.02)	-0.88 (0.39)	-2.60 (0.02)	0.39 (0.70)	1.22 (0.24)	3.76 (0.00)	6.02 (0.00)
	(aEDM, none)	0.69 (0.50)	-1.10 (0.29)	-0.18 (0.86)	-2.19 (0.04)	-0.65 (0.52)	-2.71 (0.02)	-0.23 (0.82)	-2.29 (0.04)	0.93 (0.37)	0.39 (0.70)	4.42 (0.00)	5.83 (0.00)
	(hEDM, none)	-2.88 (0.01)	-6.19 (0.00)	-4.13 (0.00)	-8.78 (0.00)	-2.98 (0.01)	-5.75 (0.00)	-3.98 (0.00)	-9.72 (0.00)	0.67 (0.51)	1.14 (0.27)	10.41 (0.00)	10.96 (0.00)
	(random, easy)	-0.12 (0.91)	-1.82 (0.09)	-1.55 (0.14)	-2.14 (0.05)	-1.78 (0.10)	-3.26 (0.01)	-1.83 (0.09)	-2.88 (0.01)	-0.22 (0.83)	-2.25 (0.04)	-1.01 (0.33)	-3.67 (0.00)
	(SMOTE, easy)	-1.36 (0.19)	-3.94 (0.00)	-3.19 (0.01)	-5.26 (0.00)	-3.20 (0.01)	-8.09 (0.00)	-2.73 (0.02)	-6.64 (0.00)	0.40 (0.69)	-3.98 (0.00)	-1.99 (0.06)	-4.85 (0.00)
	(rEDM, easy)	-1.06 (0.31)	-3.35 (0.00)	-2.70 (0.02)	-7.70 (0.00)	-5.04 (0.00)	-4.94 (0.00)	-3.98 (0.00)	-7.03 (0.00)	-1.37 (0.19)	-1.86 (0.08)	2.93 (0.01)	3.34 (0.00)
	(aEDM, easy)	-1.47 (0.16)	-1.45 (0.17)	-2.67 (0.02)	-4.71 (0.00)	-3.43 (0.00)	-5.46 (0.00)	-2.64 (0.02)	-5.22 (0.00)	0.14 (0.89)	-0.16 (0.87)	1.35 (0.20)	2.58 (0.02)
	(hEDM, easy)	-3.60 (0.00)	-6.21 (0.00)	-5.17 (0.00)	-9.23 (0.00)	-6.69 (0.00)	-10.91 (0.00)	-6.17 (0.00)	-10.94 (0.00)	0.50 (0.63)	-2.08 (0.05)	5.01 (0.00)	8.93 (0.00)
F1	(random, none)	-1.11 (0.28)	-0.31 (0.76)	-1.04 (0.31)	-0.17 (0.87)	-1.20 (0.25)	0.09 (0.93)	-2.31 (0.04)	-0.31 (0.76)	-1.79 (0.09)	0.43 (0.67)	-0.10 (0.92)	-1.62 (0.13)
	(SMOTE, none)	-1.00 (0.34)	-1.64 (0.12)	-1.27 (0.22)	-1.54 (0.15)	-0.98 (0.34)	-1.18 (0.26)	-1.00 (0.34)	-2.10 (0.05)	-0.17 (0.87)	-1.16 (0.26)	-2.12 (0.05)	-3.57 (0.00)
	(rEDM, none)	-0.86 (0.40)	0.69 (0.50)	1.13 (0.28)	1.48 (0.16)	0.95 (0.36)	-0.02 (0.99)	0.69 (0.50)	1.15 (0.27)	0.14 (0.89)	0.29 (0.78)	3.34 (0.00)	4.71 (0.00)
	(aEDM, none)	-0.93 (0.37)	-0.69 (0.50)	0.64 (0.53)	0.16 (0.87)	-0.15 (0.88)	-0.36 (0.73)	0.52 (0.61)	0.05 (0.96)	-0.40 (0.69)	-0.29 (0.78)	4.19 (0.00)	4.93 (0.00)
	(hEDM, none)	-0.68 (0.51)	-2.05 (0.06)	0.09 (0.93)	-0.58 (0.57)	0.78 (0.45)	0.29 (0.77)	1.00 (0.33)	-0.60 (0.56)	1.00 (0.33)	-0.05 (0.96)	8.35 (0.00)	7.63 (0.00)
	(random, easy)	-1.14 (0.27)	-1.03 (0.32)	-1.28 (0.22)	-0.37 (0.72)	-1.03 (0.32)	-0.96 (0.35)	-1.50 (0.16)	-1.13 (0.28)	-1.76 (0.10)	-0.84 (0.41)	-1.37 (0.19)	-4.10 (0.00)
	(SMOTE, easy)	-2.01 (0.06)	-3.04 (0.01)	-1.53 (0.15)	-3.28 (0.01)	-1.75 (0.10)	-3.38 (0.00)	-2.03 (0.06)	-3.94 (0.00)	-1.18 (0.26)	-2.18 (0.05)	-2.39 (0.03)	-5.53 (0.00)
	(rEDM, easy)	-0.25 (0.80)	-0.21 (0.84)	0.35 (0.73)	0.84 (0.41)	-0.66 (0.52)	0.93 (0.37)	-0.24 (0.81)	0.95 (0.36)	-1.62 (0.13)	-0.72 (0.48)	1.93 (0.07)	1.86 (0.08)
	(aEDM, easy)	-1.16 (0.26)	-0.19 (0.85)	-1.24 (0.23)	-0.59 (0.57)	-1.34 (0.20)	0.49 (0.63)	-1.31 (0.21)	-0.25 (0.80)	0.05 (0.96)	0.78 (0.45)	0.89 (0.39)	0.91 (0.38)
	(hEDM, easy)	-1.68 (0.11)	-2.93 (0.01)	-0.11 (0.91)	-0.44 (0.67)	-0.66 (0.52)	-0.02 (0.98)	-0.29 (0.77)	-0.23 (0.82)	0.39 (0.70)	0.91 (0.38)	3.31 (0.00)	4.83 (0.00)

recompute the same fairness metrics using F1 scores and find that, across most settings, the F1 gap between classes decreases following hardness-based resampling. The changes are mostly consistent and significant for CIFAR-10 (e.g., Δ_{he} reduction of about 0.001-0.003), as well as for CIFAR-100 in the first case study (improvement of roughly 0.005-0.015 for Δ_{he}). The only clear exception occurs on CIFAR-100 in the second case study, where we observe a slight degradation in F1 (around 0.001-0.002 for Δ_{he}) that is not statistically significant. We attribute this discrepancy primarily to the interplay between recall gains and precision losses. Across both case studies, precision declines appear less sensitive to the quality of synthetic samples used for oversampling than recall improvements. Consequently, when synthetic data are of limited quality—as in the CIFAR-100 EDM set used in Case Study 2—the imbalance-induced precision loss can outweigh the recall gain, producing an overall reduction in F1. In contrast, when synthetic data better preserve the geometry and topology of class manifolds (as in Case Study 1) or when the number of samples per class is sufficiently large to mitigate the adverse effects of undersampling (as in CIFAR-10 in Case Study 2), the recall improvement dominates and F1 increases accordingly. This pattern supports the interpretation that F1 changes depend jointly on the fidelity of oversampled data and the dimensional complexity of the underlying dataset, rather than reflecting a fundamental limitation of hardness-based resampling itself.

Changes to macro-recall depend on hardness estimates and data quality. As we observed before, oversampling high-quality hard samples can raise overall recall, while undersampling easy classes tends to reduce it; hence, improvements to dataset-level averages occur only when the former outweighs the latter. This balance, in turn, hinges on both the reliability of the hardness estimator and the quality of the data used for oversampling. The first dependency is evident in Case Study 2, where improvements in the quality of synthetic samples led to corresponding gains in macro-recall. In contrast, the drops observed

in Case Study 1 diminished under similar conditions. These trends are even clearer when analyzing the t-test results. However, when comparing the two case studies directly, a more nuanced pattern emerges. In Case Study 1, macro-recall on CIFAR-10 fluctuates around baseline with no statistically significant change, while CIFAR-100 shows a modest decline. In contrast, Case Study 2 consistently improves recall on both datasets. At first glance, these cross-case differences may seem inconsistent, but they are not: while the earlier analysis demonstrates sensitivity to oversample quality, this direct comparison reveals an additional dependence on the accuracy of the hardness estimates. As discussed in Section 5.4.1, the used hardness estimates were less reliable in Case Study 1 due to sampling-induced hardness shifts, explaining the observed degradation despite access to real data. Together, these results suggest that improvements to macro-level metrics depend jointly on the fidelity of the hardness estimates and the quality of the data used for oversampling, with neither factor alone sufficient to guarantee gains.

