Investigating the Impact of Hard Samples on Accuracy Reveals In-class Data Imbalance

Pawel Pukowski[1] **Haiping Lu**[1]

[1]University of Sheffield

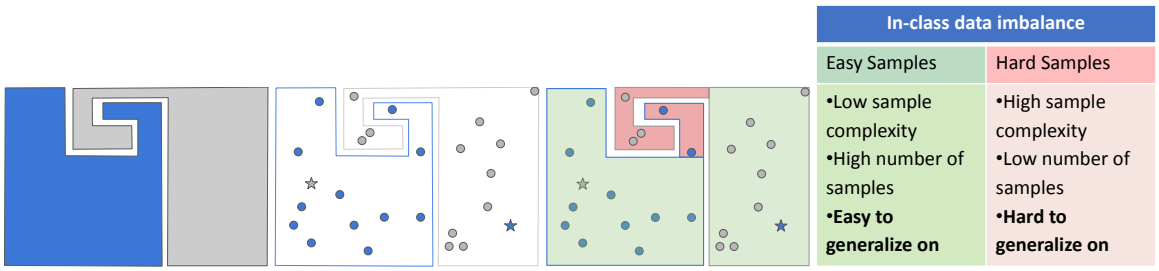
Abstract In the AutoML domain, test accuracy is heralded as the quintessential metric for evaluating model efficacy, underpinning a wide array of applications from neural architecture search to hyperparameter optimization. However, the reliability of test accuracy as the primary performance metric has been called into question, notably through research highlighting how label noise can obscure the true ranking of state-of-the-art models. We venture beyond, along another perspective where the existence of hard samples within datasets casts further doubt on the generalization capabilities inferred from test accuracy alone. Our investigation reveals that the distribution of hard samples between training and test sets affects the difficulty levels of those sets, thereby influencing the perceived generalization capability of models. We unveil two distinct generalization pathways—toward easy and hard samples—highlighting the complexity of achieving balanced model evaluation. Finally, we propose a benchmarking procedure for comparing hard sample identification methods, facilitating the advancement of more nuanced approaches in this area. Our primary goal is not to propose a definitive solution but to highlight the limitations of relying primarily on test accuracy as an evaluation metric, even when working with balanced datasets, by introducing the *in-class data imbalance problem.* By doing so, we aim to stimulate a critical discussion within the research community and open new avenues for research that consider a broader spectrum of model evaluation criteria. The anonymous code is available at `https://github.com/PawPuk/CurvBIM` blueunder the GPL-3.0 license.

1 Introduction

The pivotal role of test set performance in machine learning is undeniable. In AutoML the accuracy on test sets guides the automated selection (Lim et al., 2019; Kerssies, 2023), tuning (Makarova et al., 2022), and evaluation of models (Ericsson et al., 2023), ensuring that the systems developed are both robust and capable of adapting to new, unseen challenges. Yet, this evaluation method is not immune to the challenges posed by data problems, such as dataset imbalance and label noise. Traditional concerns of imbalance refer to disproportionate representation of classes within the data, where one class significantly outnumbers others, which can cause models to perform well on majority classes while neglecting minority ones, misleadingly inflating perceived generalization capabilities (Li et al., 2021; Chawla et al., 2002). Label noise, as highlighted by Northcutt et al. (2021), introduces additional uncertainty, corrupting the test data with incorrect labels, which can significantly affect the perceived model performance (see Figure 1a). In this paper, we shed light on an overlooked aspect that further complicates the reliability of test set performance: *the distribution of hard and easy samples across training and test sets.*

The manifold hypothesis, which states that high-dimensional data tend to occupy lower-dimensional manifolds (Fefferman et al., 2016), provides a framework for distinguishing between easy and hard samples in machine learning contexts. In this context, inference involves disentangling class manifolds (Naitzat et al., 2020), and learning aims at finding the appropriate space transformations for this task. Consequently properties such as intrinsic dimension (Ansuini et al., 2019; Pope et al., 2021), persistent homology (Birdal et al., 2021), curvature (Kienitz et al., 2022) and

 © 2024 the authors, released under CC BY 4.0



(a) Theoretical scenario (b) Real-world scenario (c) Easy-hard regions (d) In-class data imbalance

Figure 1: Let's consider a hypothetical binary classification scenario (a) featuring two distinct class manifolds. The real-world equivalent of this scenario would involve a point cloud non-uniformly sampled from the class manifolds, with some added label noise (stars in b). In this work, we propose that the difficulty of training is derived from the geometrical and topological properties of the class manifolds, leading to areas with higher/lower sample complexity (red/green respectively), due to factors like curvature and homology. Consequently, we observe the emergence of the in-class data imbalance problem (d), which stems from the fact that, although it is more challenging to learn from hard samples because of their sample complexity, the datasets are predominantly composed of easy, not hard, samples.

entanglement (Kaufman and Azencot, 2023) of these manifolds serve as complexity measures. The fact that those characteristics are often local rather than global, implies that the sample complexity differs across each class manifold, resulting in areas that are easier and harder to learn (see Figure 1c). Throughout this work, we conceptualize class manifolds as topological structures populated by samples of a specific class, encompassing both observed samples in the dataset and hypothetical, unobserved ones. Furthermore, drawing inspiration from Narayanan and Niyogi (2009), we define sample complexity as the number of random samples from a given neighborhood within a class manifold needed to reliably classify a new data sample from the same neighborhood into one of the predefined classes with high probability.

Building on this geometric foundation, our work reveals an intriguing observation: *the localised property of sample complexity paired with non-uniform sampling prevalent in real-world scenarios results in an in-class data imbalance within individual class manifolds* (see Figure 1c). In our examination of in-class data imbalance, we observe foundational similarities to classical, between-class data imbalance problems, characterized by the existence of two distinct groups within a dataset: one exhibiting high accuracy on the test set and the other showing poor accuracy. Additionally, we find that there's a notable discrepancy in sample quantities, with the easy group vastly outnumbering the hard group by as much as 13 times on MNIST (Deng, 2012) and 6 times on FashionMNIST (Xiao et al., 2017) and KMNIST (Clanuwat et al., 2018).

Delving deeper, we find that, unlike in the between-class data imbalance problem, the differences in performance are not solely a matter of sample quantity but also reflect the differing complexities between these groups. This is apparent in our experiments: adding 3,250 easy samples boosted accuracy on easy samples in the test set from 47% to 80%, whereas incorporating 5,000 hard samples improved the accuracy on hard samples in the test set from 17% to only 24%, when measured on MNIST. Another distinction between in-class data imbalance and the between-class data imbalance problem arises from the fact that both hard and easy samples originate from the same class, thereby facilitating some level of information sharing between them. Yet, this sharing is uneven and not dataset-consistent, as evidenced by our empirical results: training exclusively on hard samples led to 50% accuracy on easy samples in the test set, whereas focusing solely on easy samples during training resulted in a considerably lower accuracy on hard samples in the test set, at 15% on MNIST. Furthermore, increasing the ratio of hard samples in the training set has a positive effect on the

accuracy of easy samples from the test set on MNIST and KMNIST, *but not on FashionMNIST*, suggesting dataset-dependent level of information sharing.

The existence of in-class data imbalance indicates the need for a more thoughtful strategy in our data practices, from augmentation to dataset compilation and sample division into training and test sets, challenging the status quo of random splitting.

In this regard, our contributions are three-fold:

- **Impact of In-class Data Imbalance**: Through empirical analysis, we demonstrate the existence of two distinct groups within the class manifold, highlighting how the distribution of hard samples between training and test sets impacts model training difficulty and the perceived generalization performance.
- **Identifying Hard Samples**: We adapt the stragglers concept from Ciceri et al. (2024)—samples misclassified at the *inversion point*, where class manifolds transition from expanding to contracting, indicating a shift from learning easy to harder samples—to the multiclass classification setting. This extension provides a nuanced approach to identifying and learning from hard samples, enriching our understanding of sample complexity within diverse classification contexts.
- **Benchmarking Procedure for Hard/Easy Sample Segregation**: We introduce a benchmarking procedure for evaluating methods that segregate hard from easy samples, facilitating a standardized assessment of strategies aimed at addressing in-class data imbalance.

