Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science
Dept. of Computer Engineering and Microelectronics
Remote Sensing Image Analysis Group

Sample-based Hardness Scores in Deep Learning Curricula: A new Perspective on Learning with Noisy Labels

Master of Science in Computer Science

12th of January, 2022

Tom Burgert

Matriculation Number: 0405030

Supervisor: Prof. Dr. Begüm Demir

Advisor: Dr. Mahdyar Ravanbakhsh

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, 12.01.2022

T.

Tom Burgert

Abstract

Large auto-generated datasets with imperfect labels have sparked interest in developing methods for training deep neural networks (DNN) robust to label noise. In this context, this thesis hypothesizes that the distinction of hard and noisy-labeled samples plays a crucial role for state-of-the-art performance. It follows the common assumption in deep learning that more diverse samples are harder to learn from, but are of higher informative value for generalization. In order to identify globally hard samples, the thesis establishes a notion of sample difficulty that is derived from implicit learning orders in DNN curricula. Considering the difficulty of a sample, learning dynamics under label noise are examined. Further, as a necessary condition, the relation of sample difficulty and informativeness is investigated. Experimental results show an absence of correlation between the two measures for standard benchmarks in noise robust training, CIFAR-10 and CIFAR-100 and a non-linear correlation for TinyImagenet. As a consequence, the central hypothesis is partially rejected. In fact, it is concluded that the identification of easy and thus, less noisy samples is sufficient for the development of powerful noise robust methods. The findings are incorporated into label noise research by illustrating a powerful approach towards state-of-the-art performance by training a semi-supervised method with clean, rather easy samples.

Zusammenfassung

Große automatisch generierte Datensätze mit imperfekten Annotationen haben Interesse an der Entwicklung von Methoden zum Training von tiefen neuronalen Netzen geweckt, die gegenüber inkorrekten Labels robust sind. In diesem Zusammenhang stellt diese Masterarbeit die Hypothese auf, dass die Unterscheidung zwischen schwierigen und inkorrekt annotierten Datenpunkten eine entscheidende Rolle für die State-of-the-Art Performance spielt. Sie folgt der beim Deep Learning üblichen Annahme, dass vielfältigere Datenpunkte schwieriger zu erlernen sind, aber einen höheren Informationswert für die Generalisierung vom Datensatz haben. Um global schwierige Datenpunkte zu identifizieren, wird in dieser Arbeit ein Begriff der Lernschwierigkeit eingeführt, der von der impliziten Lernabfolge von tiefen neuronalen Netzen abgeleitet ist. Unter Berücksichtigung des Schwierigkeitsgrades eines Datenpunktes wird die Lerndynamik unter teilweise inkorrekt annotierten Daten untersucht. Außerdem wird als notwendige Bedingung der Zusammenhang zwischen der Schwierigkeit eines Datenpunktes und der Informativität untersucht. Experimentelle Ergebnisse zeigen eine fehlende Korrelation zwischen den beiden Variablen für Standard-Benchmarks für Training mit imperfekten Annotationen, CIFAR-10 und CIFAR-100 und eine nichtlineare Korrelation für TinyImagenet. Die zentrale Hypothese wird als Folge dessen teilweise zurückgewiesen. Es wird geschlussfolgert, dass die Identifizierung einfacher und damit weniger häufig falsch annotierter Datenpunkte für die Entwicklung leistungsfähiger Methoden zum Lernen mit imperfekten Annotationen ausreicht. Die Erkenntnisse werden genutzt, um die Entwicklung einer Methode mit State-of-the-Art Performance durch Training eines tiefen neuronalen Netzes per halbüberwachten Lernen mit korrekten, eher einfachen Datenpunkten zu skizzieren. Die Experimente werden mit den geläufigen Computer Vision Benchmarks CIFAR-10, CIFAR-100 und TinyImagenet durchgeführt.

Contents

List of Acronyms

C-Score	Consistency Score
DNN	Deep Neural Network
GMM	Gaussian Mixture Model
H-Score	Hardness Score
k-NN	k-Nearest Neighbors
PD	Prediction Depth
PIE	Pruning Identified Examples
SOTA	State-Of-The-Art
SGD	Stochastic Gradient Decent
SL	Supervised Learning
SSL	Semi-Supervised Learning
SVM	Support Vector Machine
TTA	Test-Time Augmentation
TTDO	Test-Time Drop Out

List of Figures

List of Tables

1 Introduction

With the advent of deep learning, great advances have been made in the fields of computer vision, speech recognition and natural language processing. While its success is strongly tied to theoretical foundations that enabled the development of complex architectures and training strategies, another key to the success of deep neural networks (DNN) lays in its ability to learn from big data archives. However, acquiring manual annotations for them in order to carry out supervised learning tasks remains an expensive procedure in practice. Therefore, more and more annotation processes involve crowdsourcing procedures or rely on automated label generation. While cheaper to obtain, these approaches come with the risk of label noise. For example, non-expert annotators might mislabel samples due to a lack of knowledge, while automated procedures can induce noise due to imprecise mappings or outdated data. As a consequence, there has been rising interest in the development of methods that are robust to label noise in recent years.

A deeper understanding of the task of learning under label noise can be established by taking a closer look at the learning behavior of DNN. Principally, the distributed and hierarchical representation inherent to the architecture of DNN enables a structural learning process that generalizes from the dominant patterns in the dataset [2]. In fact, it has been shown, that DNN follow implicit curricula independent of the order of the data samples presented to the network [69]. Under ideal conditions, e.g., noise-free data, they consistently start to learn from easier more representative samples, while learning from harder (i.e., more difficult) samples only at later stages of the training. At earlier stages of the learning process the networks generalize, while at later stages they start to memorize [2]. Nonetheless, these general observations can be transferred to the field of noisy label research. When training with moderately noisy-labeled datasets, the overfitting on mislabeled samples takes place at later stages of the training process, concurrently with learning harder samples [10] [35].

A shortcoming of the majority of works in the label noise literature is a strong focus on improving benchmark scores on datasets under varying artificial noise injection. In contrast to these well established benchmarks, this thesis aims in providing a broader perspective on the evaluation and development of label noise robust methods by taking into account the implicit curricula of DNN. Build upon the observations on training dynamics of easy, hard and noisy-labeled samples, we postulate the following hypothesis:

Hypothesis 1 (H1): *The key challenge in learning from noisy-labeled datasets lays in the distinction between hard and noisy-labeled samples.*

The central postulate on learning under label noise further relies on two additional hypotheses, namely:

Hypothesis 2 (H2): *Samples can be ranked consistently according to their hardness in implicit curricula of neural networks. In particular, this rank is subset consistent, i.e., has no dataset sampling-bias.*

Hypothesis 3 (H3): *The hardness rank correlates with the informative value that a sample can provide for generalization. Harder samples are less redundant for generalization.*

Hypothesis 2 is a necessary condition in order to justify a particular focus on harder samples. If the initialization or the optimization procedure of the network has a profound influence on the order of learning, no global hard samples can be determined. Another problem for identifying samples that are hard-to-learn is related to the co-occurrence with other samples. The relative hardness (i.e., difficulty) of two samples need to be independent of the set in which they co-occur. Furthermore, only if harder samples are of greater value for generalization (as stated in Hypothesis 3), it is reasonable to particularly identify them for the evaluation and development of new noise robust methods. From a proof of Hypothesis 3, it would follow that the performance of a model suffers, if the memorization of mislabeled samples begins before the learning of hard samples is finished.

In general, the knowledge about hard samples can allow specific experiments (e.g., on loss functions, training curricula, selection procedures) to distinguish hard from noisy-labeled samples. For example, a method that is designed to filter out noisy-labeled samples (e.g., [47]) could be impacted differently with respect to its ability to generalize depending on the amount of easy or hard samples being filtered out together with the noisy-labeled samples. In the following, this thesis investigates the truthfulness of Hypothesis 2 and Hypothesis 3 in order to enable a new perspective for the development of new noise robust methods and provide a more in-depth understanding of the learning dynamics under label noise. The thesis is structured as follows:

Chapter 2 provides a fundamental introduction to the relevant research disciplines in deep learning for examining the postulated hypotheses. At first, literature investigating sample difficulty from a learning perspective of DNN is presented. Thereafter, research directions and state-of-the-art methods in training under label noise are described. Finally, the chapter concludes with a short summary of current trends in semi-supervised learning, that is a training approach that becomes increasingly relevant for the development of noise robust methods.

A brief summary of the datasets and experimental setup used throughout the thesis is given in Chapter 3.

In Chapter 4 the truthfulness of Hypothesis 2 is investigated. Before analyzing training dynamics under label noise, an easy-to-compute metric for describing sample difficulty from the perspective of DNN curricula is introduced. Further, possible training design choices are studied in order to make the metric subset consistent. Afterwards, the consistency of difficulty ranks as well as training dynamics under label noise are studied.