5.6 Conclusion

This chapter examined how hardness-based resampling influences class-level fairness and overall performance across two complementary case studies. Through systematic experiments and statistical analysis, we identified consistent behavioral patterns and underlying mechanisms that explain when and why resampling improves or degrades fairness. The key empirical insights are summarized below.

- The effect of hardness-based resampling is not confined to the most extreme classes but is distributed across the entire class spectrum, with the largest gains observed for the hardest classes.
- The mechanism underlying these improvements involves increasing recall on hard classes via oversampling while slightly reducing recall on easy classes via undersampling.
- The resulting changes in macro-recall and recall-based fairness depend on three main factors: the quality of the oversampled data, the accuracy of hardness estimates used for pruning, and the degree of the induced data imbalance (α).
- Hardness-based resampling tends to widen precision gaps across classes. These changes appear primarily driven by the induced data imbalance rather than the quality of the oversampled data. Consequently, whether F1 gaps decrease depends on the changes in recall.

Key modulators. The magnitude and direction of the effects described above are controlled by three interrelated factors:

- **Quality of oversampling data.** Empirically, higher-fidelity synthetic samples increase recall on hard classes: the better the realism and diversity of the generated images, the larger the recall gains on those classes. However, our results also highlight that current diffusion models still fall short of producing data matching real-sample quality, indicating a need for further advances in generative data fidelity and diversity.

- **Accuracy of hardness estimates.** Our experiments reveal estimator accuracy to be critical when guiding pruning: better estimates lead to smaller drops in recall on easy classes after pruning because they more reliably identify genuinely easy samples to remove. This underscores the importance of developing more precise and reliable hardness estimators.
- **Degree of induced data imbalance (α).** The magnitude of the resampling-induced imbalance scales the effect size across all metrics. Larger α amplifies changes in recall, precision, and F1 (both positive and negative), increasing both potential gains and potential harms. Determining the empirical threshold at which benefits from addressing hardness imbalance turn into drawbacks from introducing too large data imbalance, and formalizing its dependence on dataset and model characteristics, remains an open direction for future study.

Trade-offs

These observations describe what changes under hardness-based resampling and why. In practice, applying these methods also requires explicit trade-off choices. We summarize the main trade-offs we observed and their practical implications.

Precision vs. recall. Improving recall-based fairness typically reduces precision-based fairness. That is because oversampling hard classes increases the model’s tendency to predict those classes (TP and FP both rise), which raises recall but lowers precision on the target classes. Hence, practitioners must decide whether recall-oriented fairness (e.g., minimizing missed detections) or precision-oriented fairness (e.g., minimizing false alarms) is the priority for the application at hand.

Fairness vs. overall performance (estimator quality). In our experiments degradation in macro-recall and overall accuracy arises primarily from pruning the wrong samples rather than from pruning too aggressively. That is because easy classes can tolerate pruning even 75% of their sample count without meaningful change in performance. Meanwhile, poor hardness estimators can misidentify difficult yet informative samples as “easy,” leading to under-representation of crucial data and thereby lowering overall performance. Conversely, accurate estimators such as AUM not only improve fairness but also prevent the macro-recall degradation typically caused by misidentifying informative samples. Nevertheless, if fairness is the overriding objective—such as in high-stakes domains where equitable treatment outweighs aggregate accuracy—it remains possible to further increase fairness at the expense of macro-recall. While this trade-off is rarely desirable in benchmarking contexts, we believe that acknowledging its existence is important for settings where fairness has intrinsic priority.

Degree of induced data imbalance (α). The induced data imbalance is introduced deliberately to mitigate hardness imbalance. Empirically, for moderate values of α , this strategy improves both fairness and macro-recall: recall on hard classes rises faster than it decreases on easy ones. However, this benefit comes with an inherent tension. While the induced imbalance mitigates the negative effects of hardness imbalance, it simultaneously introduces those associated with data imbalance. Although pruning easy classes has a relatively small

effect on their recall, it is not negligible. Beyond a certain pruning threshold, this degradation should accumulate to the point where easy classes begin to underperform relative to hard ones, especially as hard classes become overrepresented and increasingly dominate gradient updates. A threshold value of α must therefore exist beyond which the negative effects of induced imbalance outweigh its benefits.

Interpreting F1 as an aggregate outcome. F1 integrates the effects of both precision and recall, and its behavior therefore reflects the balance between their opposing trends. Achieving such balance depends primarily on the quality of the synthetic data and the reliability of hardness estimates: high-fidelity data ensure stronger recall improvements, while accurate estimators prevent pruning wrong samples and preserve overall performance.

In summary, our results demonstrate that hardness-based resampling can improve fairness even in balanced datasets, primarily by reducing recall and F1 gaps across classes. Under favorable conditions—high-quality synthetic data and accurate hardness estimation—these fairness gains can occur alongside improvements in macro-level performance, rather than at its expense. However, the dynamics behind these improvements differ markedly from those observed under conventional data imbalance: while data imbalance is a question of frequency, hardness imbalance is a question of difficulty. As a result, its correction is inherently more sensitive to estimator fidelity and quality of data for oversampling, making hardness imbalance a more challenging and theoretically rich problem.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis set out to formally introduce hardness imbalance, structurize the insights from hardness-aware learning and showcase the practical value of this field. We have done so by systematically reviewing prior work, developing unified conceptual frameworks, and conducting empirical studies. In this final chapter, we summarize our key findings in response to our core research questions and outline promising directions for future research. This allows us both to reflect on the contributions made and to highlight open challenges that remain in hardness-aware learning.

How to estimate hardness? We performed a review of current hardness estimators, identifying two primary families: data-based and model-based approaches. Data-based estimators rely on intrinsic dataset properties, which we broadly grouped into geometry-related issues (e.g., manifold curvature or overlap), sampling-related issues (e.g., within- and between-class data imbalances), and error-related issues (e.g., label noise). These methods are theoretically grounded and interpretable but, as suggested by evidence across the literature, are also highly sensitive to the curse of dimensionality. Model-based estimators, in contrast, assess hardness from model behaviour—through quantities such as confidence, loss dynamics, or gradient information. While often heuristic and less interpretable, they are computationally cheaper and seamlessly integrate into modern learning pipelines.

Together, these two families capture complementary aspects of sample hardness: data-based estimators describe the intrinsic difficulty of the data itself, whereas model-based ones reflect how this difficulty manifests during learning. Yet, prior works tend to treat them in isolation, resulting in fragmented and incompatible taxonomies of hard samples. Drawing on these insights, we proposed a unified taxonomy that organizes both the causes and types of hardness, bridging conceptual gaps between the two perspectives. This taxonomy brings structure to a previously disjointed field and establishes a foundation for systematic evaluation and future research in hardness-aware learning.

Which hardness estimators to use? Benchmarking hardness is inherently challenging due to the absence of ground-truth difficulty labels. Consequently, comparisons between estimators must rely on indirect approaches—such as downstream performance analysis,

synthetic difficulty injection, or proxy-based evaluation—all of which provide only partial insights. Our investigation of how benchmarking is commonly conducted in hardness-aware research revealed two critical open questions:

- *Are data-based hardness estimators a viable alternative to more widely used model-based estimators?*
- *How robust is the common approach of estimating hardness over a single seed?*

To address the first question, we directly compared some of the reviewed data- and model-based estimators. The results indicate that data-based estimators are not a viable alternative. Across datasets and metrics, they exhibited significantly weaker correlations with accuracy (serving as the proxy-based measure). This outcome reinforces our earlier hypothesis that data-based estimators are highly sensitive to the curse of dimensionality: quantities that should correlate—such as intrinsic dimension and volume—failed to do so empirically. Applying these estimators to PCA-reduced representations did not resolve the issue, most likely because the dimensionality reduction distorted manifold geometry and topology, further degrading reliability.

To answer the second question, we examined the robustness of model-based estimators across random seeds. To move beyond coarse, correlation-based comparisons, we introduced two complementary metrics: *top-t stability*, which quantifies consistency in identifying the most extreme samples, and *resampling-ratio stability*, which assesses reliability for our downstream task of hardness-based resampling. This analysis revealed that stability varies substantially across estimators and that single-seed estimates are particularly unreliable when focusing on the hardest samples—those most relevant for tasks such as label-noise detection or outlier removal.