2 Relevant Literature

In machine learning, hardness is contextually defined. In SVMs, hardness is attributed to samples that lie close to the decision boundary, critical for defining the optimal separating hyperplane. As such they utilize geometric margins as a direct measure of sample difficulty (Cortes and Vapnik, 1995). Active learning defines hard samples as those that offer a balance between high uncertainty and representativeness, maximizing learning efficiency and distribution coverage (Budd et al., 2021; Ren et al., 2021).

Curriculum learning and active learning both often rely on uncertainty-based metric to asses the hardness, but due to the different usage of the hardness notion their techniques developed in different directions. Curriculum learning, first introduced by Bengio et al. (2009), employs a structured approach, progressively increasing task difficulty based on the model's current confidence, often utilizing competence-based metrics not typically found in active learning (Wang et al., 2021; Soviany et al., 2022). Active learning, in contrast, selects samples based on expected model change, a metric unique to its methodology, aiming to label those that most improve the model (Settles, 2009; Zhou and Wu, 2023). In hard negative mining, hardness is defined by samples that are incorrectly classified as positive, focusing on misclassification rate and loss values to identify the most challenging examples (Canévet and Fleuret, 2015; Shrivastava et al., 2016). This emphasizes the selection of negatives that contribute most to learning, diverging from the uncertainty-based metrics commonly used in other domains.

In the literature, adversarial examples and anomalies are frequently identified as hard samples (Ho and Nvasconcelos, 2020; Yang et al., 2023). This classification stems from the inherent challenges these samples present in model training and generalization. Adversarial examples, intentionally engineered to exploit model weaknesses, underscore the limitations in current learning algorithms' ability to withstand malicious inputs (Goodfellow et al., 2014; Brendel et al., 2017; Serban et al., 2020). Conversely, anomalies represent outliers within the data distribution, which are naturally occurring yet difficult for models to accurately classify due to their rarity or deviation from the norm (Hodge and Austin, 2004; Chandola et al., 2009; Goldstein and Uchida, 2016).

Our work is inspired by Ciceri et al. (2024), who introduced the notion of *stragglers* in binary classification contexts. They defined stragglers as points misclassified by a model at a specific

phase—when it shifts from decreasing intra-class variability (while increasing inter-class variability) to doing the opposite. Their focus was on establishing the presence of stragglers and demonstrating their effect on generalization. We extend their methodology to multiclass classification settings (Section 3). However, our work diverges significantly from Ciceri et al. (2024) beyond this point, as we focus on studying the in-class imbalance (Section 4), and propose a benchmarking procedure for hard sample identifiers (Section 5).

Our work also bears a close resemblance to research on small disjuncts (Holte et al., 1989; Japkowicz, 2001; Jo and Japkowicz, 2004). These studies suggest that the problem of between-class imbalance is not an issue in itself, but rather that it is accompanied by the problem of small disjuncts. Each disjunct can be considered a submanifold of the class manifold, with small disjuncts corresponding to submanifolds (or neighborhoods) that are sparsely sampled. The sparse sampling often means they are harder to learn, although this depends on their sample complexity. We argue that there is significant merit in rekindling interest in this topic through the lens of sample complexity introduced in our paper. This is because the majority of works on within-class imbalance address the imbalance issues using density-related variables (Anwar et al., 2014; Li et al., 2024). We contend that this approach oversimplifies the problem, and that sample complexity offers a more thorough and robust perspective. We hope this insight will generate increased interest in this under-researched area (Leevy et al., 2018).

3 Towards Objective Identifiers of Hard Samples

Reflecting on the diverse interpretations of hardness examined in Sec. 2, it becomes clear that most definitions are tailored to specific contexts within machine learning. This observation underscores an absence of a hardness measure that is objective, rooted in the data itself, and transferable across different learning contexts. To bridge this gap, we extend and refine the methodology introduced by Ciceri et al. (2024), shifting our focus towards a detailed examination of hard regions within datasets that inherently challenge learning algorithms.

Background: Ciceri et al. (2024) document non-monotonous development of the radii of class manifolds in binary classification setting see Appendix A for description of the radii metric). Initially, these manifolds diverge, enhancing intra-class variability while concurrently reducing their radii, effectively minimizing inter-class variability. This phase transitions at a critical juncture that the authors call the *inversion point*, beyond which the trend reverses. This inversion point serves as a marker for identifying *stragglers*, a unique subset of data samples characterized by their misclassification by the model at this pivotal moment. Crucially, Ciceri et al. (2024) also demonstrate that stragglers are nearly universal, showing a substantial overlap between sets identified under different models or optimisation settings. In conjunction with the research of Arpit et al. (2017), which identifies distinct learning paces for easy and hard examples in real datasets, these results support a conjecture of a two-stage learning process: initially concentrating on easy samples before transitioning to a focus on the harder ones.

Multiclass scenario: While Ciceri et al. (2024) laid the groundwork with their analysis, it is crucial to acknowledge that they focused on binary classification. This raises an intriguing question about the applicability and generalization of their findings to the more complex scenario of multiclass classification tasks. To address this gap, we generalize the methodology established by Ciceri et al. (2024) to datasets in their original, multiclass setting. We focus on the radii metric, as calculating it is less computationally intensive than calculating inter-class distances in multiclass setting. We trained an ensemble of 100 Fully Connected Neural Networks (FC NN), using the same experimental set-up as Ciceri et al. (2024), with the critical distinction of utilizing the entire dataset for both training and testing. This approach was deliberately chosen to ensure comprehensive identification of stragglers across the whole dataset rather than a subset. While this method inherently increases the likelihood of overfitting, we maintain that this aspect does not detract from our findings. We posit that the difficulty attributed to hard samples is linked to their data-derived characteristics, a

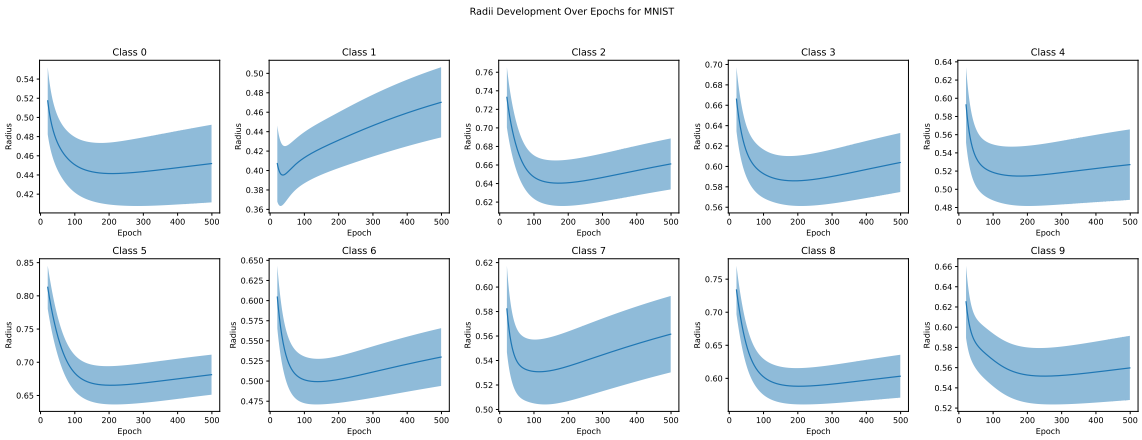


Figure 2: By generalizing the method introduced by Ciceri et al. (2024) to multiclass MNIST, distinct inversion points emerge for each class. This observation signifies that the dynamics of manifolds segmentation are class-specific—a nuance not captured by before due to previous focus on the binary classification setting. The results on other datasets are available in Appendix C.

factor that remains unaffected by the degree of fit between the model and the data.In Appendix B we present a pseudocode and experimental setup.