The objective of Chapter 5 is to prove Hypothesis 3. Multiple experiments are conducted in order to understand the informative values that samples of different degrees of difficulty can provide for generalization. In addition, the outcome of the experiments is put into context with Hypothesis 1 and consequences for noise robust learning are discussed.

Chapter 6 illustrates a possible direction of integrating the observations made throughout the experiments of the thesis into label noise research. A final summary of discoveries and an outlook of future work is provided in Chapter 7.

2 Related Work

2.1 Sample Difficulty in Deep Learning Curricula

During the last decade, great advances have been made in improving neural network architectures, regularization techniques and training strategies. However, while these methods constantly become better, there has been rising interest in understanding the learning dynamics of deep neural networks. New research objectives include explaining decisions of neural networks or understanding sample difficulty during the course of learning. In particular, the research field of sample difficulty studies the structures and regularities of datasets with respect to the ability of DNN to learn and generalize from them. In the literature, different properties of neural networks and its respective training procedures are studied in order to serve as a measure of sample difficulty in implicit curricula of neural network. In the following, various notions of difficulty are described.

Training prediction/loss: Based on the assumption that it is beneficial to train neural networks in an order of increasing complexity, the inventors of curriculum learning, Bengio et al. [5], propose to use loss values of samples to determine its difficulty. Following the idea of deriving information from predictions during training, Toneva et al. [65] suggest to use the first iteration in which a sample is learned and remains learned for every subsequent iteration as a notion of sample difficulty.

Hold-out prediction: Aiming on distinguishing prototypical from atypical samples in a dataset, Carlini et al. [9] define the metric *Holdout Retraining* that compares the sample confidences by models that were trained with and without the respective sample. Elaborating these ideas, Jiang et al. [33] propose the *consistency score (c-score)* that describes sample difficulty by the probability of predicting the correct label for a sample that is omitted during training. In particular, the score is composed of the expected accuracy for a hold-out sample for training sets with varying sizes that are subsampled from a given dataset. While resource demanding to compute, the authors also study the correlations of the *c-score* with easy-to-compute proxies such as iteration learned [65], the entropy of the softmax layer of a neural network or feature representation based difficulty rankings. They find that prediction based proxies have higher correlation with the *c-score* than representation based proxies.

High-level feature representation: Concerned with developing noise robust training curricula, Guo et al. [22] rank samples by difficulty according to a density-based unsupervised clustering algorithm that makes use of feature representations extracted from a pretrained network. For each class cluster centers are assigned to samples that have a higher density than neighboring samples while maintaining relatively larges distances to samples with higher densities (other cluster centers). Once established, the distances to the respective cluster centers are then used to compute the difficulty scores.

Network representation: There exists a group of works that define sample difficulty by its hierarchical representation inside a neural network at inference time. For example, Wu et al. [69] develop a policy using reinforcement learning to select the minimal configuration of blocks in a ResNet [26] that is necessary to classify a sample correctly. The number of respective blocks can be interpreted as the sample difficulty. Following a similar approach, Baldock et al. [4] introduce the measure *prediction depth* that aims in summarizing the computational difficulty of making a prediction for a given sample. In particular, they construct k-NN classifier probes from the feature representations in all hidden layers of a neural network for the training set. The prediction depth is determined by the first layer in which the prediction of the k-NN probe coincides with all subsequent layers including the final prediction of the network. They further show high correlations between the *prediction depth* and other statistical learning measures such as uncertainty, confidence and learning speed. Furthermore, Hooker et al. [30] establish a notion of difficulty by applying model pruning. Specifically, they measure the degree of disagreement in the predictions for a sample of a pruned and a non-pruned model. By human evaluation they prove the assumption that *Pruning Identified Examples (PIE)* (high disagreement of predictions) tend to be visually more challenging.

Learning signal: Agarwal and Hooker [1] adopt the variance of gradients during the training procedure of a neural network as a measure of difficulty. In particular, in each iteration the variance of the gradients of a sample with respect to its input pixels is computed. The metric is able to identify clusters of samples with distinct semantic properties such as homogeneous backgrounds and consistent object representations.

Cross machine learning models: In contrast to previous mentioned works, Mangalam and Prabhu [48] compare the learning ability of shallow machine learning models such as SVM [14] or Random Forest [29] with the order of learning in DNN curricula. They show that most samples that can be learned by shallow classifiers are learned by neural networks at earlier stages of the training. Hence, they suggest to derive a notion of difficulty by adopting a cross model perspective that is decoupled from the inductive bias of the neural network.

While multiple viewpoints on sample difficulty in deep learning curricula are established, many of them show a high correlation among each other. By assumption and visual examination, these approaches have in common that difficult samples are related to outliers and memorized samples including samples that depict several objects, contain occlusions or cluttered backgrounds, and are sometimes related to mislabeled samples. Easier samples, in contrast, are understood as prototypical, following standard representations of the objects regarding shape, viewpoint, texture and color.

2.2 Robust Learning under Label Noise

Facing the issues of big data archives, the development of noise robust methods to successfully learn from partially mislabeled data is continuously gaining more relevance in the deep learning community. Generally, noise robust methods can be divided into two groups of approaches that make different assumptions about the data they aim to learn from. Methods belonging to the first of the two groups assume the existence of a clean and trustworthy subset of data. There are various techniques to enable noise-robust learning by exploiting this subset. Hendrycks et al. [28] use the clean data to estimate a noise transition matrix to improve the predictions of a classifier trained on the noisy-labeled data. Lee et al. [39] build class prototypes in feature space from the clean subset of data. During training these prototypes are then compared to the samples from the noisy set resulting in different similarity-based attention weights for the samples. In contrast, Li et al. [43] use the clean data to train an auxiliary model leveraged by a knowledge graph that produces additional soft labels for the noisy-labeled dataset contributing to the standard loss function of the original labels.

The second group of approaches aims to achieve more generalization by not utilizing any clean data and learn directly from the noisy-labeled data. Some approaches in this scenario propose specific training design choices that are inherently noise robust. This is mainly achieved by specialized architectures [59] [61] and noise robust loss functions [20] [76], but can also include semi-supervised data augmentation techniques. Zhang et al. [75], for example, generate additional data by convex combinations of pairs of samples and their labels, leading to more noise robustness during training. However, the majority of approaches belonging to the second group perform some sort of auxiliary task for handling label noise during training. Such tasks usually extend the classic training procedure of a neural network by a specific method or subroutine that is designed to detect and handle noise. These routines either perform a process of re-weighting or discarding noisy-labeled samples [22] [50] [47] [25] [52] [23] [32] [68] [40] or, alternatively, incorporate noise specific information into a standard loss function [24] [43] [31] [44] [41] [72]. The information for noise handling is derived by exploiting prediction values [50] [52] [31] [44] [40] [72], feature representations [22] [24] [43] or additional information from an ensemble of networks. Ensembles of networks may be used in form of collaborative networks [47] [25] [23] [68] [40], student-teacher networks [32] [43] [41] or as a self-ensemble network [50] [41].

For most part, the relevant works published in recent years belong to the second group of approaches. Therefore, the in-depth presentation of noise robust methods focuses on approaches that directly learn from the noisy-labeled data. In particular, Guo et al. [22] design a training curriculum based on complexity clusters. The authors show that most samples with noisy labels get assigned to clusters with higher complexities and hence, only influence the training process at later stages, at which the model has learned the dominant patterns already. Han et al. [24] build multiple prototypes for each class that are used to produce corrected pseudo-labels based on a similarity measure in feature space during the training process. The loss function of the neural network is mutually influenced by the observed and corrected labels. Following the strategy of sample selection, Malach et al. [47] maintain two networks that are being updated only if their predictions disagree. The strategy assumes that correctly labeled hard samples are more likely to produce ambiguous predictions than mislabeled easy samples. Inspired by this work, Wei et al. [68] train two networks with small loss instances only, which are derived from a joint loss to ensure the agreement of both networks. Addressing noise-robustness via loss function design, Huang et al. [31] employ a label refurbishment loss that makes use of the exponential moving average of predictions to progressively correct wrong labels. The ELR algorithm of Liu et al. [44] exploits prediction values from the early learning phase to impose a regularization term to the standard loss to neutralize the gradient of samples with false labels. Furthermore, Li et al. [41] incorporate information on mislabeled samples into the loss function within a meta-learning step. During this phase multiple mini-batches of synthetic noisy labels are generated by a random neighbor label transfer. Each synthetic mini-batch updates a copy of the current model enforcing consistent predictions with a self-ensemble teacher model and meta-updated models at each iteration. Interpreting the noisy labels as a learnable parameter, Yi and Wu [72] propose to update the distribution of the label matrix in each backpropagation step. These continuously refined pseudo-labels are then used as a training target. Further, the loss function is extended by a compatibility loss that enforces the pseudo-label distribution to be of similar kind as the original noisy labels. With the DivideMix framework, Li et al. [40] introduce a multi-stage noise robust method that builds upon the semi-supervised learning technique MixMatch [6]. DivideMix maintains two separate networks that model its per-sample loss distribution with a Gaussian Mixture Model (GMM) [8] in order to split the samples into a labeled set (mostly clean) and an unlabeled set (mostly noisy). Then, the two subsets are forwarded to the respective other network which applies a strategy of label co-refinement to the labeled set and a label co-guessing strategy to the unlabeled set. The strategies are supported by multiple data augmentations. At the last stage, the produced pseudo-labels are incorporated into the semi-supervised MixMatch algorithm that provides the learning signal for updating the parameters of the two networks.