Can hardness-based resampling improve fairness in balanced setting? Many techniques for mitigating data imbalance are implicitly hardness-aware. Yet, in the absence of targeted ablations, it remains unclear whether their benefits arise from correcting data imbalance or from addressing hardness imbalance. To isolate this effect, we conducted controlled experiments testing whether hardness-based resampling—undersampling easy classes and oversampling hard ones—can improve fairness even when the dataset is already balanced. A positive result would challenge the prevailing assumption that uniform class distributions guarantee fairness.

Our experiments revealed that hardness-based resampling reduces recall disparities across classes but tends to increase precision gaps. The latter effects stem primarily from the induced imbalance, whereas recall improvements depend on the quality of oversampling data, the accuracy of hardness estimates, and the degree of imbalance introduced. Consequently, improvements in F1-based fairness and even overall accuracy are achievable, but only when both hardness estimates and generated data are of high quality. These results validate the practical applicability of hardness-aware methods and call into question the traditional focus on data balance. By showing that fairness improvements hinge on hardness rather than data balance, we underscore the need to rethink conventional approaches through a hardness-aware perspective.

6.2 Future Work

This thesis has brought much-needed structure to the field of hardness-aware learning, both by formalizing a taxonomy of hard samples and by identifying key limitations in existing benchmarking strategies for hardness estimators. It has also demonstrated the practical value of adopting a hardness-aware perspective. In this final section, we outline several open directions that, if pursued, could further advance the understanding and application of hardness in modern learning systems.

Poor performance of data-based estimators Our literature review (Chapter 3) and experiments (Chapter 4) indicate that data-based hardness estimators perform poorly when applied directly in the input space, with the curse of dimensionality emerging as the most likely cause. This limitation substantially constrains their practical use: if reliable estimation requires projection into model-derived latent spaces, then the supposed interpretability and model-agnostic appeal of these methods become less convincing. Since latent spaces inevitably encode model biases, relying on them undermines the goal of developing theory-driven, data-grounded measures of hardness. Addressing this issue remains an open challenge. Progress may require either improving the robustness of geometric and topological estimators in high-dimensional settings or identifying dimensionality reduction strategies that better preserve the intrinsic data structure. Advancing in either direction would strengthen the theoretical foundation of hardness-aware learning and help re-establish the distinct role of data-based approaches within it.

Limited knowledge on training dynamics Our literature review (Chapter 3) and experiments (Chapter 4) indicate that incorporating training dynamics into hardness estimation improves both stability and accuracy of hardness estimates. This observation builds on the growing evidence that learning unfolds in distinct phases and that model dynamics encode rich information about sample difficulty. Yet, our understanding of what these dynamics truly capture remains at a very early stage. Jia et al. (2023) demonstrated that label noise can be identified using signals derived solely from training dynamics, showing that such signals are both informative and transferable. Furthermore, several studies have shown that training can be broadly divided into two stages, with models first focusing on easier samples before progressing to harder ones. However, these findings only scratch the surface. We do not yet know how much of a model’s behaviour is encoded in its dynamics, how early in training these signals become meaningful, or whether they can differentiate between distinct forms of hardness such as label noise, geometric complexity, or class overlap. More broadly, the structure of the learning process itself—its phases, transitions, and implicit representations—remains largely unmapped. Understanding this hidden landscape may prove essential not only for building more faithful hardness estimators but also for uncovering the deeper mechanisms through which neural networks learn.

Unstructured benchmarking in the literature As discussed in Chapter 4, benchmarking hardness estimators remains fundamentally challenging due to the absence of ground-truth hardness labels. Without such reference, researchers have adopted diverse and often incomparable evaluation strategies. Our review identified three broad benchmarking paradigms

currently used in the literature, but none provides a principled way to assess estimator reliability in the absence of ground truth. This lack of structure extends to how “hard samples” themselves are defined. In the absence of a shared taxonomy, almost each study introduces its own categorization—such as easy versus hard, clean versus mislabeled, or easy, complex, rare, and mislabeled—making cross-paper insights difficult to reconcile. This heterogeneity fragments the field and obscures cumulative understanding. Future progress may depend on the development of more coherent benchmarking frameworks and the adoption of common taxonomies that enable systematic comparison, integration, and synthesis of results across studies. We hope that the taxonomy proposed in Chapter 3 can serve as an initial foundation toward this goal, while recognizing that it remains incomplete, as different domains exhibit distinct sources of hardness—such as ambiguity and context dependence in NLP, or occlusion and illumination in computer vision.

Lack of fairness oriented evaluation metrics A further concern identified in this thesis is the over-reliance on standard accuracy metrics (e.g., top-1, top-5) in model evaluation, including many benchmark competitions. While accuracy remains useful, it frequently masks poor performance on classes that are underrepresented or intrinsically hard. Recent work shows that strong data augmentation can improve average accuracy even as some minority classes suffer large drops in class-level accuracy (Balestriero et al. 2022, Kirichenko et al. 2023). From the viewpoint of hardness-aware learning, this issue is especially critical: the goal is to improve performance on hard classes, not merely to boost aggregate averages. Accordingly, this thesis argues that evaluation protocols and benchmarks should include fairness-aware metrics—such as the gap- and dispersion-based measures proposed in Chapter 5—as standard practice.

Although the importance of fairness metrics is well understood in data imbalanced settings, such considerations are rarely extended to balanced datasets, as hardness imbalance is not yet widely recognized as a key source of unfairness. Our experiments on simple datasets like CIFAR-10 and CIFAR-100 already reveal substantial class-level disparities, implying that the problem is likely even more pronounced in complex datasets such as ImageNet. Addressing this overlooked dimension of fairness will require shifting community norms toward making hardness-aware metrics a primary criterion in benchmarking and competition settings.

Toward hardness-aware generative modeling Our experiments in Chapter 5 showed that the effectiveness of hardness-based resampling depends strongly on the quality of the data used for oversampling. In our first case study, where real holdout samples were available, the improvements in class-level fairness were more substantial than in the second case study, which relied on synthetic data generated by a diffusion model. Within the latter, the largest relative gains were achieved when oversampling the hardest synthetic samples—suggesting that these represent the most informative regions of the generative distribution. A few recent studies have also noted that samples produced by generative models tend to be biased toward easier regions of the data manifold, and that encouraging the generation of harder samples can yield measurable downstream benefits. Taken together, these findings suggest that current generative models still struggle to capture the full hardness spectrum present in real data.

As we detailed in Section 5.2.3, this shortfall may stem, at least in part, from the evalu-

ation criteria that guide generative model development. Metrics such as the Inception Score and Fréchet Inception Distance prioritize entropy and global statistical similarity, inadvertently favoring the generation of easy, high-confidence samples. As a result, high-scoring generative models may still underperform in reproducing the harder, more informative regions of the true data distribution. Future work should therefore pursue two complementary directions: (1) developing hardness-aware evaluation metrics that assess the faithfulness of the generated hardness spectrum, and (2) designing hardness-aware generative models that explicitly target hard or underrepresented samples. While early efforts along these lines exist, our findings highlight their importance and suggest that this direction remains both underexplored and promising.

Data-Hardness Trade-offs Our experiments in Chapter 5 revealed that the positive impact of hardness-based resampling tends to grow with the degree of induced data imbalance. In other words, the benefits of correcting hardness imbalance often outweigh the drawbacks of introducing moderate levels of data imbalance. Nevertheless, this relationship cannot be unbounded. It is reasonable to expect a tipping point beyond which excessive imbalance degrades model performance—simply because certain classes become too underrepresented for the model to learn them effectively. While our study did not explicitly locate this point, its existence is almost certain and raises several open questions. Can this trade-off be characterized theoretically, or can it only be identified empirically? How does it depend on the hardness spectrum of the dataset? Addressing these questions could lay the groundwork for a more principled understanding of how data imbalance and hardness imbalance interact, ultimately enabling the design of adaptive resampling strategies that maintain fairness without sacrificing representational diversity.

Bibliography

- Agarwal, C., D'souza, D. & Hooker, S. (2022), Estimating example difficulty using variance of gradients, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 10368–10378.
- Ahn, S., Ko, J. & Yun, S. (2023), CUDA: curriculum of data augmentation for long-tailed recognition, *in* 'The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023', OpenReview.net.
URL: <https://openreview.net/forum?id=RgUPdudkWLN>
- Ali-Gombe, A. & Elyan, E. (2019), 'Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network', *Neurocomputing* **361**, 212–221.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. & Asari, V. K. (2019), 'A state-of-the-art survey on deep learning theory and architectures', *electronics* **8**(3), 292.
- Ansuini, A., Laio, A., Macke, J. H. & Zoccolan, D. (2019), 'Intrinsic dimension of data representations in deep neural networks', *Advances in Neural Information Processing Systems* **32**.
- Arora, S., Ge, R., Neyshabur, B. & Zhang, Y. (2018), Stronger generalization bounds for deep nets via a compression approach, *in* 'International conference on machine learning', PMLR, pp. 254–263.
- Arriaga, O., Palacio, S. & Valdenegro-Toro, M. (2023), Difficulty estimation with action scores for computer vision tasks, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 245–253.
- Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H. & Khan, F. S. (2025), 'Foundation models defining a new era in vision: a survey and outlook', *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Balcan, M.-F., Hanneke, S. & Vaughan, J. W. (2010), 'The true sample complexity of active learning', *Machine learning* **80**(2), 111–139.
- Balestriero, R., Bottou, L. & LeCun, Y. (2022), 'The effects of regularization and data augmentation are class dependent', *Advances in Neural Information Processing Systems* **35**, 37878–37891.