Analysis: A detailed examination of our results (Figure 2) reveals patterns that corroborate the presence of an inversion point across all class manifolds during training. We find a global minimum for all classes, confirming that the inversion point is a recurring feature in the multiclass classification context. However, this inversion point is not uniform across classes. For instance, in the MNIST dataset, the class manifold for digit 1 reaches this inflection point sooner than others, indicating a class-dependent onset of phase two in the training. We hypothesize that this happens due to the distinct geometrical and topological properties of class manifolds, such as entanglement and intrinsic dimension.

We also notice that the contrast between the radii at the start of training and at the inversion point is markedly greater than what was reported in the binary classification scenarios, and also seems to be class-dependent. This suggests that the process of intra-class contraction and inter-class divergence is more pronounced when dealing with a full spectrum of class labels.

Lastly, while most classes exhibit an inversion point, it is noteworthy that in some instances an inversion point is not evident. Although such cases are rare, they present an important caveat to the universality of the inversion phenomenon and hint at the potential for certain classes to follow alternative learning dynamics that do not conform to the expected pattern. These findings highlight the complexity of learning dynamics at play and reinforces the notion that a *one-size-fits-all approach may not be sufficient when dealing with multiclass datasets* (Cheung and Yeung, 2021; Balestriero et al., 2022).

4 In-class Data Imbalance

Experimental setup: In this section we use stragglers to identify hard samples, an intuitive method now empirically proven to effectively mark complex cases. To obtain a straggler set, we train a model until k inversion points are found, one for each class. These result in k class-level straggler sets, which we later aggregate to form a dataset-level straggler set. We then distribute stragglers across training and test sets in proportions reflecting train-to-test ratios. For each ratio, we create three unique sets of stragglers. We then train three models with distinct initializations for each straggler set. To explore the impact of the quantity of hard and easy samples on the perceived

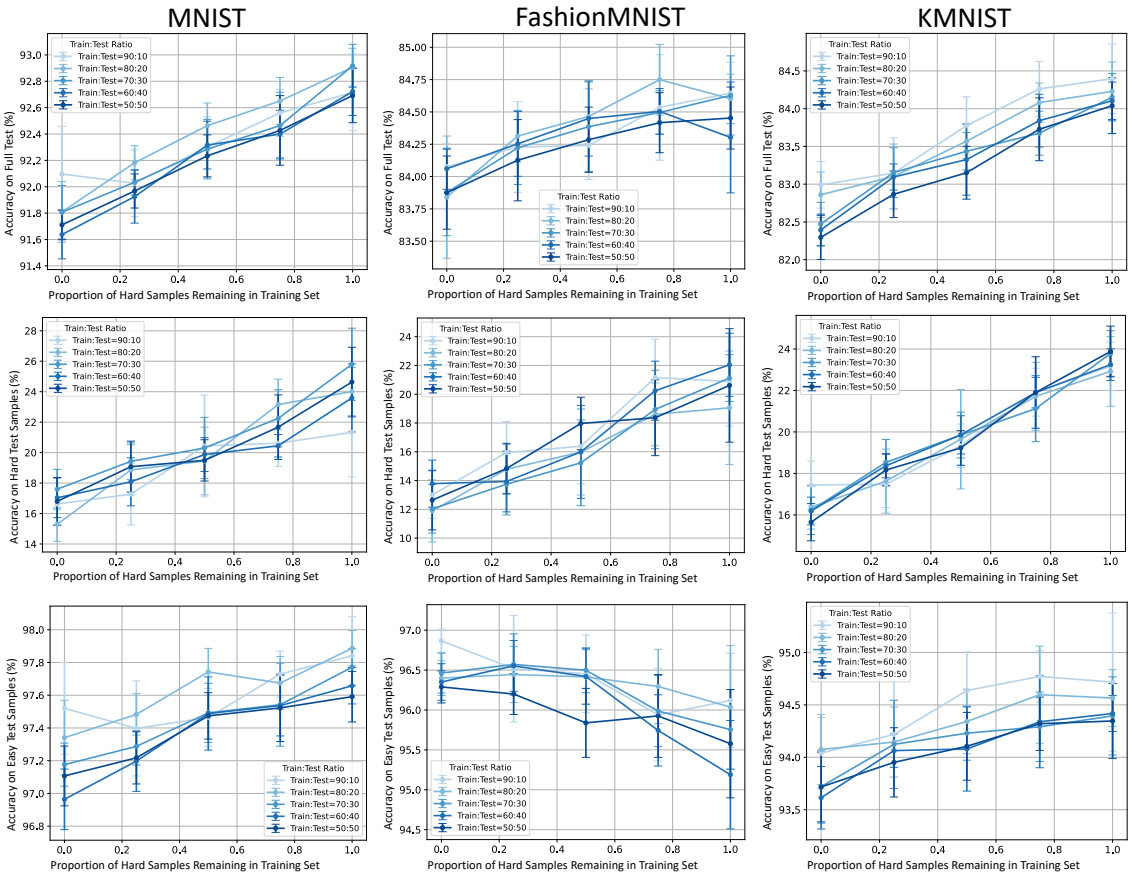


Figure 3: Increasing the proportion of hard samples in the training set improves accuracy on the entire test set (Row 1), but the difference comes mostly from improved accuracy on other hard samples (Row 2), rather than easy samples (Row 3).

generalization of the models, we conduct the experiments multiple times, each time removing a certain percentage of hard (see Fig. 3) or easy (see Fig. 4) samples from the training set. In other words we start with the full dataset, and gradually remove hard (see Fig. 3) or easy (see Fig. 4) samples from the training set. In Appendix D we present a pseudocode and experimental setup.

Similarities to between-class data imbalance: The observed ratios of stragglers to non-stragglers in our datasets—1:6 for FashionMNIST and KMNIST, and a more extreme 1:13 for MNIST—reflect an in-class data imbalance reminiscent of the classical majority:minority dynamic. This similarity extends to the performance on the test set, with hard samples consistently showing lower accuracy (18-28% on MNIST) compared to easy samples (97-98% on MNIST), akin to the disparity seen between majority and minority classes. Moreover, the way to counteract this imbalance appears similar to class imbalance strategies: introducing more samples from the specific group one wishes to improve accuracy on—be it hard or easy—yields the most significant gains. This suggests that *in scenarios where data augmentation is employed, prioritizing the inclusion of hard samples might yield more substantial improvements.* This insight opens avenues for future exploration, particularly in integrating such considerations into Automatic Data Augmentation (Cubuk et al., 2018; Lim et al., 2019).