Most state-of-the art methods in noise robust training develop certain aspects of DivideMix further and rely on some sort of semi-supervised learning. For example, Nishi et al. [51] propose to use two augmentation strategies of different magnitudes at the individual stages of DivideMix. While the loss modeling stage for selecting a labeled and an unlabeled set performs better under weak augmentations, the MixMatch algorithm that produces the learning signal profits from stronger augmentations. Another recent example relying on semi-supervision is the enhanced version of the ELR framework [44] that incorporates components from MixMatch such as MixUp data augmentation and weight averaging into the basic ELR algorithm resulting in competitive performance on standard benchmarks.

2.3 Semi-Supervised Learning

Recent success of state-of-the-art noise robust methods have increased the relevance of semi-supervised techniques for learning under label noise. Improving the supervised learning signal by making use of additional unlabeled data, modern semi-supervised learning methods build upon two central concepts, *pseudo-labeling* and *consistency regularization*. Early works in the field propose to use entropy regularization on unlabeled data [21] [53], following the assumption that the decision boundary should be distant to high-density regions of the dataset. In fact, the use of *pseudo-labels* has a similar effect on the decision boundary. Lee [38] uses the argmax of the predictions on the unlabeled data as supervised training targets, if the largest class probabilities fall above a fixed threshold. Recent works further encourage the predictions on unlabeled data to be confident by sharpening the distribution of the predictions [6] [71]. Bachman et al. [3] introduce the idea of *consistency regularization* in form of perturbing the input with Gaussian noise processes and the model with dropout noise [60]. They suggest to regularize the standard loss by a consistency term that assures consistent predictions under stochastic transformations of an unlabeled sample. Sajjadi et al. [54] popularize this approach by replacing the Gaussian noise process by weak data augmentation techniques such as cropping, rotating and flipping. Further, Laine and Aila [36] improve *consistency regularization* by additionally using the exponential moving average of predictions at different epochs. Addressing the problem of delayed updates under a large number of iterations per epoch (i.e., big datasets), Tarvainen and Valpola [64] propose to use an exponential moving average of the model weights instead. In contrast to augmentation based approaches, Miyato et al. [49] conceive a *consistency training* on unlabeled data that elaborates ideas from adversarial training. In order to impose consistency they perturb samples by approximating the most adversarial direction in the input space that the model is sensitive to.

State-of-the-art approaches like UDA [71] or FixMatch [58] apply different magnitudes of augmentations to the two sub-tasks *pseudo-labeling* and *consistency regularization*. While artificial labels are created for high-confidence samples by using weak augmentations, the consistency loss is applied under strong augmentations generated by RandAugment [16] or CTAugment [7]. Approaches like MixMatch [6] and ReMixMatch [7] extend the common semi-supervised training pipeline by MixUp [75] data augmentation. From labeled and unlabeled samples additional data is generated by convex combinations of pairs of samples and their (guessed) labels. Recent generic enhancements of these methods include approaches that propose a curriculum for learning flexible class threshold for the selection of high-confidence samples in the *pseudo-labeling* stage [74] or further regularization by self-supervised representation learning [66].

3 Dataset Description and Experimental Setup

This Chapter provides a brief summary of the datasets used throughout the experiments of the thesis. Further, the hyperparameters used for training the neural networks are described.

Datasets. Methods developed for learning under label noise are usually evaluated on four computer vision benchmarks, two clean datasets with simulated label noise, CIFAR-10 and CIFAR-100 [34] and two real-world datasets with a natural amount of label noise, Clothing1M [70] and WebVision [42]. The clean datasets are favorable for ablation studies, since noise rates can be manually controlled and the handling of noisy labels can be empirically observed. Therefore, the experiments conducted throughout this thesis focus on the CIFAR datasets. Both CIFAR-10 and CIFAR-100 contain 50,000 training images and 10,000 test images of size 32×32. They are composed of 10 respectively 100 balanced classes. Example images of CIFAR-10 are shown in Figure 3.1. Additionally, experiments are carried out on TinyImagenet [37], a simplified version of ImageNet [17], in order to provide results on three different datasets with increasing complexity. TinyImagenet is composed of 100,000 images of size 64×64 divided into 200 balanced classes.

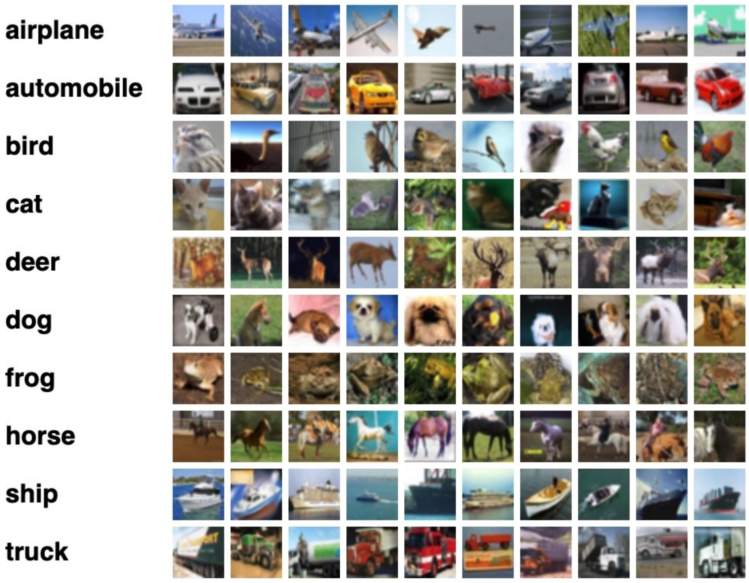


Figure 3.1: Example images from CIFAR-10 dataset [13].

Setup. The experiments in Chapter 4, Chapter 5 and Chapter 6 are implemented with PyTorch. The standard training time of neural networks is set to 30 epochs with a batch size of 128. The semi-supervised methods in Chapter 6 are trained for 250 epochs. When optimized by AdamW [45], the learning rate is set to 1×10^{-3} with a weight decay of 1×10^{-4}. The SGD optimizer is run with a maximum learning rate of 1×10^{-2}, a weight decay of 1×10^{-4} and a momentum of 0.9. The optimization steps are scheduled by an OneCycleLR [57]. Each score is the results of five independent training runs with different random seeds. Data augmentation strategies for CIFAR-10 and CIFAR-100 include horizontal flips and random crops. Samples from TinyImagenet are augmented by AutoAugment [15].

4 Subset Consistent Hardness Score

This chapter aims in establishing a procedure to produce a hardness score (*h-score*) that ranks samples according to their difficulty in implicit curricula of deep neural networks. In order to provide a sufficient supplementary perspective on training under label noise, the *h-score* needs to be subset consistent (see Hypothesis 2). In particular, if the *h-score* is applied to a dataset and a subset of it, samples A and B that occur in both the original dataset and the subset need to be ranked consistently. If sample A is assigned with an *h-score* higher than sample B in the original dataset, the same has to hold true for the subset. In other words, the ranking of two samples needs to be independent of the samples with which they co-occur. A subset consistent *h-score* can be understood as a global difficulty score that has no subsampling bias.

Why is a subset consistent hardness ranking relevant for noisy label training? Due to the lack of diverse datasets that include different proportions of label noise by nature, a common approach for evaluating noise robust methods comprises the injection of synthetic label noise to a clean dataset. Any synthetic injection of label noise produces a clean and a noisy subset of the dataset. A phenomena described in the label noise literature is the tendency of DNN to learn from hard and noisy-labeled samples only at later stages of the training [10] [35]. To better quantify this learning behavior under different noise rates, the existence of a global subset consistent hardness ranking is beneficial. Once the *h-score* ranking is produced for the original (i.e., clean) dataset, any difficult sample can be tracked during the learning process under different noise rates (leading to differing clean subsets), since its relative degree of hardness is subset independent. This can enable detailed studies on how to focus more on learning from hard while neglecting the noisy-labeled samples at later stages of the training and thus, alleviates the development of noise robust methods.