- Bengio, Y., Louradour, J., Collobert, R. & Weston, J. (2009), Curriculum learning, in ‘Proceedings of the 26th annual international conference on machine learning’, pp. 41–48.
- Berner, J., Grohs, P., Kutyniok, G. & Petersen, P. (2021), ‘The modern mathematics of deep learning’, *arXiv preprint arXiv:2105.04026* **78**, 3.
- Bisla, D., Saridena, A. N. & Choromanska, A. (2021), A theoretical-empirical approach to estimating sample complexity of dnns, in ‘Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition’, pp. 3270–3280.
- Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M. K. (1989), ‘Learnability and the vapnik-chervonenkis dimension’, *Journal of the ACM (JACM)* **36**(4), 929–965.
- Brahma, P. P., Wu, D. & She, Y. (2015), ‘Why deep learning works: A manifold disentanglement perspective’, *IEEE transactions on neural networks and learning systems* **27**(10), 1997–2008.
- Buetti-Dinh, A., Galli, V., Bellenberg, S., Ilie, O., Herold, M., Christel, S., Boretska, M., Pivkin, I. V., Wilmes, P., Sand, W. et al. (2019), ‘Deep neural networks outperform human expert’s capacity in characterizing bioleaching bacterial biofilm composition’, *Biotechnology Reports* **22**, e00321.
- Carvalho, M., Pinho, A. J. & Brás, S. (2025), ‘Resampling approaches to handle class imbalance: a review from a data perspective’, *Journal of Big Data* **12**(1), 71.
- Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002), ‘Smote: synthetic minority over-sampling technique’, *Journal of artificial intelligence research* **16**, 321–357.
- Chen, Y. (n.d.), ‘Pytorch cifar models’, <https://github.com/chenyaofo/pytorch-cifar-models>. Accessed: 2025-7-22.
- Ciceri, S., Cassani, L., Osella, M., Rotondo, P., Valle, F. & Gherardi, M. (2024), ‘Inversion dynamics of class manifolds in deep learning reveals tradeoffs underlying generalization’, *Nature Machine Intelligence* **6**(1), 40–47.
- Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N. & Al Najada, H. (2015), ‘Survey of review spam detection using machine learning techniques’, *Journal of Big Data* **2**(1), 23.
- Cybenko, G. (1989), ‘Approximation by superpositions of a sigmoidal function’, *Mathematics of control, signals and systems* **2**(4), 303–314.
- Dablain, D., Krawczyk, B. & Chawla, N. V. (2022), ‘Deepsmote: Fusing deep learning and smote for imbalanced data’, *IEEE transactions on neural networks and learning systems* **34**(9), 6390–6404.
- Darlow, L. N., Crowley, E. J., Antoniou, A. & Storkey, A. J. (2018), ‘Cin10 is not imagenet or cifar-10’, *arXiv preprint arXiv:1810.03505*.
- Deng, L. (2012), ‘The mnist database of handwritten digit images for machine learning research’, *IEEE Signal Processing Magazine* **29**(6), 141–142.

- Ding, X., Zhang, X., Ma, N., Han, J., Ding, G. & Sun, J. (2021), Repvgg: Making vgg-style convnets great again, in ‘Proceedings of the IEEE/CVF conference on computer vision and pattern recognition’, pp. 13733–13742.
- Doersch, C. (2016), ‘Tutorial on variational autoencoders’, *arXiv preprint arXiv:1606.05908* .
- Dominguez-Catena, I., Paternain, D. & Galar, M. (2024), ‘Metrics for dataset demographic bias: A case study on facial expression recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **46**(8), 5209–5226.
- Dundar, M., Krishnapuram, B., Bi, J. & Rao, R. B. (2007), Learning classifiers when the training data is not iid., in ‘IJCAI’, Vol. 2007, San Francisco, CA, pp. 756–61.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. & Thrun, S. (2017), ‘Dermatologist-level classification of skin cancer with deep neural networks’, *nature* **542**(7639), 115–118.
- Facco, E., d’Errico, M., Rodriguez, A. & Laio, A. (2017), ‘Estimating the intrinsic dimension of datasets by a minimal neighborhood information’, *Scientific reports* **7**(1), 12140.
- Fefferman, C., Mitter, S. & Narayanan, H. (2016), ‘Testing the manifold hypothesis’, *Journal of the American Mathematical Society* **29**(4), 983–1049.
- Feng, C., Zhong, Y. & Huang, W. (2021), Exploring classification equilibrium in long-tailed object detection, in ‘2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021’, IEEE, pp. 3397–3406.
URL: <https://doi.org/10.1109/ICCV48922.2021.00340>
- Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F., Fernández, A., García, S., Galar, M., Prati, R. C. et al. (2018), ‘Cost-sensitive learning’, *Learning from imbalanced data sets* pp. 63–78.
- Ferracci, T., Goldmann, L. T., Hinel, A. & Passino, F. S. (2024), ‘Targeted synthetic data generation for tabular data via hardness characterization’, *arXiv preprint arXiv:2410.00759* .
- Fotouhi, S., Asadi, S. & Kattan, M. W. (2019), ‘A comprehensive data level analysis for cancer diagnosis on imbalanced data’, *Journal of biomedical informatics* **90**, 103089.
- Friedman, N. & Yakhini, Z. (2013), ‘On the sample complexity of learning bayesian networks’, *arXiv preprint arXiv:1302.3579* .
- Gamba, M., Chmielewski-Anders, A., Sullivan, J., Azizpour, H. & Bjorkman, M. (2022), Are all linear regions created equal?, in ‘International Conference on Artificial Intelligence and Statistics’, PMLR, pp. 6573–6590.
- Garg, S., Jha, S., Mahloulifar, S., Mahmood, M. & Wang, M. (2022), ‘Overparameterization from computational constraints’, *Advances in Neural Information Processing Systems* **35**, 13557–13569.

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2020), ‘Generative adversarial networks’, *Communications of the ACM* **63**(11), 139–144.
- Griffin, G., Holub, A., Perona, P. et al. (2007), Caltech-256 object category dataset, Technical report, Technical Report 7694, California Institute of Technology Pasadena.
- Guo, X., Yin, Y., Dong, C., Yang, G. & Zhou, G. (2008), On the class imbalance problem, *in* ‘2008 Fourth international conference on natural computation’, Vol. 4, IEEE, pp. 192–201.
- Han, H., Wang, W.-Y. & Mao, B.-H. (2005), Borderline-smote: a new over-sampling method in imbalanced data sets learning, *in* ‘International conference on intelligent computing’, Springer, pp. 878–887.
- Hanneke, S. (2009), *Theoretical foundations of active learning*, Carnegie Mellon University.
- Hasanin, T. & Khoshgoftaar, T. (2018), The effects of random undersampling with simulated class imbalance for big data, *in* ‘2018 IEEE international conference on information reuse and integration (IRI)’, IEEE, pp. 70–79.
- He, H., Bai, Y., Garcia, E. A. & Li, S. (2008), Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *in* ‘2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)’, Ieee, pp. 1322–1328.
- He, K., Zhang, X., Ren, S. & Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *in* ‘Proceedings of the IEEE international conference on computer vision’, pp. 1026–1034.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 770–778.
- He, Y., Shen, Z. & Cui, P. (2021), ‘Towards non-iid image classification: A dataset and baselines’, *Pattern Recognition* **110**, 107383.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. (2017), ‘Gans trained by a two time-scale update rule converge to a local nash equilibrium’, *Advances in neural information processing systems* **30**.
- Ho, J., Jain, A. & Abbeel, P. (2020), ‘Denoising diffusion probabilistic models’, *Advances in neural information processing systems* **33**, 6840–6851.
- Holte, R. C., Acker, L., Porter, B. W. et al. (1989), Concept learning and the problem of small disjuncts., *in* ‘IJCAI’, Vol. 89, pp. 813–818.
- Japkowicz, N. (2001), Concept-learning in the presence of between-class and within-class imbalances, *in* E. Stroulia & S. Matwin, eds, ‘Advances in Artificial Intelligence, 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2001, Ottawa, Canada, June 7-9, 2001, Proceedings’, Vol. 2056 of *Lecture Notes in Computer Science*, Springer, pp. 67–77.
- URL:** https://doi.org/10.1007/3-540-45153-6_7