Nuances emerging from in-class data imbalance: The similarities between in-class and between-class data imbalance problems lead to an inevitable question: Are the approaches commonly used to address between-class data imbalance applicable to the in-class data imbalance problem? As

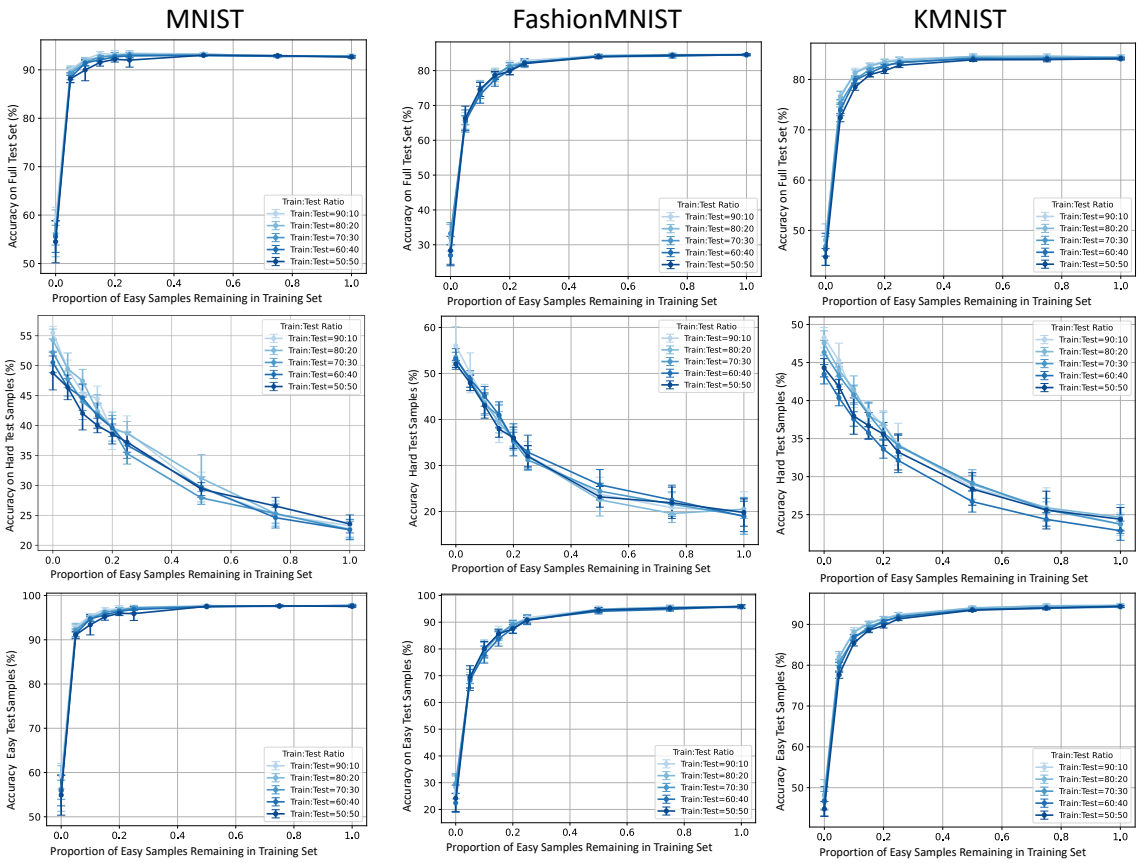


Figure 4: Increasing the proportion of easy samples in the training set improves accuracy on easy samples in the test set (Row 3), resulting in increased overall accuracy (Row 1), while decreasing the accuracy on hard samples in the test set (Row 2), which resembles results we would get when adding samples from the majority class in between-class data imbalance problem.

an example, let's consider the F1 score, which is arguably the most popular metric used when working with multiclass imbalanced datasets. The F1 score primarily evaluates the balance between precision and recall, effectively addressing between-class imbalances by penalizing models that overly favor one class over the others. However, this metric assumes uniform difficulty within classes, a presumption that overlooks the nuanced reality of in-class data imbalances where samples vary significantly in complexity. The F1 score, while indicating overall balance in datasets like MNIST, fails to capture these subtleties. This observation is supported by our results, which we detail in Appendix E.

The initial distinction between in-class and between-class data imbalance arises from the fact that both hard and easy samples originate from the same class, inherently sharing a level of information. However, our experiments show that this information is unevenly shared. We notice that training solely on hard samples considerably boosts accuracy on easy samples (50-61% on MNIST). Conversely, training with only easy samples yields markedly lower accuracy on hard samples (14-19% on MNIST). These results become even more surprising when we consider that hard samples constitute a minority within the dataset.

The second significant observation from our experiments underscores the role of sample complexity in learning performance. Remarkably, a modest pool of merely 3,000 easy samples is sufficient to elevate the accuracy on easy samples, from 50-61% to 90-93% on MNIST. This

enhancement starkly contrasts with the marginal gains observed when incorporating all 5,000 hard samples on MNIST (improvement from 14-19% to 18-28%). The impact of sample complexity becomes even more apparent when we study the curves in Figure 3c, with the function being the most smooth for FashionMNIST, which is the most complex of the tested datasets.

We also find that the impact of adding hard samples on the accuracy on easy samples varies across datasets, suggesting that the level of information that hard samples have on easy samples might differ from dataset to dataset. We see that while MNIST and KMNIST benefit from the inclusion of hard samples, *FashionMNIST exhibits a decline in the performance*, possibly due to its higher complexity or more prevalent label errors. Interestingly, the train:test ratio has no impact on our results. This suggests that hard and easy samples exhibit distinct generalization capabilities, with a certain amount of shared information between these groups that appears to be dataset-dependent.

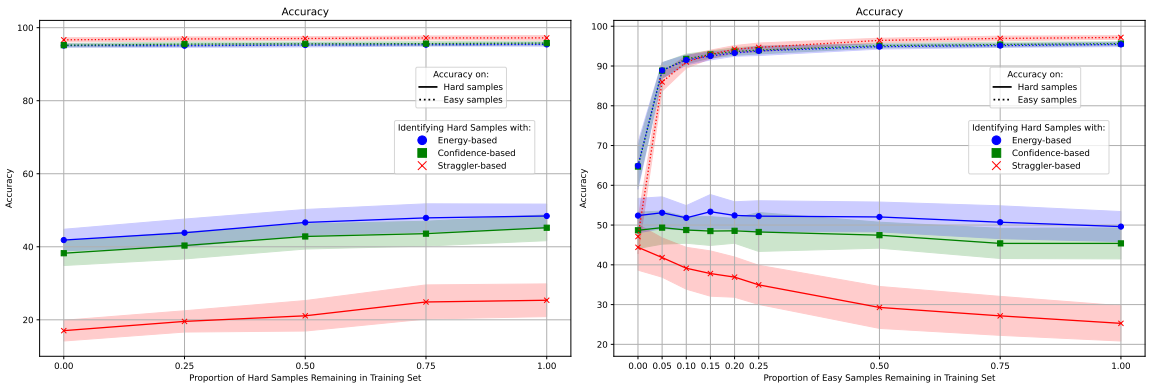
Finally, we investigated the distribution of hard samples across classes, our findings (Figure 8) reveal a non-uniform distribution strongly correlated with class-level accuracies. In fact, the Pearson correlation coefficient between the class-level error rates and the class-level distribution of hard samples is above 0.9 for all hard sample identifiers and datasets used in our paper (see Appendix E for more information). Furthermore, variations in observed in-class data imbalance, based on the choice of hard sample identifier, are attributed to the significant overlap among hard samples. Specifically, there is an 87.42% ± 1.09 overlap between the hard samples identified by confidence- and energy-based methods. This overlap reduces to 45.67% ± 3.07 when comparing straggler- and energy-based methods, and to 49.48% ± 3.04 between straggler- and confidence-based methods. This explains the similarities between the results obtained via energy- and confidence-based methods, and the stark differences from the results obtained via straggler-based method.

5 Benchmarking Procedure for Hard Sample Identification Methods

Motivation: After identifying the existence of the in-class data imbalance our objective shifts to introducing a benchmarking procedure for the methods that identify hard samples. The knowledge from the between-class data imbalance suggests that the better the technique for identifying hard samples the larger the observed in-class data imbalance. This is because hard samples are harder to learn than easy samples, and we should see lower accuracy on hard samples than easy samples. To verify if this theory-based scenario happens in practise we explore both confidence-based and energy-based approaches as hard sample identifiers, as they are widely recognized for estimating sample difficulty in curriculum learning and for detecting anomalies. Our objective is to contrast these approaches with the more objective, dataset-level, straggler-based approach.