The experiments for establishing a subset consistent hardness score are divided into two parts. For simplicity, the first part conducts experiments on subset consistency by considering clean data only (see Sections 4.1, 4.2, 4.4.) The second part includes further studies of *h-score* consistency under the injection of label noise (see Section 4.5).

4.1 H-Score Definition and Metrics

Inspired by Jiang et al. [33] that propose the *c-score* (see 2.1) as a measure of sample difficulty, we adopt its suggested proxies (1) p_L (softmax confidence on the annotated class), (2) p_{max} (max softmax confidence across all classes) and (3) E (negative entropy of softmax confidences) as a baseline for developing a subset consistent *h-score*. These statistical measures have the advantage over the *c-score* that they are easy to compute and describe sample difficulty from a

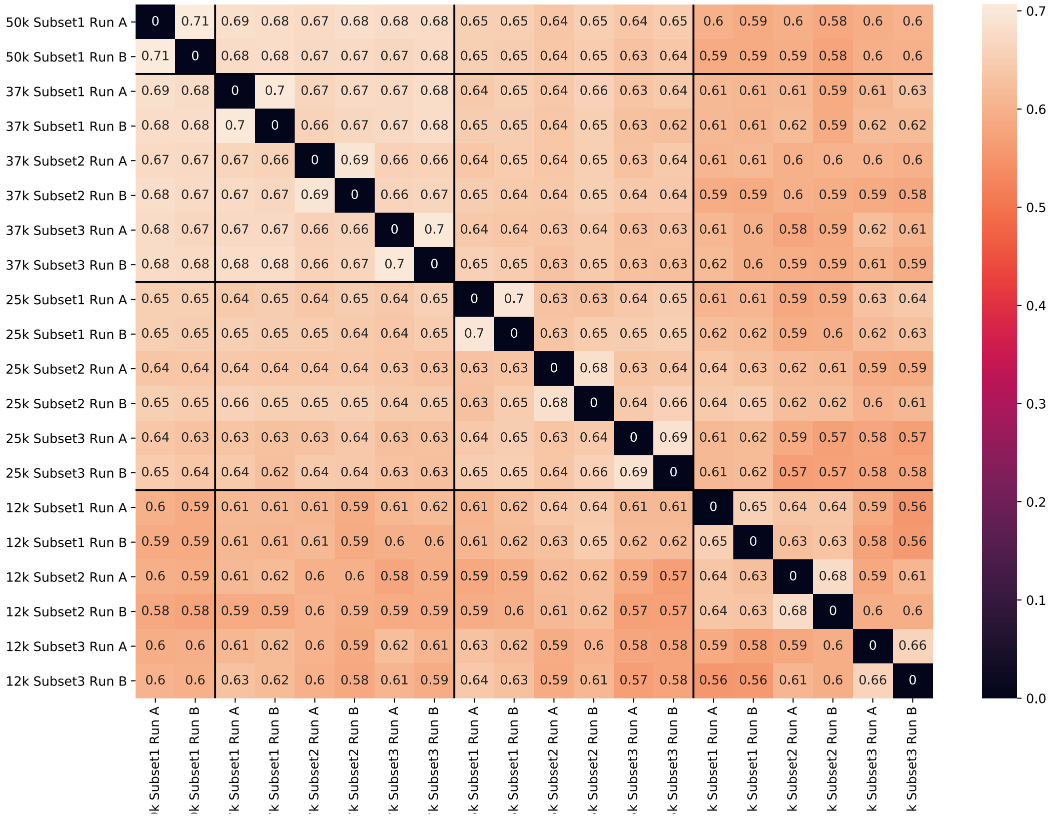


Figure 4.1: *H-score* (p_L) consistency by *hard-intersect* of 20% hardest samples for CIFAR-10 in different training runs (A, B) different subset sizes (75%, 50%, 25%) and three different subset-samplings.

viewpoint of learning velocity due to the course of learning. Concretely, we define

$$hscore : (X,t) \mapsto [0,1]^k, hscore(X,t) = 1 - \frac{\sum_{i=1}^{t} g(X)^{(t)}}{t}, g \in \{p_L, p_{max}, E\} \tag{4.1}$$

where $p_L(X)^{(t)}$ is the vector containing all softmax confidences on the correct class, $p_{max}(X)^{(t)}$ the vector of the maximums of the softmax confidences across all classes and $E(X)^{(t)}$ the corresponding negative entropy of softmax confidences, for samples X at epoch t.

We evaluate our experiments for subset consistent *h-scores* under two metrics. Following the discoveries in label noise research that easy samples are learned at earlier stages of the training while hard and noisy-labeled samples are simultaneously learned at later stages (see Section 1, Section 2.2), consistent ranking of more difficult samples has more relevance than one of easy samples. Therefore, we introduce the metric *hard-intersect* that puts emphasis on subset consistency of difficult samples in order to evaluate the *h-score* rankings. It computes the overlap of the top k hardest samples (according to the *h-score*) that are part of both (sub)sets. In particular, let X and Y be subsets of a dataset D, with $X \cap Y \neq \emptyset$ and $H^X = h(X)$ and $H^Y = h(Y)$ the respective *h-scores*. Let further be $H'^X = \{H_i^X \mid X_i \in X \cap Y\}$ and $H'^Y = \{H_i^Y \mid Y_i \in X \cap Y\}$ the *h-scores* of the samples that are part of both subsets and r a ranking function in descending order. Then:

$$hard\text{-}intersect(k) = \frac{|\{r(H'^X)_i \mid 1 \leq i \leq k\} \cap \{r(H'^Y)_i \mid 1 \leq i \leq k\}|}{k} \tag{4.2}$$

Besides the heuristic metric *hard-intersect*, we also evaluate the *h-score* rankings by a statistical metric that applies the *spearman rank correlation* [18] to all samples that occur in both (sub)sets. It is defined as follows:

$$\rho = 1 - \frac{6\sum_{i=1}^{n}[r(H_i'^X) - r(H_i'^Y)]^2}{n(n^2-1)} \tag{4.3}$$

where n is the number of samples in the intersection of the two (sub)sets, $|X \cap Y|$. Both metrics relate to the principle of macro averaging and are computed by averaging the class-wise scores. A class independent perspective of the hardest samples (micro averaging) would neglect hard samples of easier classes that possibly result in lower *h-scores* than easy samples of hard classes.

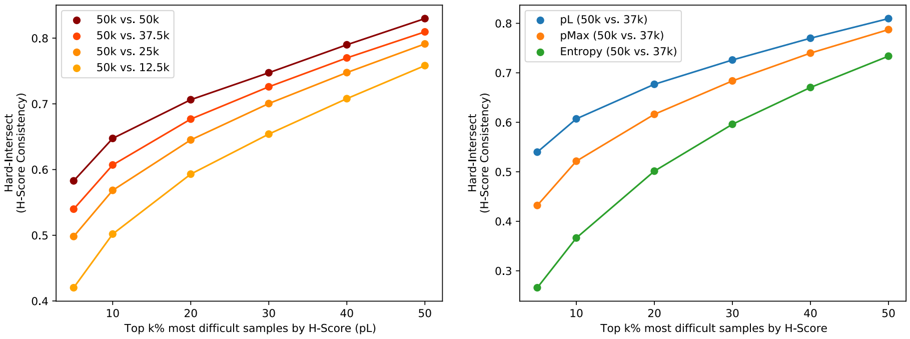


Figure 4.2: **Left:** *H-score* (p_L) consistency by *hard-intersect* for CIFAR-10 dataset and different subset sizes. Scores are averaged by multiple subset-samplings and runs per size. **Right:** Comparison of different proxies for the *c-score* by *hard-intersect* for CIFAR-10 dataset (50k samples) and a 75% (37k samples) subset. Scores are averaged by multiple subset-samplings and runs per size.

Before particular training design choices for improving *h-score* consistency are studied, we compare the three proxies in an initial experiment following a basic training procedure for different training runs, subset sizes and subset samplings for the CIFAR-10 dataset. The results in Figure 4.1 suggest that the subset consistency of the *h-score* (based on p_L) is invariant to subset-sampling and training runs under the *hard-intersect* metric. Further, the *h-score* consistency of a certain subset size is always slightly higher under the same subset sampling (self-consistency) than a different one (subset-consistency). There is a linear relationship between the reduction of the size of one of the two subsets and the subset consistency (see 4.2; left). The same tendencies can be observed when evaluating the consistency by the *spearman rank correlation*. Moreover, as Figure 4.3 emphasizes, there exists a strong correlation between the heuristic and the statistical metric. Figure 4.2 (right) shows that p_L is the proxy with the highest *h-score* consistency. Following the results of the initial experiment, we base our *h-score* on the p_L measure. Due to the linear relationship between reducing the size of a subset and its *h-score* consistency, the experiments for training design choices to produces the optimal subset consistent *h-score* will be only evaluated for the original dataset with itself (self-consistency) and a subset of size 75% (subset-consistency).