- Jia, Q., Li, X., Yu, L., Bian, J., Zhao, P., Li, S., Xiong, H. & Dou, D. (2023), Learning from training dynamics: Identifying mislabeled data beyond manually designed features, in ‘Proceedings of the AAAI Conference on Artificial Intelligence’, Vol. 37, pp. 8041–8049.
- Jo, T. & Japkowicz, N. (2004), ‘Class imbalances versus small disjuncts’, *ACM Sigkdd Explorations Newsletter* **6**(1), 40–49.
- Johnson, J. M. & Khoshgoftaar, T. M. (2019), ‘Survey on deep learning with class imbalance’, *Journal of big data* **6**(1), 1–54.
- Karras, T., Aittala, M., Aila, T. & Laine, S. (2022), ‘Elucidating the design space of diffusion-based generative models’, *Advances in neural information processing systems* **35**, 26565–26577.
- Kaufman, I. & Azencot, O. (2023), Data representations’ study of latent image manifolds, in ‘International Conference on Machine Learning’, PMLR, pp. 15928–15945.
- Kaushik, C., Liu, R., Lin, C., Khera, A., Jin, M. Y., Ma, W., Muthukumar, V. & Dyer, E. L. (2024), Balanced data, imbalanced spectra: Unveiling class disparities with spectral imbalance, in ‘Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21–27, 2024’, OpenReview.net.
URL: <https://openreview.net/forum?id=s4EYBJ30WY>
- Kawaguchi, K., Deng, Z., Ji, X. & Huang, J. (2023), How does information bottleneck help deep learning?, in ‘International Conference on Machine Learning’, PMLR, pp. 16049–16096.
- Kienitz, D., Komendantskaya, E. & Lones, M. A. (2022), The effect of manifold entanglement and intrinsic dimensionality on learning, in ‘Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022’, AAAI Press, pp. 7160–7167.
URL: <https://doi.org/10.1609/aaai.v36i7.20676>
- Kirichenko, P., Ibrahim, M., Balestriero, R., Bouchacourt, D., Vedantam, S. R., Firooz, H. & Wilson, A. G. (2023), ‘Understanding the detrimental class-level effects of data augmentation’, *Advances in Neural Information Processing Systems* **36**, 17498–17526.
- Kothapalli, V. (2023), ‘Neural collapse: A review on modelling principles and generalization’, *Trans. Mach. Learn. Res.* **2023**.
URL: <https://openreview.net/forum?id=QTXocpAP9p>
- Krizhevsky, A., Hinton, G. et al. (2009), ‘Learning multiple layers of features from tiny images’.
- Kubat, M. (1997), Addressing the curse of imbalanced training sets: one-sided selection, in ‘Proceedings of the 14th international conference on machine learning’, Morgan Kaufmann, pp. 179–186.

- Kubat, M., Holte, R. C. & Matwin, S. (1998), ‘Machine learning for the detection of oil spills in satellite radar images’, *Mach. Learn.* **30**(2-3), 195–215.
URL: <https://doi.org/10.1023/A:1007452223027>
- Kumar, M., Packer, B. & Koller, D. (2010), ‘Self-paced learning for latent variable models’, *Advances in neural information processing systems* **23**.
- Lampinen, J. & Vehtari, A. (2001), ‘Bayesian approach for neural networks—review and case studies’, *Neural networks* **14**(3), 257–274.
- Le, Y. & Yang, X. (2015), ‘Tiny imagenet visual recognition challenge’, *CS 231N* **7**(7), 3.
- Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A. & Seliya, N. (2018), ‘A survey on addressing high-class imbalance in big data’, *J. Big Data* **5**, 42.
URL: <https://doi.org/10.1186/s40537-018-0151-6>
- Lessmann, S., Baesens, B., Seow, H.-V. & Thomas, L. C. (2015), ‘Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research’, *European journal of operational research* **247**(1), 124–136.
- Levina, E. & Bickel, P. (2004), ‘Maximum likelihood estimation of intrinsic dimension’, *Advances in neural information processing systems* **17**.
- Li, B., Han, Z., Li, H., Fu, H. & Zhang, C. (2022), Trustworthy long-tailed classification, in ‘IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022’, IEEE, pp. 6960–6969.
URL: <https://doi.org/10.1109/CVPR52688.2022.00684>
- Li, D., Wang, Z., Chen, Y., Jiang, R., Ding, W. & Okumura, M. (2024), ‘A survey on deep active learning: Recent advances and new frontiers’, *IEEE Transactions on Neural Networks and Learning Systems* **36**(4), 5879–5899.
- Li, W., Wang, L., Li, W., Agustsson, E. & Van Gool, L. (2017), ‘Webvision database: Visual learning and understanding from web data’, *arXiv preprint arXiv:1708.02862*.
- Li, Y. (2018), ‘Curvature-aware manifold learning’, *Pattern Recognition* **83**, 273–286.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2017), Focal loss for dense object detection, in ‘Proceedings of the IEEE international conference on computer vision’, pp. 2980–2988.
- Liu, X.-Y., Wu, J. & Zhou, Z.-H. (2008), ‘Exploratory undersampling for class-imbalance learning’, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**(2), 539–550.
- Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C. & Ho, T. K. (2019), ‘How complex is your classification problem? a survey on measuring classification complexity’, *ACM Computing Surveys (CSUR)* **52**(5), 1–34.
- Lorena, A. C., Maciel, A. I., de Miranda, P. B., Costa, I. G. & Prudêncio, R. B. (2018), ‘Data complexity meta-features for regression problems’, *Machine Learning* **107**(1), 209–246.

- Ma, N., Zhang, X., Zheng, H.-T. & Sun, J. (2018), Shufflenet v2: Practical guidelines for efficient cnn architecture design, *in* ‘Proceedings of the European conference on computer vision (ECCV)’, pp. 116–131.
- Ma, Y., Jiao, L., Liu, F., Li, L., Ma, W., Yang, S., Liu, X. & Chen, P. (2024), ‘Unveiling and mitigating generalized biases of dnns through the intrinsic dimensions of perceptual manifolds’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Ma, Y., Jiao, L., Liu, F., Li, Y., Yang, S. & Liu, X. (2023), Delving into semantic scale imbalance, *in* ‘The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023’, OpenReview.net.
URL: <https://openreview.net/forum?id=07tc5kKRIO>
- Ma, Y., Jiao, L., Liu, F., Yang, S., Liu, X. & Li, L. (2023), Curvature-balanced feature manifold learning for long-tailed classification, *in* ‘Proceedings of the IEEE/CVF conference on computer vision and pattern recognition’, pp. 15824–15835.
- Magai, G. (2023), Deep neural networks architectures from the perspective of manifold learning, *in* ‘2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)’, IEEE, pp. 1021–1031.
- Magris, M. & Iosifidis, A. (2023), ‘Bayesian learning for neural networks: an algorithmic survey’, *Artificial Intelligence Review* **56**(10), 11773–11823.
- Maini, P., Garg, S., Lipton, Z. & Kolter, J. Z. (2022), ‘Characterizing datapoints via second-split forgetting’, *Advances in Neural Information Processing Systems* **35**, 30044–30057.
- Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M. & Zeineddine, H. (2019), ‘An experimental study with imbalanced classification approaches for credit card fraud detection’, *IEEE Access* **7**, 93010–93022.
URL: <https://doi.org/10.1109/ACCESS.2019.2927266>
- Marchesi, R., Micheletti, N., Kuo, N. I.-H., Barbieri, S., Jurman, G. & Osmani, V. (2023), ‘Generative ai mitigates representation bias and improves model fairness through synthetic health data’, *medRxiv* pp. 2023–09.
- McCloskey, M. & Cohen, N. J. (1989), Catastrophic interference in connectionist networks: The sequential learning problem, *in* ‘Psychology of learning and motivation’, Vol. 24, Elsevier, pp. 109–165.
- Mindermann, S., Brauner, J. M., Razzak, M. T., Sharma, M., Kirsch, A., Xu, W., Hölting, B., Gomez, A. N., Morisot, A., Farquhar, S. et al. (2022), Prioritized training on points that are learnable, worth learning, and not yet learnt, *in* ‘International Conference on Machine Learning’, PMLR, pp. 15630–15649.
- Mohammed, R., Rawashdeh, J. & Abdullah, M. (2020), Machine learning with oversampling and undersampling techniques: overview study and experimental results, *in* ‘2020 11th international conference on information and communication systems (ICICS)’, IEEE, pp. 243–248.