Benchmarking setup: Both confidence-based and energy-based methods employ a predetermined threshold to identify hard samples, clearly differentiating them from the straggler-based approach, which does not require such a threshold and thus avoids the bias introduced by it. However, to make the results more comparable, we need to establish this threshold using the straggler count. After determining the threshold, we train neural networks following the setup described in Sections 3 and 4, and extract the most uncertain data samples using either confidence or energy as the metric of uncertainty. These 3 sets of hard and easy samples are then used to re-evaluate the degree of in-class data imbalance on MNIST, as illustrated in Fig. 5.

Strengths of straggler-based method: After combining the results of the performance on easy samples with those on hard test samples in Figure 5, the characteristics of in-class data imbalance become even more apparent. Across all identification methods used, we notice that: 1) models perform better on easy than on hard samples; and 2) to achieve the largest accuracy improvement for either easy or hard samples, we need to incorporate more of the corresponding samples into the training set. Both observations closely resemble the between-class data imbalance as thoroughly described in the previous section. However, we find that the degree of this in-class data imbalance can become less or more pronounced, depending on how we identify hard samples. For example,



(a) Increasing the number of hard samples in the training set

(b) Increasing the number of easy samples in the training set

Figure 5: The in-class data imbalance becomes bigger when we combine the results of the performance on easy samples with those on hard samples in a single figure. It becomes clear that models achieve significantly better accuracy on easy samples than on hard samples, a discrepancy also observed in the between-class data imbalance between majority and minority classes. This in-class data imbalance can become either less or more pronounced depending on how successfully we manage to divide the dataset into hard and easy samples. The overlap between hard and easy sample sets, due to less precise identification, leads to a lower in-class data imbalance, as observed when comparing the accuracies of straggler-based methods with those of confidence-based or energy-based methods. At the top, green (confidence-based) and blue (energy-based) lines overlap each other at this scale.

using confidence-based and energy-based approaches to identify hard samples makes the gap between the accuracy on easy and hard samples appear smaller than what the straggler-based method suggests. This implies that hard samples identified by the confidence-based and energy-based methods contain a high percentage of easy samples, making them appear to perform better on hard samples. Furthermore, using these methods makes it seem like training on hard samples alone allows the model to generalize to over 70% onto easy samples. However, the clearest indication that confidence-based and energy-based methods are poor at identifying hard samples is the fact that they make it seem like adding easy samples to the training set, which constitute the majority of class manifolds, has a negligible effect on performance on hard samples, a conclusion completely contradictory to common knowledge from between-class data imbalance. Consequently, we posit that this variation in the degree of in-class data imbalance is crucial information that can be used to benchmark methods that identify hard samples. This idea stems from the resemblance of the in-class data imbalance to the between-class data imbalance thoroughly described in Sec. 4. The better we identify hard and easy samples, the more pronounced we can expect the in-class data imbalance to be.

Distinction between hard samples and anomalies: In our analysis, the expected performance disparity between confidence- and energy-based methods, well-documented in the context of out-of-distribution and anomaly detection (Liu et al., 2020), does not manifest in the realm of hard sample identification. This observation underscores that residing in a low-density region does not inherently make a sample difficult to classify, nor does presence in a high-density area guarantee ease of classification. The essence of this insight lies in the divergence between anomaly detection, which often equates low density with being an anomaly, and the geometric complexities involved in learning from class manifolds. Samples located in geometrically simpler, yet sparsely populated areas of the manifold, may not present significant classification challenges, prompting us to reevaluate our understanding of what constitutes a hard sample in machine learning. This revelation

underscores the necessity of distinguishing between anomalies and hard samples, advocating for methodologies that are attuned to the nuanced geometry of data distributions.

6 Broader Impact Statement

Our methodology emphasizes the importance of recognizing and addressing in-class data imbalance, fostering the development of AI systems that are more accurate and equitable. By enhancing model sensitivity to hard samples, we aim to reduce bias in AI predictions, underscoring that not all data are created equal. Focusing on hard samples for tasks such as fine-tuning and data augmentation could not only improve performance but also decrease computational complexity due to the reduced data size needed, thereby contributing to a reduction in the environmental impact associated with these tasks.

7 Future Work

Our work suggests directions for future research:

- **Expanding to Imbalanced Datasets**: Generalizing our methods to imbalanced datasets is crucial, as accuracy is less informative there.
- **Clarifying Sample Hardness**: We posit that hard samples are in complex class manifold regions. A deeper investigation into the nature and origins of sample complexity is needed.
- **Leveraging Between-Class Tools**: Considering between-class imbalance as an unsupervised variant of in-class imbalance, adapting existing imbalance correction tools for in-class challenges warrants exploration.

8 Conclusion and Limitations

This study extends the foundational work of Ciceri et al. (2024), illustrating that the dynamic radii changes of class manifolds, indicative of non-monotonous learning progression, persist in multiclass classification scenarios albeit with increased instability. Through the segmentation of MNIST, FashionMNIST, and KMNIST datasets into easy and hard samples, we reveal the pivotal role of hard sample distribution in modulating training and testing difficulty, thereby challenging the conventional reliance on test accuracy as a primary indicator of model generalization. Our investigation underscores the distinction between performance on easy versus hard samples. Finally, we introduce novel and intuitive benchmarking procedure for evaluating the effectiveness of various methods in identifying hard samples.

While our findings shed light on important aspects of sample difficulty and generalization, we acknowledge several limitations. The theoretical basis of our analysis heavily leans on the manifold hypothesis, which may not fully encompass the intricacies of data geometry in complex datasets. The simplicity of the models and datasets employed in our study, while facilitating a clearer understanding of foundational dynamics, may not accurately reflect the challenges present in more sophisticated machine learning tasks. Furthermore, the benchmarking procedure we introduce is not a comprehensive benchmark, but rather a set of guidelines that enable the comparison of hard sample identifiers.

In conclusion, this paper does not advocate for a specific solution but rather sheds light on the limitations of the widely utilized metric of test accuracy in ML and AutoML. It is our hope that subsequent research will be directed towards uncovering more effective methods for detecting hard samples and gaining a clearer insight into the disparate generalization types, ultimately leading to the development of models tailored to address these challenges.

References

Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. (2019). Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Anwar, N., Jones, G., and Ganesh, S. (2014). Measurement of data complexity for classification problems with unbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(3):194–211.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Balestriero, R., Bottou, L., and 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., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Birdal, T., Lou, A., Guibas, L. J., and Simsekli, U. (2021). Intrinsic dimension, persistent homology and generalization in neural networks. *Advances in Neural Information Processing Systems*, 34:6776–6789.

Brendel, W., Rauber, J., and Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*.

Budd, S., Robinson, E. C., and Kainz, B. (2021). A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062.

Canévet, O. and Fleuret, F. (2015). Efficient sample mining for object detection. In *Asian Conference on Machine Learning*, pages 48–63. PMLR.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Cheung, T.-H. and Yeung, D.-Y. (2021). Adaaug: Learning class-and instance-adaptive data augmentation policies. In *International Conference on Learning Representations*.

Ciceri, S., Cassani, L., Osella, M., Rotondo, P., Valle, F., and Gherardi, M. (2024). Inversion dynamics of class manifolds in deep learning reveals tradeoffs underlying generalization. *Nature Machine Intelligence*, pages 1–8.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. (2018). Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Ericsson, L., Li, D., and Hospedales, T. (2023). Better practices for domain adaptation. In *International Conference on Automated Machine Learning*, pages 4–1. PMLR.

Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.

Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Ho, C.-H. and Nvasconcelos, N. (2020). Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22:85–126.

Holte, R. C., Acker, L., Porter, B. W., et al. (1989). Concept learning and the problem of small disjuncts. In *IJCAI*, volume 89, pages 813–818.

Japkowicz, N. (2001). Concept-learning in the presence of between-class and within-class imbalances. In *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 14*, pages 67–77. Springer.