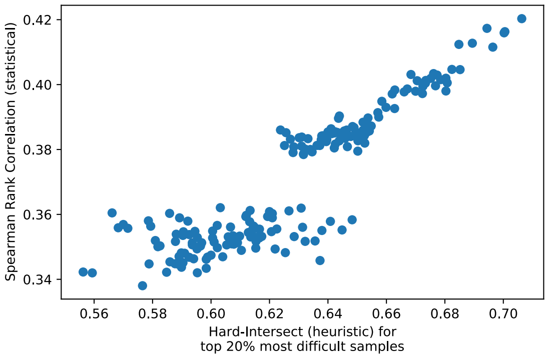


Figure 4.3: Correlation of *hard-intersect* of 20% hardest samples and *spearman rank correlation* for all *h-scores* based on p_L build on subset sizes and samplings of CIFAR-10 presented in Figure 4.1. The correlation coefficient is 0.9.

4.2 Training Design Choices for H-Score Consistency

This section covers experiments with different architectural and training design choices for increasing the subset (and self) consistency of the *h-score*. By making use of the statistical measure p_L for learning speed we use the proxy of the *c-score* [33] that revealed the highest subset consistency.
In the experiments we analyze the following design choices:

- pretraining vs. no pretraining: DNN pretrained on ImageNet [17] or with random weight initialization
- AdamW optimizer [45] vs. Stochastic Gradient Decent (SGD) optimizer: AdamW with weight decay or SGD optimizer with one-cycle scheduler [57]
- different DNN architectures: ResNet18 [26], ResNet50 [26], ResNet101 [26], VGG19 [56], EfficientNetB1 [63], EfficientNetB4 [63], InceptionNetV3 [62]
- data augmentation: with and without standard data augmentations for datasets (cropping, flipping for CIFAR-10/CIFAR-100, AutoAugment [15] for TinyImagenet)
- dropout layers [60]: adding dropout layers to the hidden layers of the neural network with differing rates of 0.1, 0.3, 0.5, 0.7
- test-time augmentation: augmentation applied at multiple inference passes to the train set for obtaining p_L (independent from training, obtained after each epoch)
- test-time dropout: dropout applied at multiple inference passes to the train set for obtaining p_L (independent from training, obtained after each epoch)
- ensembles of networks: use of multiple *h-scores* runs to produce final scores
- cumulative scores until different iterations (as suggested by Jiang et al.[33]): average p_L until a certain epoch

An overview of all combinations of design choices that are part of the experiments can be found in Table 4.4. For detailed results of the experiments evaluated under both metrics *hard-intersect* and *spearmanr* we refer the reader to Table 4.5 for CIFAR-10, Table 4.6 for CIFAR-100 and Table 4.7 for TinyImagenet.

Starting with basic design choices, the *h-score* consistency is indifferent to the choice of using a pretrained network or not (see Figure 4.4; left). The results in Figure 4.4 (middle) suggest that under little regularization, an increase in depth of ResNets can increase the *h-score* consistency. The experiments in Figure 4.4 (right) and more detailed in Table 4.5 show that different architectures require different optimization strategies to achieve optimal consistency. While ResNets perform better under an AdamW optimizer, VGG19 and EfficientNets produce better results with SGD optimizer. While an EfficientNetB1 with SGD and a ResNet18 with AdamW

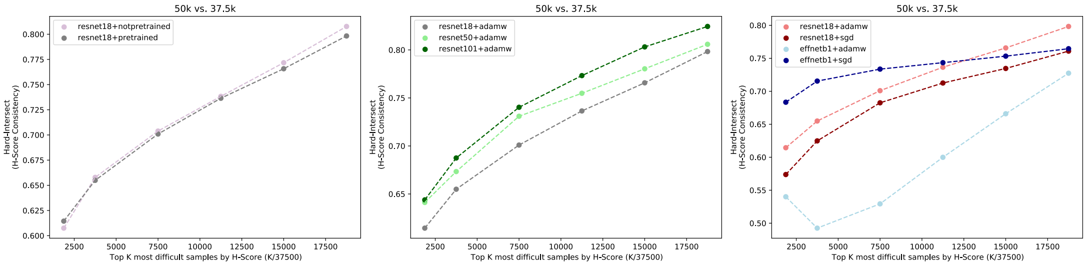


Figure 4.4: Class-wise *h-score* consistency between full dataset and a subset of 75% of CIFAR-10 measured by *hard-intersect* evaluated for different K most difficult samples. **Left:** Pretrained model versus random weight initialization. **Middle:** ResNet architectures of different depth. **Right:** Resnet18 and EfficientNetB1 with AdamW or SGD Optimizer.

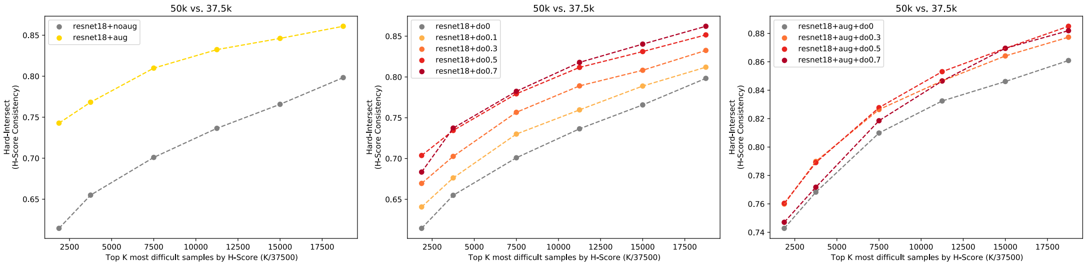


Figure 4.5: Class-wise *h-score* consistency between full dataset and a subset of 75% of CIFAR-10 measured by *hard-intersect* evaluated for different K most difficult samples. **Left:** Data augmentation versus no data augmentation. **Middle:** Using dropout with different rates (without data augmentation). **Right:** Data augmentation combined with dropout at different rates.

achieve similar consistency scores under little regularization, we tested the effect of regularization methods on ResNet18, before evaluating the best design choices for different architecture and optimization strategies. Figure 4.5 (left; middle) emphasizes that data augmentation as well as additional dropout layers have a positive impact on the *h-score* consistency. While applied without data augmentation, a dropout rate of 0.7 shows the strongest consistency in *h-scores*. Under data augmentation strategies the optimal dropout rate amounts to 0.3 (see Figure 4.5; right). The lower dropout rate for the combined experiment can be explained by the additional regularization effect due to data augmentation. Computing p_L with three (test-time) augmentations increases the subset consistency further (see Figure 4.6; left). However, the positive effect is independent of the choice of three or five augmentations. In contrast to test-time augmentation, test-time dropout did not reveal a positive impact on the *h-score* consistency. Further, the use of multiple training runs to produce an ensembled *h-score* is beneficial for reaching the highest consistencies. In particular, using five training runs achieves the best performance across all architectures (see Figure 4.6; middle, Table 4.5). Under optimal training design choices (regularization and optimization) all architectures attain an *h-score* consistency of above 0.9 for the 20% hardest samples ranked in the intersection of the full dataset and a 75% subset of CIFAR-10. ResNet18 and ResNet50 achieve the highest consistencies (Figure 4.6; right). It is an increase of over 50% in comparison to a baseline without any regularization. The same tendencies hold for an evaluation of self-consistency as well as adopting the *spearman rank correlation* metric (see Table 4.5). Averaging p_L until epoch 30 shows a slight advantage over epoch 20 when more regularization methods are used. The results for CIFAR-10 are consistent with the results for CIFAR-100 and TinyImagenet (see Table 4.6, see Table 4.7). In summary, the best training design choices for subset consistency of the *h-score* used in the following sections include data augmentation, dropout, test-time augmentation and an ensembled average of five training runs. Even though ResNet18 trained with AdamW produced the highest consistency (see ID 23 in Table 4.4), other architectures (sometimes trained with SGD) revealed comparable consistency when trained under the optimal training design choices.