- Mukhoti, J., Kirsch, A., Van Amersfoort, J., Torr, P. & Gal, Y. (2021), ‘Deterministic neural networks with inductive biases capture epistemic and aleatoric uncertainty’, *arXiv preprint arXiv:2102.11582* **2**.
- Naitzat, G., Zhitnikov, A. & Lim, L.-H. (2020), ‘Topology of deep neural networks’, *Journal of Machine Learning Research* **21**(184), 1–40.
- Narayanan, H. & Mitter, S. (2010), ‘Sample complexity of testing the manifold hypothesis’, *Advances in neural information processing systems* **23**.
- Narayanan, H. & Niyogi, P. (2009), On the sample complexity of learning smooth cuts on a manifold., in ‘COLT’.
- Nasiri, S., Ganglberger, W., Nassi, T., Meulenbrugge, E.-J., Moura Junior, V., Ghanta, M., Gupta, A., Stone, K. L., Kjaer, M. R., Sum-Ping, O. et al. (2025), ‘Caisr: Achieving human-level performance in automated sleep analysis across all clinical sleep metrics’, *Sleep* p. zsaf134.
- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. & Ng, A. (2011), Reading digits in natural images with unsupervised feature learning, in ‘NIPS workshop on deep learning and unsupervised feature learning’.
URL: <https://api.semanticscholar.org/CorpusID:16852518>
- Neyshabur, B. (2017), ‘Implicit regularization in deep learning’, *arXiv preprint arXiv:1709.01953*.
- Neyshabur, B., Bhojanapalli, S. & Srebro, N. (2018), A pac-bayesian approach to spectrally-normalized margin bounds for neural networks, in ‘6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings’, OpenReview.net.
URL: https://openreview.net/forum?id=Skz_WfbCZ
- Olah, C. (2014), ‘Neural networks, manifolds, and topology’, <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>. Posted April 6, 2014, Accessed September 18, 2024.
- Papayan, V., Han, X. & Donoho, D. L. (2020), ‘Prevalence of neural collapse during the terminal phase of deep learning training’, *Proceedings of the National Academy of Sciences* **117**(40), 24652–24663.
- Paul, M., Ganguli, S. & Dziugaite, G. K. (2021), ‘Deep learning on a data diet: Finding important examples early in training’, *Advances in neural information processing systems* **34**, 20596–20607.
- Pennisi, M., Palazzo, S. & Spampinato, C. (2021), Self-improving classification performance through gan distillation, in ‘Proceedings of the IEEE/CVF International Conference on Computer Vision’, pp. 1640–1648.
- Pleiss, G., Zhang, T., Elenberg, E. & Weinberger, K. Q. (2020), ‘Identifying mislabeled data using the area under the margin ranking’, *Advances in Neural Information Processing Systems* **33**, 17044–17056.

- Pope, P., Zhu, C., Abdelkader, A., Goldblum, M. & Goldstein, T. (2021), The intrinsic dimension of images and its impact on learning, in ‘9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021’, OpenReview.net.
URL: <https://openreview.net/forum?id=XJk19XzGq2J>
- Prusa, J., Khoshgoftaar, T. M., Dittman, D. J. & Napolitano, A. (2015), Using random undersampling to alleviate class imbalance on tweet sentiment data, in ‘2015 IEEE international conference on information reuse and integration’, IEEE, pp. 197–202.
- Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X. & Wang, X. (2021), ‘A survey of deep active learning’, *ACM computing surveys (CSUR)* **54**(9), 1–40.
- Ren, Z., Lin, T., Feng, K., Zhu, Y., Liu, Z. & Yan, K. (2023), ‘A systematic review on imbalanced learning methods in intelligent fault diagnosis’, *IEEE Transactions on Instrumentation and Measurement* **72**, 1–35.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. (2016), ‘Improved techniques for training gans’, *Advances in neural information processing systems* **29**.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018), Mobilenetv2: Inverted residuals and linear bottlenecks, in ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 4510–4520.
- Sarridis, I., Koutlis, C., Papadopoulos, S. & Diou, C. (2024), ‘Flac: Fairness-aware representation learning by suppressing attribute-class associations’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Sasada, T., Liu, Z., Baba, T., Hatano, K. & Kimura, Y. (2020), ‘A resampling method for imbalanced datasets considering noise and overlap’, *Procedia Computer Science* **176**, 420–429.
- Seedat, N., Crabbé, J., Bica, I. & van der Schaar, M. (2022), ‘Data-iq: Characterizing subgroups with heterogeneous outcomes in tabular data’, *Advances in Neural Information Processing Systems* **35**, 23660–23674.
- Seedat, N., Imrie, F. & van der Schaar, M. (2024), ‘Dissecting sample hardness: A fine-grained analysis of hardness characterization methods for data-centric ai’, *arXiv preprint arXiv:2403.04551* .
- Settles, B. (2009), ‘Active learning literature survey’.
- Siddiqi, S., Qureshi, F., Lindstaedt, S. & Kern, R. (2023), ‘Detecting outliers in non-iid data: A systematic literature review’, *IEEE Access* **11**, 70333–70352.
- Simonyan, K. & Zisserman, A. (2015), Very deep convolutional networks for large-scale image recognition, in Y. Bengio & Y. LeCun, eds, ‘3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings’.
URL: <http://arxiv.org/abs/1409.1556>

- Sinha, S., Ohashi, H. & Nakamura, K. (2020), Class-wise difficulty-balanced loss for solving class-imbalance, *in* ‘Proceedings of the Asian conference on computer vision’.
- Sinha, S., Ohashi, H. & Nakamura, K. (2022), ‘Class-difficulty based methods for long-tailed visual recognition’, *International Journal of Computer Vision* **130**(10), 2517–2531.
- Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S. & Morcos, A. (2022), ‘Beyond neural scaling laws: beating power law scaling via data pruning’, *Advances in Neural Information Processing Systems* **35**, 19523–19536.
- Soviany, P., Ionescu, R. T., Rota, P. & Sebe, N. (2022), ‘Curriculum learning: A survey’, *International Journal of Computer Vision* **130**(6), 1526–1565.
- Strathern, M. (1997), “improving ratings’: audit in the british university system’, *European review* **5**(3), 305–321.
- Stutz, D., Hein, M. & Schiele, B. (2020), Confidence-calibrated adversarial training: Generalizing to unseen attacks, *in* ‘International conference on machine learning’, PMLR, pp. 9155–9166.
- Suresh, S., Das, B., Abrol, V. & Roy, S. D. (2024), ‘On characterizing the evolution of embedding space of neural networks using algebraic topology’, *Pattern Recognition Letters* **179**, 165–171.
- Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A. & Choi, Y. (2020), ‘Dataset cartography: Mapping and diagnosing datasets with training dynamics’, *arXiv preprint arXiv:2009.10795*.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 2818–2826.
- Tao, L., Dong, M. & Xu, C. (2023), Dual focal loss for calibration, *in* ‘International Conference on Machine Learning’, PMLR, pp. 33833–33849.
- Tharwat, A. & Schenck, W. (2023), ‘A survey on active learning: State-of-the-art, practical challenges and research directions’, *Mathematics* **11**(4), 820.
- Tishby, N. & Zaslavsky, N. (2015), Deep learning and the information bottleneck principle, *in* ‘2015 IEEE information theory workshop (ITW)’, Ieee, pp. 1–5.
- Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y. & Gordon, G. J. (2019), An empirical study of example forgetting during deep neural network learning, *in* ‘7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019’, OpenReview.net.
URL: <https://openreview.net/forum?id=BJlxm30cKm>
- Turay, T. & Vladimirova, T. (2022), ‘Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey’, *IEEE access* **10**, 14076–14119.