Jo, T. and Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1):40–49.

Kaufman, I. and Azencot, O. (2023). Data representations' study of latent image manifolds. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15928–15945. PMLR.

Kerssies, T. (2023). Neural architecture search for visual anomaly segmentation. *arXiv preprint arXiv:2304.08975*.

Kienitz, D., Komendantskaya, E., and Lones, M. (2022). The effect of manifold entanglement and intrinsic dimensionality on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7160–7167.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30.

Li, M., Zhang, X., Thrampoulidis, C., Chen, J., and Oymak, S. (2021). Autobalance: Optimized loss functions for imbalanced data. *Advances in Neural Information Processing Systems*, 34:3163–3177.

Li, S., Song, L., Wu, X., Hu, Z., Cheung, Y.-m., and Yao, X. (2024). Multi-class imbalance classification based on data distribution and adaptive weights. *IEEE Transactions on Knowledge and Data Engineering*.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. (2019). Fast autoaugment. *Advances in Neural Information Processing Systems*, 32.

Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Makarova, A., Shen, H., Perrone, V., Klein, A., Faddoul, J. B., Krause, A., Seeger, M., and Archambeau, C. (2022). Automatic termination for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pages 7–1. PMLR.

Naitzat, G., Zhitnikov, A., and Lim, L.-H. (2020). Topology of deep neural networks. *The Journal of Machine Learning Research*, 21(1):7503–7542.

Narayanan, H. and Niyogi, P. (2009). On the sample complexity of learning smooth cuts on a manifold. In *COLT*.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021). Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., and Goldstein, T. (2021). The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. (2021). A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Serban, A., Poll, E., and Visser, J. (2020). Adversarial examples on object recognition: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 53(3):1–38.

Settles, B. (2009). Active learning literature survey.

Shrivastava, A., Gupta, A., and Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. (2022). Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Wang, X., Chen, Y., and Zhu, W. (2021). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yang, Y., Zhang, C., Zhou, T., Wen, Q., and Sun, L. (2023). Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. *arXiv preprint arXiv:2306.10347*.

Zhou, X. and Wu, O. (2023). Which samples should be learned first: Easy or hard? *IEEE Transactions on Neural Networks and Learning Systems*.

Acknowledgements. This work was supported by the UK Research and Innovation (UKRI) Engineering and Physical Sciences Research Council (EPSRC) under the PhD Scholarship Grant.

Submission Checklist

1. For all authors…

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes

   (b) Did you describe the limitations of your work? Yes

   (c) Did you discuss any potential negative societal impacts of your work? No, but we don't think it's necessary due to the theoretical nature of our work

   (d) Did you read the ethics review guidelines and ensure that your paper conforms to them? `https://2022.automl.cc/ethics-accessibility/` Yes

2. If you ran experiments…

   (a) Did you use the same evaluation protocol for all methods being compared (e.g., same benchmarks, data (sub)sets, available resources)? Yes. All experiments were run using the same architecture, and the same hyperparameters. We describe all experimental settings in Appendix B and D

   (b) Did you specify all the necessary details of your evaluation (e.g., data splits, pre-processing, search spaces, hyperparameter tuning)? Yes

   (c) Did you repeat your experiments (e.g., across multiple random seeds or splits) to account for the impact of randomness in your methods or data? Yes

   (d) Did you report the uncertainty of your results (e.g., the variance across random seeds or splits)? Yes, the standard deviation is visible in the figures

   (e) Did you report the statistical significance of your results? No. We have not conducted any tests to measure the statistical significance of our results.

   (f) Did you use tabular or surrogate benchmarks for in-depth evaluations? Not applicable here

   (g) Did you compare performance over time and describe how you selected the maximum duration? Not applicable here

   (h) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes, we have included the used resources and the approximate amount of time in Appendix B and D.

   (i) Did you run ablation studies to assess the impact of different components of your approach? No. We have not tried to see the impact of different architectures on our work. Similarly, we have not run an ablation study on the index of the latent layer used to compute the radii of class manifolds.

3. With respect to the code used to obtain your results…

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results, including all requirements (e.g., `requirements.txt` with explicit versions), random seeds, an instructive `README` with installation, and execution commands (either in the supplemental material or as a URL)? Yes

   (b) Did you include a minimal example to replicate results on a small subset of the experiments or on toy data? Yes

   (c) Did you ensure sufficient code quality and documentation so that someone else can execute and understand your code? We hope we did.

(d) Did you include the raw results of running your experiments with the given code, data, and instructions? No

(e) Did you include the code, additional data, and instructions needed to generate the figures and tables in your paper based on the raw results? The code provided is enough to generate the figures from this paper.

4. If you used existing assets (e.g., code, data, models)…

   (a) Did you cite the creators of used assets? Yes

   (b) Did you discuss whether and how consent was obtained from people whose data you're using/curating if the license requires it? No

   (c) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Not applicable

5. If you created/released new assets (e.g., code, data, models)…

   (a) Did you mention the license of the new assets (e.g., as part of your code submission)? Yes

   (b) Did you include the new assets either in the supplemental material or as a URL (to, e.g., GitHub or Hugging Face)? Yes. We provided link to anonymous github repository.

6. If you used crowdsourcing or conducted research with human subjects…

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? N/A

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? N/A

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A

7. If you included theoretical results…

   (a) Did you state the full set of assumptions of all theoretical results? Not applicable

   (b) Did you include complete proofs of all theoretical results? No proofs are necessary in our paper

A Radii of Class Manifolds

Throughout our paper, we have utilized the squared gyration radii of the class manifolds to identify the inversion points. We have adapted the formula from Ciceri et al. (2024) for the multiclass setting as follows:

$$R_i^2(t) = \frac{1}{2n_i^2} \sum_{x,y \in \mathcal{M}_i(t)} \|x - y\|^2,$$

where $\mathcal{M}_i(t)$ is the internal representation of the manifold of class i at epoch t, and $n_i = |\mathcal{M}_i(t)|$ is the number of elements from class i (for all k labels in the dataset). This metric effectively captures the essence of intra-class variability by summing the distances between all point pairs in the feature space, with a larger squared gyration radius indicating more dispersed points.

B Delving into Experiment from Section 3

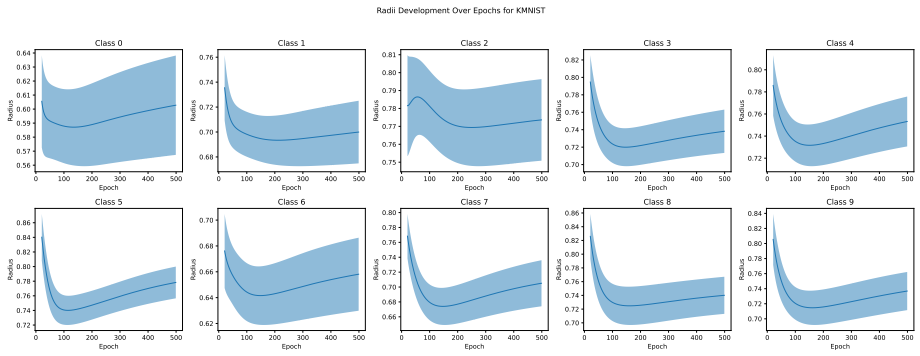
For training, we use FC ReLU NN with a configuration $784 - 20 - 20 - 10$, where 784 is the input dimensionality for MNIST, KMNIST, and FashionMNIST datasets. The radii of class manifolds are computed using their latent representations from the first hidden layer. For training, we employ the Adam optimizer with a learning rate of 0.001 and weight decay of 0.01, alongside cross-entropy loss. The training is conducted over 500 epochs. When performing experiments on the entire dataset, we combine the train and test sets, shuffle the obtained full dataset, and normalize it before training. For subset analyses, we extract a subset after shuffling and then normalize this subset. We have added a `compute_radii` flag to expedite training when radii computation is not required (see line 8 of the pseudocode below for reference). Radii computation starts after epoch 20, as results prior to that are highly chaotic, a phenomenon also observed by Ciceri et al. (2024). Figures 2 and 6 represent the mean and standard deviation over 100 runs. Our implementation is based on PyTorch (Paszke et al., 2019). The experiments were run on a 12-core CPU of a MacBook Pro, taking $3 - 4$ hours on each dataset.