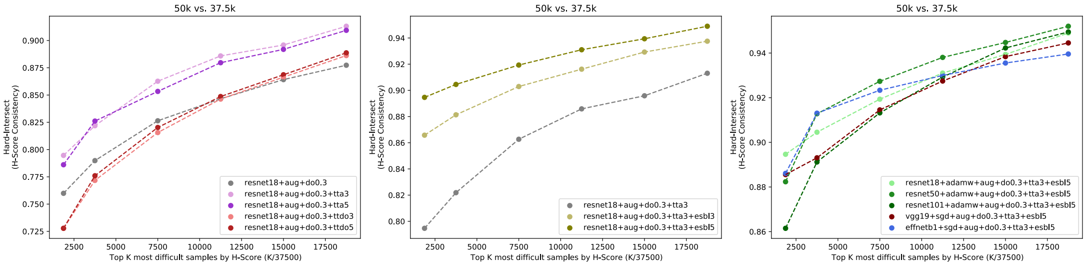
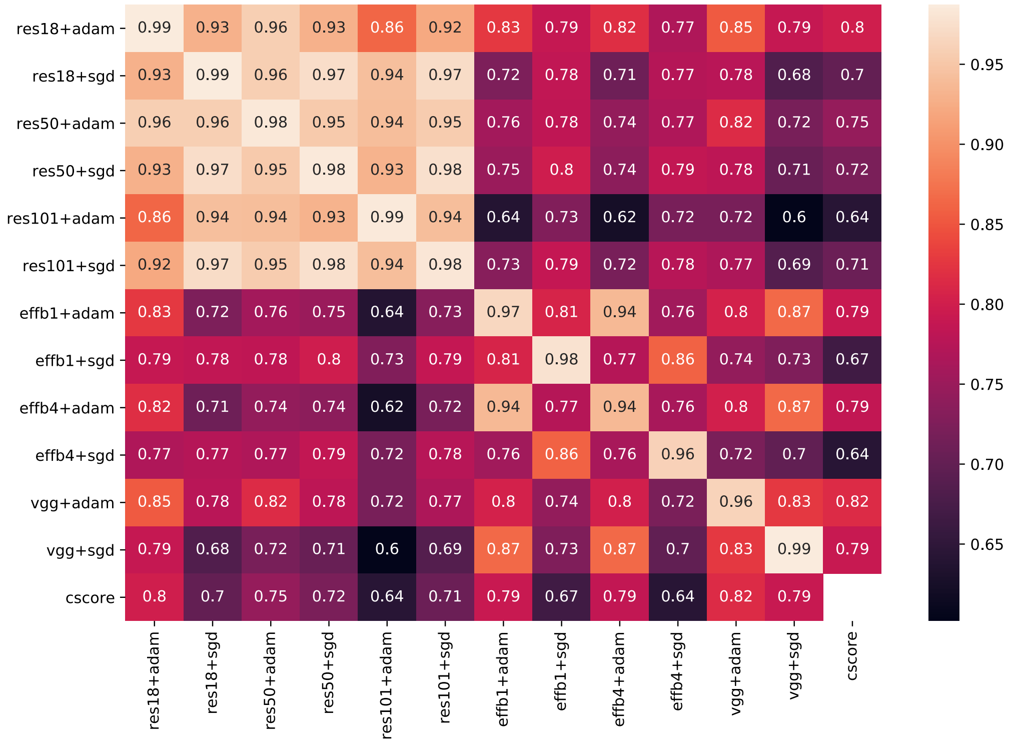


Figure 4.6: Class-wise *h-score* consistency between full dataset and a subset of 75% of CIFAR-10 measured by *hard-intersect* evaluated for different K most difficult samples. **Left:** Test-time augmentation and test-time dropout with different number of inference passes. **Middle:** Using ensembles of different sizes. **Right:** Different architectures with best optimizer with best *h-score* consistent training design choices.

4.3 Cross Model Consistency and Relationship with C-Score

Following Sections 4.1 and 4.2 that analyze the consistency of sample hardness from an intra-method perspective, this section investigates the model bias in *h-scores* rankings from an inter-method perspective. Figure 4.7 shows the *h-score* consistency measured by the *spearman rank correlation* between different model sizes of ResNets, EfficientNets and VGGs trained with either an SGD or an AdamW optimizer under the optimal training design choices for *h-score* consistency. In general, the different methods produce *h-score* rankings that follow similar characteristics. With few exceptions the *spearman rank correlation* is above 0.7 for all architectures and optimizer. It emphasizes that under an *h-score* consistent training setup, state-of-the-art DNN architectures follow similar paths when learning from a dataset. This is coherent with the empirical observations of Wu et al. [69]. Nonetheless, ResNet architectures tend to follow more consistent learning patterns than other architectures, regardless of size and optimization strategy. They produce the most similar *h-score* rankings with scores above 0.9. While network size does not have a big impact on the order of learning for EfficientNet and VGG, the choice of its optimizer influences the *h-score* ranking strongly. The variation can reach magnitudes of differences up to 0.15 (e.g., EfficientNetB1/B4 with AdamW versus EfficientNetB1 with AdamW/SGD). Compared to the *c-score* rankings [33] that are produced with an Inception network [62] as a backbone, the *spearman rank correlation* scores take on similar values to any other inter-method rank comparison of two different network families under the *h-score* between 0.7 and 0.8, including the *h-score* ranking produced with an Inception backbone.



	res18+adam	res18+sgd	res50+adam	res50+sgd	res101+adam	res101+sgd	effb1+adam	effb1+sgd	effb4+adam	effb4+sgd	vgg+adam	vgg+sgd	cscore
res18+adam	0.99	0.93	0.96	0.93	0.86	0.92	0.83	0.79	0.82	0.77	0.85	0.79	0.8
res18+sgd	0.93	0.99	0.96	0.97	0.94	0.97	0.72	0.78	0.71	0.77	0.78	0.68	0.7
res50+adam	0.96	0.96	0.98	0.95	0.94	0.95	0.76	0.78	0.74	0.77	0.82	0.72	0.75
res50+sgd	0.93	0.97	0.95	0.98	0.93	0.98	0.75	0.8	0.74	0.79	0.78	0.71	0.72
res101+adam	0.86	0.94	0.94	0.93	0.99	0.94	0.64	0.73	0.62	0.72	0.72	0.6	0.64
res101+sgd	0.92	0.97	0.95	0.98	0.94	0.98	0.73	0.79	0.72	0.78	0.77	0.69	0.71
effb1+adam	0.83	0.72	0.76	0.75	0.64	0.73	0.97	0.81	0.94	0.76	0.8	0.87	0.79
effb1+sgd	0.79	0.78	0.78	0.8	0.73	0.79	0.81	0.98	0.77	0.86	0.74	0.73	0.67
effb4+adam	0.82	0.71	0.74	0.74	0.62	0.72	0.94	0.77	0.94	0.76	0.8	0.87	0.79
effb4+sgd	0.77	0.77	0.77	0.79	0.72	0.78	0.76	0.86	0.76	0.96	0.72	0.7	0.64
vgg+adam	0.85	0.78	0.82	0.78	0.72	0.77	0.8	0.74	0.8	0.72	0.96	0.83	0.82
vgg+sgd	0.79	0.68	0.72	0.71	0.6	0.69	0.87	0.73	0.87	0.7	0.83	0.99	0.79
cscore	0.8	0.7	0.75	0.72	0.64	0.71	0.79	0.67	0.79	0.64	0.82	0.79	

Figure 4.7: The cross model consistency in *h-score* ranks on CIFAR-10 measured by *spearman rank correlation*. Last column/row represents the difficulty rank of the *c-score* proposed by Jiang et al. [33].

4.4 H-score Consistency for different Set Sizes

While Section 4.2 evaluated its experiments for a single subset size only, this section takes a closer look at the impact of varying dataset and subset sizes to *h-score* consistency. Experiments are carried out on CIFAR-10, CIFAR-100 and TinyImagenet with the training design choices that produces the highest *h-score* consistency. Details can be found in Table 4.4 for ID 23. The dataset sizes are set to $48,000$, $24,000$ and $12,000$ for CIFAR-10 and CIFAR-100 and to $96,000$, $48,000$ and $24,000$ for TinyImagenet. Subset sizes are 75%, 50% and 25% of the respective dataset size. When looking at the results of CIFAR-10 in Figure 4.8 (left), it is notable that there is only a marginal drop in consistency, when the dataset size is reduced. All dataset sizes follow similar dynamics when reducing the subset size. There is a decrease about 0.1 in *hard-intersect* consistency between the datasets intersected with subsets of 75% and subsets of 25%. The results of CIFAR-100 (see Figure 4.8; middle) and TinyImagenet (see Figure 4.8; right) suggest that in general, smaller class sizes (i.e., only a tenth of the class size in CIFAR-10) have a negative effect on the subset consistency when the dataset size is reduced. With decreasing subset size this effect is amplified. For example, the difference in *hard-intersect* of a dataset with 48000 samples from CIFAR-100 intersected with a subset of 75% and a smaller dataset with 12000 samples from CIFAR-100 intersected with a subset of 75% is about 0.05. This difference grows up to 0.09 when intersected with subsets of 25%. A similar trend is observable for TinyImagenet. Nonetheless, even a very small dataset of 12000 samples from CIFAR-100 that consists of only 120 samples per class has an intersection score of 0.7 for the 20% hardest samples that occur in both this dataset and a respective subset with 3000 samples.