- Van den Goorbergh, R., van Smeden, M., Timmerman, D. & Van Calster, B. (2022), ‘The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression’, *Journal of the American Medical Informatics Association* **29**(9), 1525–1534.
- Vandenhende, S., De Brabandere, B., Neven, D. & Van Gool, L. (2019), A three-player gan: generating hard samples to improve classification networks, in ‘2019 16th International Conference on Machine Vision Applications (MVA)’, IEEE, pp. 1–6.
- Vapnik, V. N. (1999), ‘An overview of statistical learning theory’, *IEEE transactions on neural networks* **10**(5), 988–999.
- Vu, D.-Q., Phung, T. T., Wang, J.-C. & Mai, S. T. (2024), ‘Lcsl: long-tailed classification via self-labeling’, *IEEE Transactions on Circuits and Systems for Video Technology* **34**(11), 12048–12058.
- Wang, X., Chen, Y. & Zhu, W. (2021), ‘A survey on curriculum learning’, *IEEE transactions on pattern analysis and machine intelligence* **44**(9), 4555–4576.
- Wang, Z., Mao, J., Wang, X. & Yamasaki, T. (2024), ‘Training data synthesis with difficulty controlled diffusion model’, *arXiv preprint arXiv:2411.18109*.
- Wang, Z., Pang, T., Du, C., Lin, M., Liu, W. & Yan, S. (2023), Better diffusion models further improve adversarial training, in ‘International conference on machine learning’, PMLR, pp. 36246–36263.
- Weinan, E. & Wojtowytsch, S. (2022), On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers, in ‘Mathematical and Scientific Machine Learning’, PMLR, pp. 270–290.
- Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S. & Perona, P. (2010), ‘Caltech-ucsd birds 200’.
- Xiao, T., Xia, T., Yang, Y., Huang, C. & Wang, X. (2015), Learning from massive noisy labeled data for image classification, in ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 2691–2699.
- Yang, J., Zhou, K., Li, Y. & Liu, Z. (2024), ‘Generalized out-of-distribution detection: A survey’, *International Journal of Computer Vision* **132**(12), 5635–5662.
- Yuan, S., Lin, R., Feng, L., Han, B. & Liu, T. (2025), Instance-dependent early stopping, in ‘The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24–28, 2025’, OpenReview.net.
URL: <https://openreview.net/forum?id=P42DbV2nuV>
- Zheng, M., Li, T., Zhu, R., Tang, Y., Tang, M., Lin, L. & Ma, Z. (2020), ‘Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification’, *Information Sciences* **512**, 1009–1023.
- Zhou, Z.-H. (2021), ‘Why over-parameterization of deep neural networks does not overfit?’, *Science China. Information Sciences* **64**(1), 116101.

- Zhu, W., Wu, O., Su, F. & Deng, Y. (2024), ‘Exploring the learning difficulty of data: Theory and measure’, *ACM Transactions on Knowledge Discovery from Data* **18**(4), 1–37.

Appendices

Appendix A

Additional Experimental Results

In this section we include further experiments conducted for Chapter 4. Specifically, we show the changes to True positive, True negatives, False positives, and False negatives across classes as an effect of hardness-based resampling for top-5 most extreme classes. These classes were chosen based on the recall averaged over the sixteen models trained on $\mathcal{D}^{CLP}$ with pruning rate of 50%. This means that only the five classes with the highest and lowest averaged recalls are displayed, with the former representing the easiest classes and the latter the hardest ones. We decided to use Recall instead of AUM to sort classes, as we consider it as ground truth for hardness.

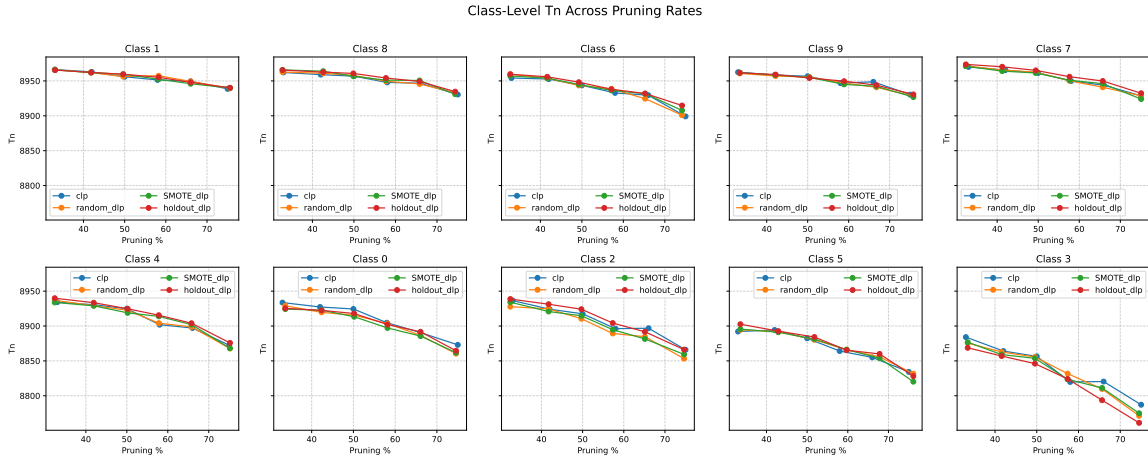


Figure A.1: *True negatives across classes on CIFAR-10.*

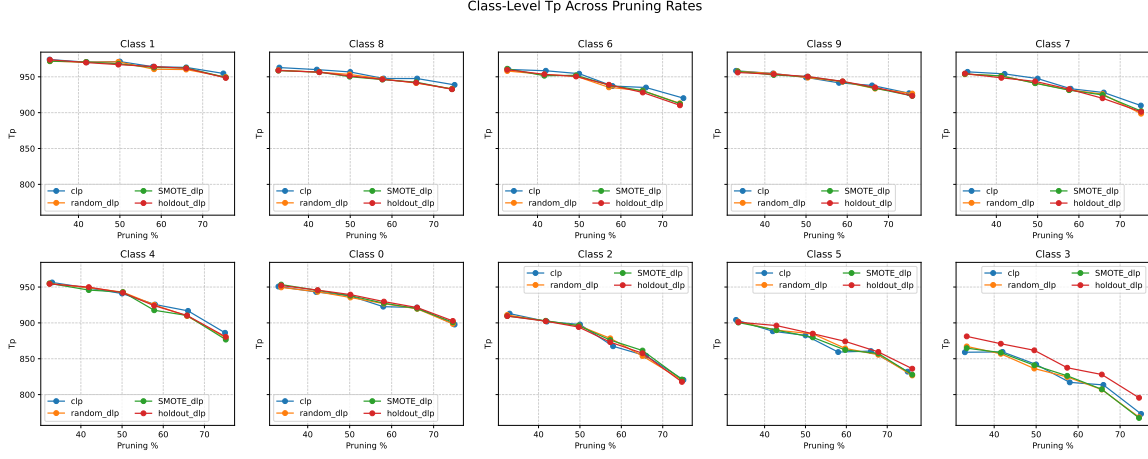


Figure A.2: True positives across classes on CIFAR-10.

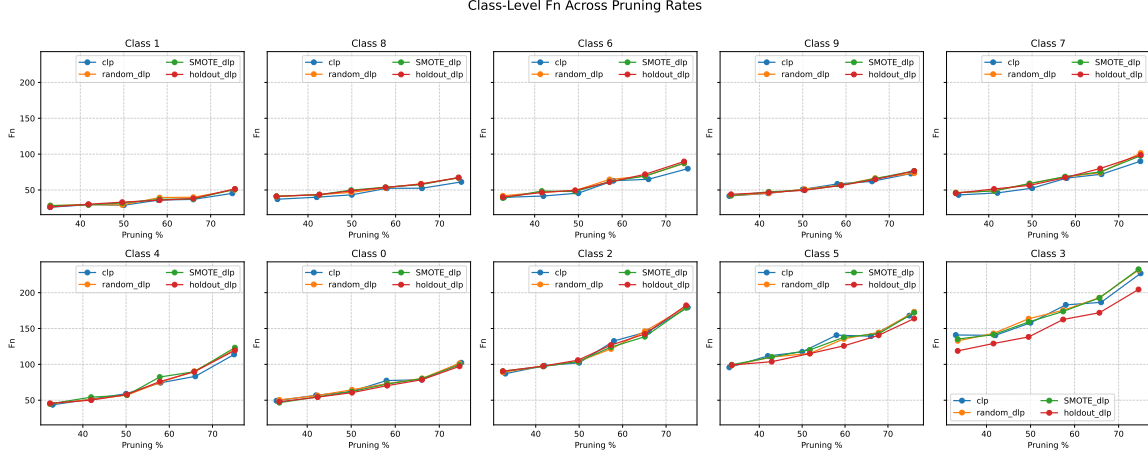


Figure A.3: False negatives across classes on CIFAR-10.