Algorithm 1 Replicate Inversion Results on Multiple Classes

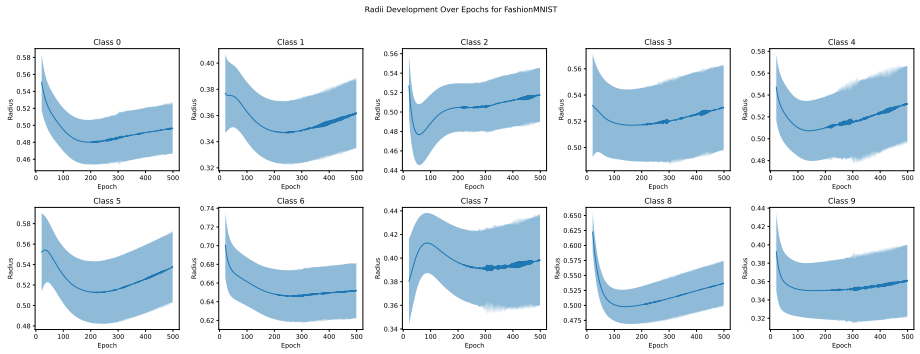
```
1:  Input: model, loader, optimizer, compute_radii
2:  Output: epoch_radii
3:  Initialize epoch_radii as an empty list
4:  for each epoch from 1 to 500 do
5:      for each batch (data, target) in loader do
6:          Perform forward and backpropagation
7:      end for
8:      if compute_radii is true AND epoch > 20 then
9:          Compute current_radii of each for the class manifolds
10:         Append tuple (epoch, current_radii) to epoch_radii
11:     end if
12: end for
13: return epoch_radii
```

C Generalization of Stragglers to Multiclass Classification on Other Datasets

We repeated the experiments from Section 3 on FashionMNIST and KMNIST to validate whether the inversion point observed in our previous studies also appears when working with other datasets (see Figure 6). The results confirm that the phenomena observed in Section 3 are not exclusive to the MNIST dataset. An interesting difference, however, is observed upon inspecting the results on FashionMNIST: we noticed that the radii start to behave chaotically after a certain



(a)



(b)

Figure 6: We see the same phenomena on KMNIST (a), and FashionMNIST (b), as we did in Sec. 3, with the radii behaving very chaotically during the later stages of learning on FashionMNIST (most likely due to it's increased complexity in comparison to MNIST and KMNIST).

period of training. We hypothesize that this behavior stems from the increased complexity of the FashionMNIST dataset; however, further experiments are required to conclusively determine the cause of this trend.

In their work, Ciceri et al. (2024) presented evidence of the inversion point phenomenon using both epoch and test error as variables. However, unlike their approach, we conducted our experiments across the entirety of the datasets. Our aim was not to discover the inversion point phenomenon but rather to utilize it for identifying all hard samples within the datasets. Consequently, we report our results using only epoch as a variable.

D Delving into Experiment from Section 4

We employ the same experimental setup as described in Appendix B. Our aim is to ensure that our results are invariant to model initialization and the obtained straggler set. Therefore, we generate three straggler sets, and for each of these sets, we measure the generalization with three networks, each with a distinct initialization. We then compute the mean and standard deviation over these nine runs to obtain Figures 3 and 4. To identify a straggler set, we train a model and compute the radii of class manifolds after every epoch. Upon detecting an inversion point for a specific class, we save the parameters into a dictionary. This process results in k models, one for each class (see pseudocode below), that define k class-level straggler sets. These are later aggregated to form dataset-level stragglers. If an inversion point is not found within 500 epochs, we rerun the `train_stop_at_inversion` with a new model initialization. The experiments were conducted on NVIDIA A100 GPUs and took between $4 - 8$ hours when working on subsets consisting of $20,000$ samples and performing 25 runs - 5 straggler sets; 5 initializations per straggler set (Figures 5, 7 and

8). The computation time increases to approximately $8 - 16$ hours when running the experiments on all $70,000$ samples, but with only 9 runs - 3 straggler sets; 3 initializations per straggler set (Figures 3 and 4).

Algorithm 2 Train Stop at Inversion

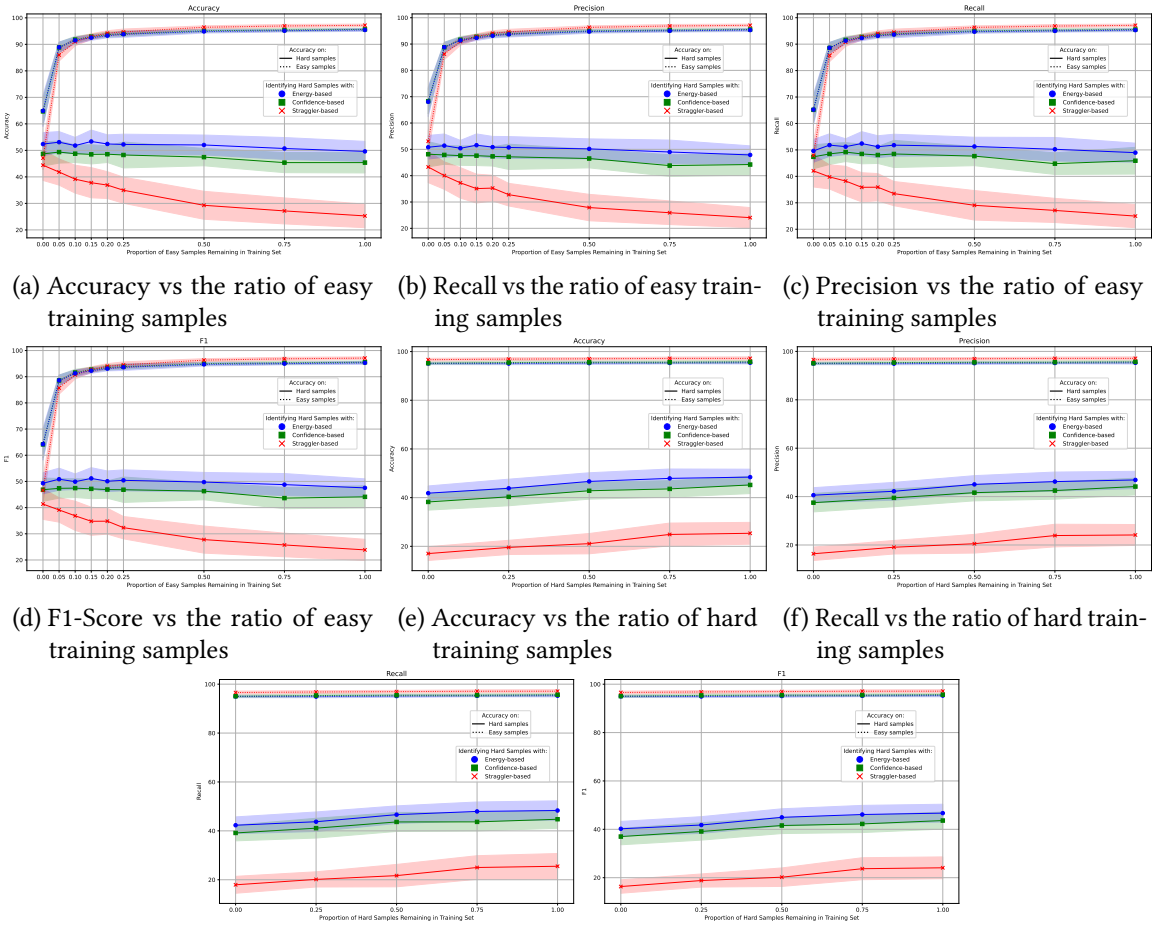
```
Input: model, loader, optimizer
Output: models, inversion_points
Initialize prev_radii with infinity for each class index
Initialize models as an empty dictionary
Initialize found_classes as an empty set
Initialize inversion_points as an empty dictionary
for each epoch from 1 to 500 do
    for each batch (data, target) in loader do
        Perform forward and backpropagation
    end for
    if epoch modulo 5 is 0 then
        Compute current_radii of each for the class manifolds
        for each class index key in current_radii do
            if key not in models AND current_radii[key] > prev_radii[key] AND epoch > 20 then
                Save current model state to models at key
                Add key to found_classes
                Record epoch as inversion point for key
            end if
        end for
        Update prev_radii with current_radii
    end if
    if all class manifolds have found models then
        Break the loop
    end if
end for
return models, inversion_points
```