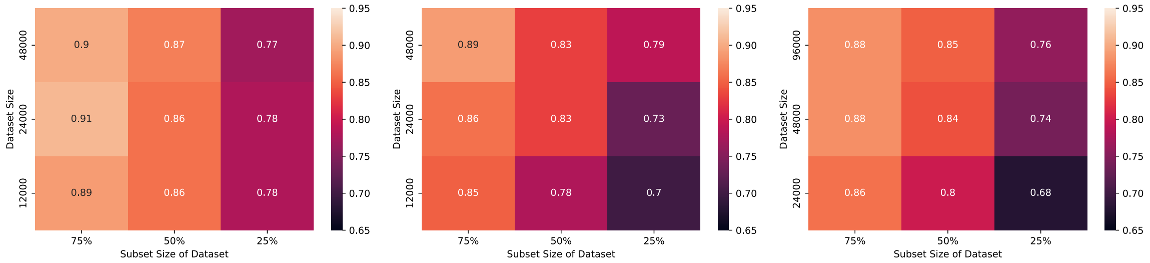


Figure 4.8: *Hard-intersect* scores for 20% hardest samples in intersection of a dataset and a subset with varying sizes. **Left:** CIFAR-100. **Middle:** CIFAR-100. **Right:** TinyImagenet.

4.5 H-score Consistency under Label Noise

In order to understand *h-score* consistency under label noise, we analyze the resulting *h-scores* for the best training design choices. The *h-score* consistency is computed between the intersecting clean samples of a noise-free and a noisy-labeled version of a dataset. Figure 4.9 (left) compares the *h-score* consistency between the full dataset with a 75% subset of CIFAR-10 without noise and the full dataset with a version consisting of 25% label noise (i.e., 75% clean subset) for a ResNet18 with AdamW optimizer. While conducted without label noise, the subset achieves a *hard-intersect* score of around 0.93 considering the 20% hardest samples, its counterpart under 25% label noise achieves a score of 0.83 only. It becomes apparent that label noise injection influences the internal understanding of hard and easy samples even for a method with high *h-score* consistency. Figure 4.9 (right) visualizes the impact of label noise under different rates. It can be observed that the *h-score* consistency decreases disproportionately strong for higher noise rates, resulting in less discriminative learning orders with respect to hard samples.

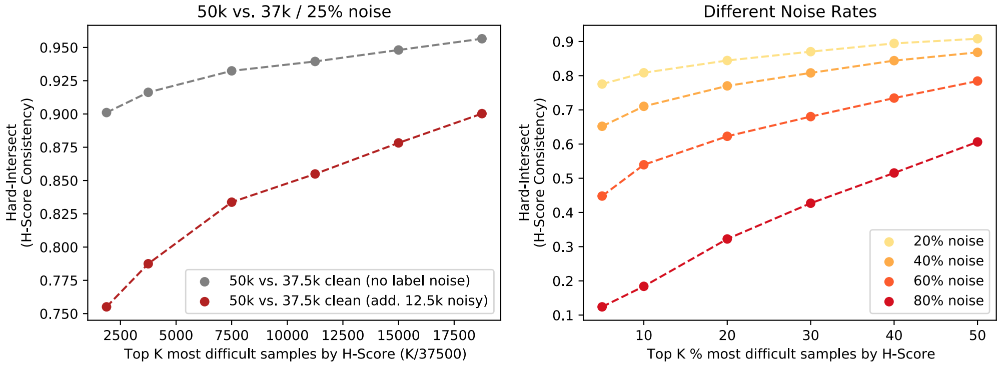


Figure 4.9: *H-score* consistency under label noise for ResNet18 and AdamW optimizer for CIFAR-10 dataset. **Left:** *hard-intersect* scores for full dataset and 75% clean subset with and without 25% additional noisy-labeled samples. **Right:** *hard-intersect* scores for different noise rates (different clean subsets).

When comparing different architectures and optimization strategies for the training design choices with the highest *h-score* consistency, the observations partly contradict the consistent results of Section 4.2. While methods based on ResNets trained with AdamW optimizer obtained the highest subset consistency scores for all dataset without label noise, the outcome for label noise scenarios is more heterogeneous. With the exception of a noise rate of 20%, Figure 4.10 emphasizes that SGD Optimizer can provide better learning consistency under label noise. For label noise scenarios of 40% and 60% ResNet architectures trained by SGD lead to highest consistency, while under stronger label noise an EfficientNetB4 backbone produces the most consistent *h-scores*. Further, it is the only method that has relative stable consistency values across all noise levels. Similar observations can be made for CIFAR-100 and TinyImagenet (see Table 4.2, Table 4.3).

When looking at the distribution of the *h-scores* under different noise rates, it becomes apparent that with increasing noise rates the *h-score* looses its discriminative power (see Figure 4.11, 4.12). While *h-scores* spread over a wide range from 0.0 to 0.8 for a noise rate of 20% with a clear peak of clean (and easy) samples at 0.0 and mostly noisy-labeled samples located at the higher *h-score* values, both observations diminish with increasing noise rates. The higher the noise rate the smaller the variance of the *h-scores* gets, eventually ranging from 0.35 to 0.55 only for 80% noise. Simultaneously, the clear duality of noisy-labeled samples located at the higher end (more difficult) of the *h-score* spectrum and the majority of clean samples located at the lower end (easier) disappears. Eventually, at 80% label noise, no clear decision boundary between the two types of samples is visible. In contrast to the findings on consistency under label noise, the EfficientNetB4 backbone produces more discriminative *h-scores* than a ResNet18 backbone in terms of correct and mislabeled samples even for smaller noise scenarios (see Figure 4.11, 4.12).

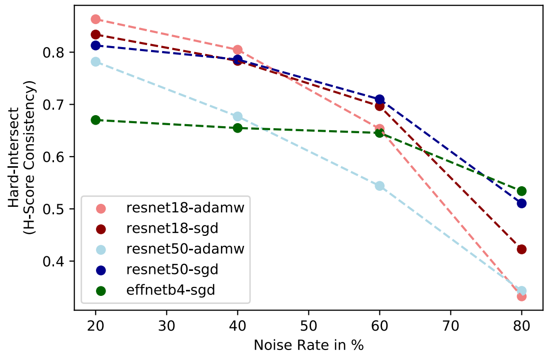


Figure 4.10: For the CIFAR-10 dataset, comparison of results of different backbones and architectures for best subset consistent training design choices under different noise rates. *H-intersect* scores are related to respective 20% hardest samples between intersecting clean samples of the noise-free and noisy-labeled dataset.

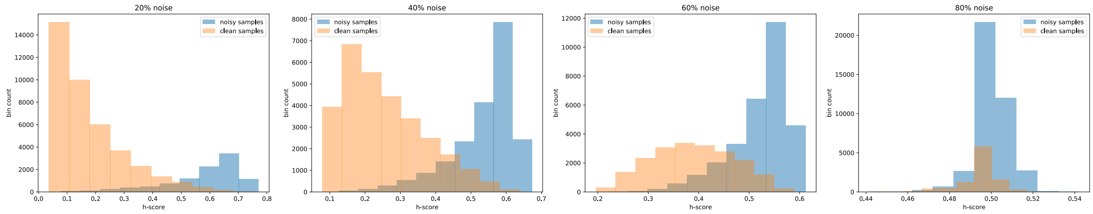


Figure 4.11: For the CIFAR-10 dataset, *h-score* distributions of clean and noisy-labeled samples under different noise rates for ResNet18 and AdamW optimizer.

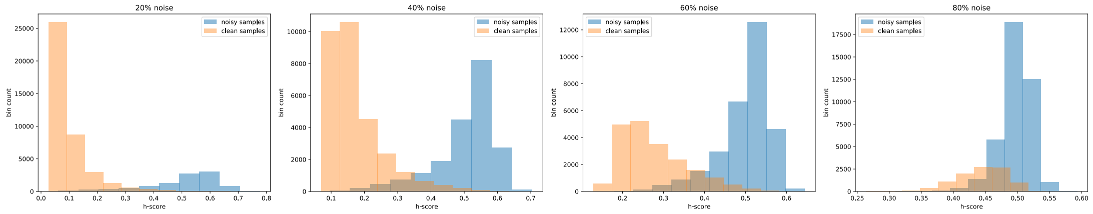


Figure 4.12: For the CIFAR-10 dataset, *h-score* distributions of clean and noisy-labeled samples under different noise rates for EfficientNetB4 and SGD optimizer.