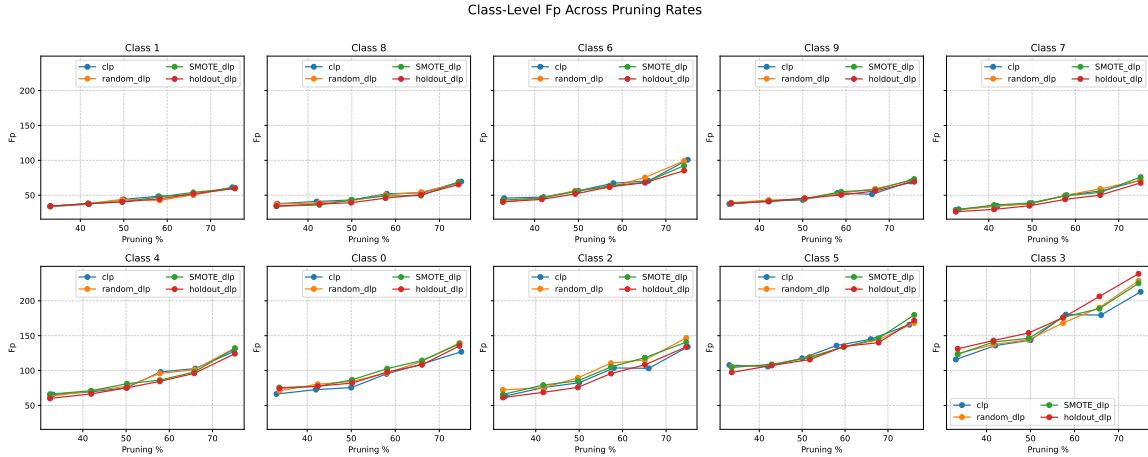


Figure A.4: False positives across classes on CIFAR-10.

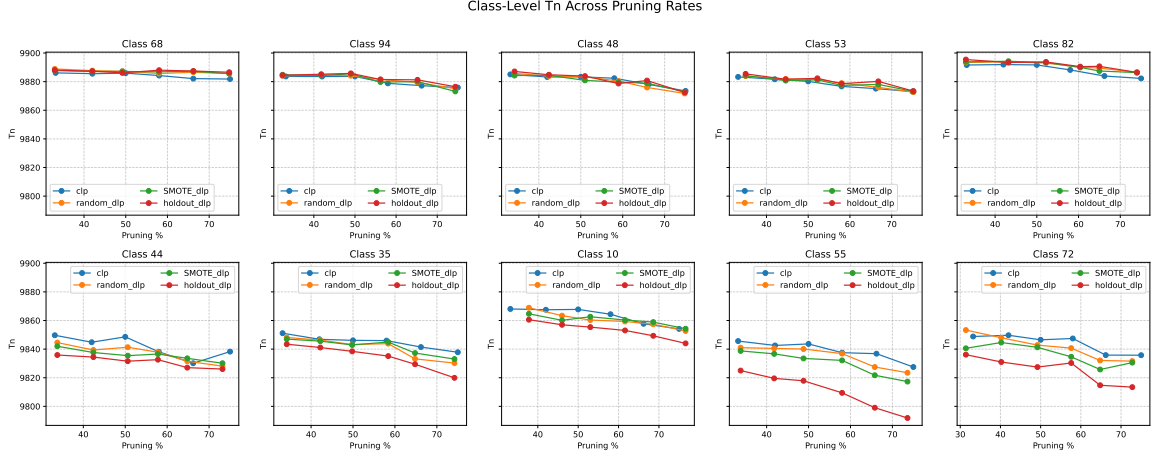


Figure A.5: True negatives across classes on CIFAR-100.

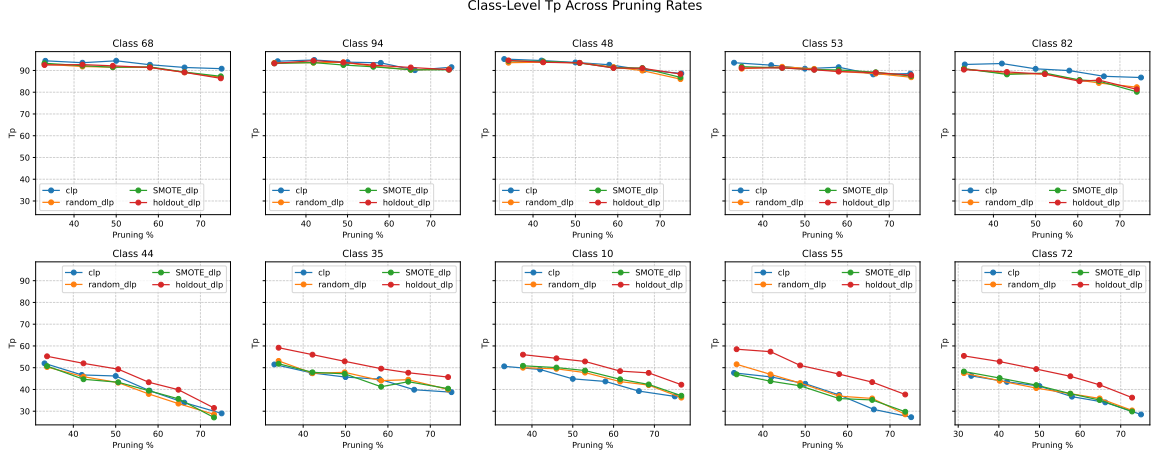


Figure A.6: True positives across classes on CIFAR-100.

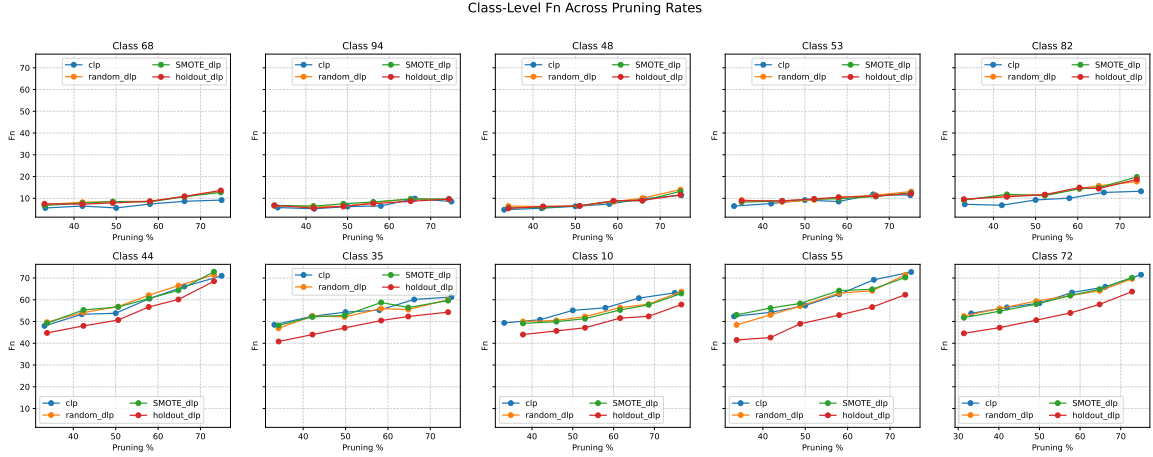


Figure A.7: False negatives across classes on CIFAR-100.

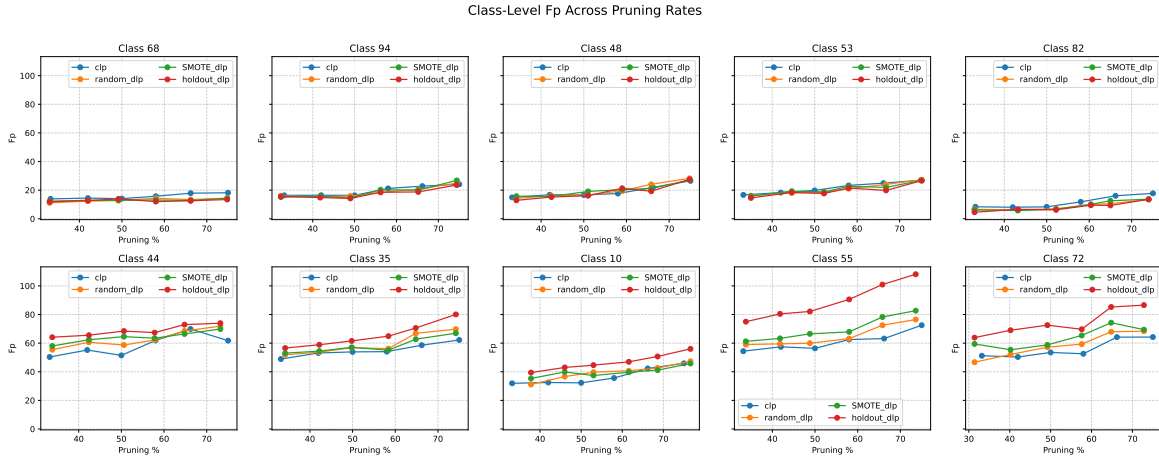


Figure A.8: False positives across classes on CIFAR-100.