E Investigating Other Metrics and Hard Samples

In this Section we attach the results of further experiments that didn't make it to the main text due to the page limit.

As described in Section 4, we first measure the value of the additional information obtained by switching from accuracy to precision, recall and F1-score. Due to the balanced nature of the used datasets we do not see any additional valuable information, as all the metrics are closely correlated to each other. We can see that when looking at Figure 7. However, this would definitely not be the case if we moved to class-imbalanced datasets. Over there we expect that accuracy would no longer grant us enough information to be able to apply our benchmarking procedure without incorporating F1-score.

We also measure the distribution of hard samples across the classes. We find that the distribution of hard samples between classes is not uniform (as we show in Figure 8). We suspect that the main reason behind it is the fact that not all classes are equally easy to learn. Running simple experiments on MNIST, KMNIST and FashionMNIST, in which we train 100 networks (using the same training configuration and hyperparameters as in previous experiments), shows that there



(a) Accuracy vs the ratio of easy training samples

(b) Recall vs the ratio of easy training samples

(c) Precision vs the ratio of easy training samples

(d) F1-Score vs the ratio of easy training samples

(e) Accuracy vs the ratio of hard training samples

(f) Recall vs the ratio of hard training samples

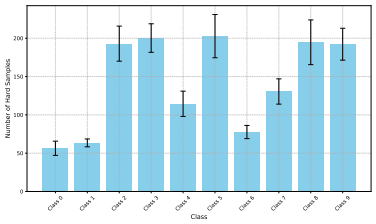
(g) Precision vs the ratio of hard training samples

(h) F1-Score vs the ratio of hard training samples

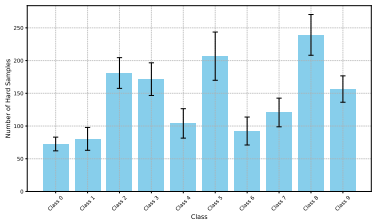
Figure 7: We reran the experiments from Section 4, this time also computing precision, recall, and F1 Score. We discovered that the metrics typically used to address between-class data imbalance are ineffective for investigating in-class data imbalance. We observed that both asymptotic precision and recall exhibit similar values. Figures a-d (e-h) display the results of modifying the ratio of easy (hard) samples in the training set.

exists a very strong correlation between the class-level number of hard samples and class-level error rate. Performing the Pearson correlation analysis backs those visual observations. We find a correlation of 0.9507 for straggler-based methods, 0.9381 for confidence-based methods, and 0.9075 for energy-based methods for the MNIST dataset. For KMNIST and FashionMNIST the correlation is 0.9577 and 0.9856, respectively.

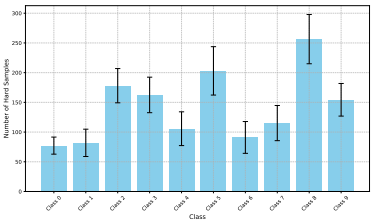
Finally, we also investigate how largely the hard samples identified by different methods overlap. We find that there is an 87.42% ± 1.09 overlap between the hard samples identified by confidence- and energy-based methods. This overlap reduces to 45.67% ± 3.07 when comparing straggler- and energy-based methods, and to 49.48% ± 3.04 between straggler- and confidence-based methods. This explains the variations in observed in-class data imbalance. This also explains why Figures 8b and 8c are so closely correlated to each other, and why both of them differ so much from the results in Figure 8a.



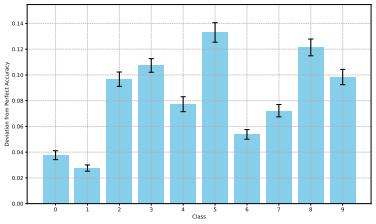
(a) Distribution of hard samples among classes obtained via straggler-based method on MNIST



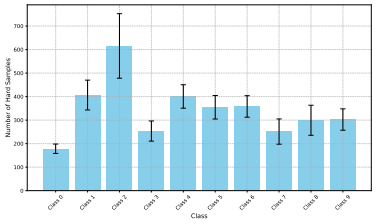
(b) Distribution of hard samples among classes obtained via confidence-based method on MNIST



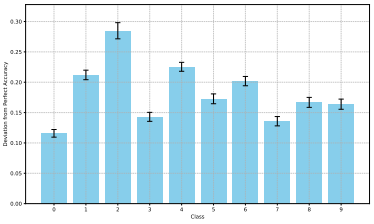
(c) Distribution of hard samples among classes obtained via energy-based method on MNIST



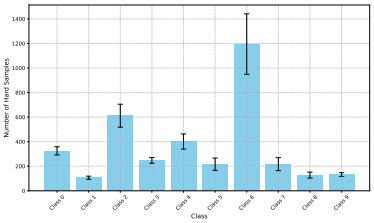
(d) Class-level accuracies on MNIST



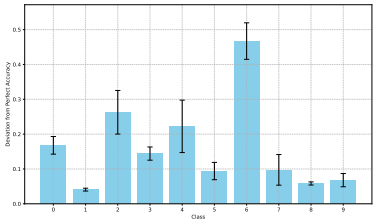
(e) Distribution of hard samples among classes obtained via straggler-based method on KMNIST



(f) Class-level accuracies on KMNIST



(g) Distribution of hard samples among classes obtained via straggler-based method on FashionMNIST



(h) Class-level accuracies on FashionMNIST

Figure 8: Examining Figures 2 and 6 leads us to the natural conclusion that the dynamics of class manifolds vary from one class to another. As a result, we can anticipate that, even though the datasets we have studied are balanced (meaning there is no between-class data imbalance), the distribution of hard samples among the classes is not uniform. This phenomenon occurs regardless of the method used for identifying hard samples (a-c). However, we find that confidence-based (b) and energy-based (c) methods yield a very similar distribution of hard samples, which diverges from that obtained via the straggler-based approach. The close resemblance between methods (b) and (c) arises from an 87.42% ± 1.09 overlap between the hard samples identified by confidence- and energy-based approaches (averaged over 15 runs). This overlap decreases to 45.67% ± 3.07 for straggler- vs energy-based approaches and 49.48% ± 3.04 for straggler- vs confidence-based approaches. One reason behind the non-uniform distribution of hard samples among classes is that not all classes are equally easy to learn. To demonstrate this, we conducted a simple experiment where we trained a network (using the same hyperparameters as in previous experiments) 100 times on MNIST and reported the class-level accuracies in (d). We also present the distribution of hard samples obtained with the straggler-based approach, and the distribution of class-level accuracies on KMNIST (e-f) and FashionMNIST (g-h). Performing the Pearson correlation analysis reveals a correlation of over 0.9 between the class-level error rates and class-level distribution of hard samples.