Another perspective on the capacity of structural learning under label noise enables a closer look at the samples located at the lower end of the *h-score* rankings (i.e., easier samples). Figure 4.13 visualizes the relative noise rates among the easiest samples up to a certain threshold considering a balanced number of easy samples per class for CIFAR-10. The superiority of an SGD Optimizer in combination with an EfficientNetB4 backbone becomes particularly apparent under 80% label noise where only a third of the easiest 20% of the samples contain label noise. For

ResNet backbones optimized with AdamW the proportion of noisy-labeled samples among the same amount of easy samples is twice as high. The result suggest that even though ResNet architectures optimized with AdamW yield to the highest *h-score* subset consistency without label noise, these methods have little structural learning capacity under high noise rates. This observation is further emphasized by the results for CIFAR-100 and TinyImagenet (see Appendix, Figure A2 and Figure A3).

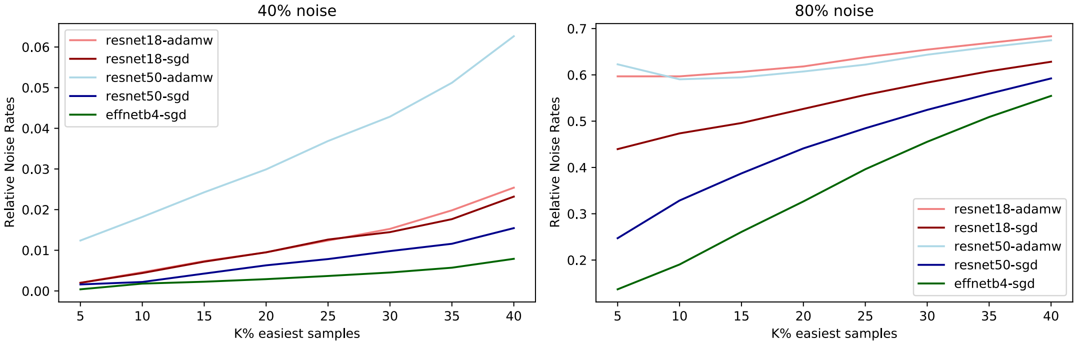


Figure 4.13: For the CIFAR-10 dataset, relative noise rates among top K% easiest samples according to *h-score*. **Left:** under 40% label noise. **Right:** under 80% label noise.

4.6 Conclusions

Experiments have shown that by adopting the softmax confidences of the annotated class as a difficulty measure during the course of learning subset consistent hardness scores can be produced. In particular, the consistency can be increased when using specific design choices during training. These design choices include data augmentation, dropout, test-time augmentation and an ensembled average of five training runs. While architectures of the ResNet family trained by an AdamW optimizer revealed the highest subset consistency without noise, the results for experiments with different rates of label noise are more ambiguous. During the experiments, it became apparent that SGD optimizer produce more consistent *h-scores* than AdamW optimizer when label noise is introduced. Further, when studying the decision boundary between clean and noisy-labeled samples in the respective *h-scores* rankings as well as examining the relative noise rates among the easiest samples, an EfficientNetB4 backbone produced the best results. In general, the truthfulness of Hypothesis 2 was proven. Nonetheless, the decision of the architectural backbone and the optimizer for producing the *h-score* ranking depends on the amount of artificial label noise induced to the dataset. A ResNet18 trained by an SGD optimizer is the best choice when considering all noise scenarios from 0% to 80% noise.

Table 4.1: *H-score* consistency under label noise for CIFAR-10 dataset.

Noise Rate Arch+Optim	20%	40%	60%	80%
resnet18-adamw	**0.86**	**0.80**	0.65	0.33
resnet18-sgd	0.83	0.78	**0.70**	0.42
resnet50-adamw	0.78	0.68	0.54	0.34
resnet50-sgd	0.81	0.79	**0.71**	0.51
effnetb4-sgd	0.67	0.65	0.65	**0.53**

Table 4.2: *H-score* consistency under label noise for CIFAR-100 dataset.

Noise Rate Arch+Optim	20%	40%	60%	80%
resnet18-adamw	0.83	0.71	0.59	0.41
resnet18-sgd	**0.84**	**0.79**	0.68	0.50
resnet50-adamw	0.81	0.74	0.62	0.43
resnet50-sgd	0.82	**0.79**	**0.72**	**0.58**
effnetb4-sgd	0.62	0.62	0.60	0.51

Table 4.3: *H-score* consistency under label noise for TinyImagenet dataset.

Noise Rate Arch+Optim	20%	40%	60%	80%
resnet18-adamw	**0.86**	0.76	0.62	0.35
resnet18-sgd	0.82	**0.80**	**0.76**	0.49
resnet50-adamw	0.80	0.73	0.61	0.28
resnet50-sgd	0.78	0.77	0.74	**0.64**
effnetb4-sgd	0.67	0.66	0.63	0.58

Table 4.4: Overview of Training Design Choices

ID	Arch	Optim	DA	DO	TTA	TTDO	Esbl.	Pretr.
1	ResNet18	AdamW	x	x	x	x	x	x
2	ResNet18	AdamW	x	x	x	x	x	✓
3	ResNet18	SGD	x	x	x	x	x	✓
4	ResNet50	AdamW	x	x	x	x	x	✓
5	ResNet101	AdamW	x	x	x	x	x	✓
6	VGG19	AdamW	x	x	x	x	x	✓
7	VGG19	SGD	x	x	x	x	x	✓
8	EfficientNetB1	AdamW	x	x	x	x	x	✓
9	EfficientNetB1	SGD	x	x	x	x	x	✓
10	ResNet18	AdamW	✓	x	x	x	x	✓
11	ResNet18	AdamW	x	0.1	x	x	x	✓
12	ResNet18	AdamW	x	0.3	x	x	x	✓
13	ResNet18	AdamW	x	0.5	x	x	x	✓
14	ResNet18	AdamW	x	0.7	x	x	x	✓
15	ResNet18	AdamW	✓	0.3	x	x	x	✓
16	ResNet18	AdamW	✓	0.5	x	x	x	✓
17	ResNet18	AdamW	✓	0.7	x	x	x	✓
18	ResNet18	AdamW	✓	0.5	3	x	x	✓
19	ResNet18	AdamW	✓	0.5	5	x	x	✓
20	ResNet18	AdamW	✓	0.5	x	3	x	✓
21	ResNet18	AdamW	✓	0.5	x	5	x	✓
22	ResNet18	AdamW	✓	0.3	3	x	3	✓
23	ResNet18	AdamW	✓	0.3	3	x	5	✓
24	ResNet50	AdamW	✓	0.3	3	x	3	✓
25	ResNet50	AdamW	✓	0.3	3	x	5	✓
26	ResNet101	AdamW	✓	0.3	3	x	3	✓
27	ResNet101	AdamW	✓	0.3	3	x	5	✓
28	VGG19	SGD	✓	0.3	3	x	3	✓
29	VGG19	SGD	✓	0.3	3	x	5	✓
30	EfficientNetB1	SGD	✓	0.3	3	x	3	✓
31	EfficientNetB1	SGD	✓	0.3	3	x	5	✓
32	EfficientNetB4	SGD	✓	0.3	3	x	3	✓
33	EfficientNetB4	SGD	✓	0.3	3	x	5	✓
34	InceptionV3	SGD	✓	0.3	3	x	3	✓
35	InceptionV3	SGD	✓	0.3	3	x	5	✓

Explanations. **Arch:** Architecture. **Optim** Optimizer. **DA:** Data Augmentation. **DO:** Dropout Layer with rates $0.1, 0.3, 0.5, 0.7$. **TTA:** Test-Time Augmentation. **TTDO:** Test-Time Dropout. **Esbl.:** Use of ensembles to produce *h-score*. **Pretr.:** Use a pretrained network.

Table 4.5: Results for design choice experiments for CIFAR-10 dataset.

ID	Setup Name	h-inters20	h-inters30	spearm20	spearm30
		self/sub75	self/sub75	self/sub75	self/sub75
1	r18-ad-xx-notpretrained	0.68/0.70	0.69/0.70	0.78/0.79	0.78/0.80
2	r18-ad-xx-do0-ta0-to0-e1	0.71/0.70	0.69/0.70	0.78/0.78	0.73/0.77
3	r18-sg-xx-do0-ta0-to0-e1	0.69/0.68	0.69/0.68	0.72/0.72	0.72/0.72
4	r50-ad-xx-do0-ta0-to0-e1	0.73/0.73	0.74/0.73	0.81/0.80	0.81/0.81
5	r101-ad-xx-do0-ta0-to0-e1	0.76/0.74	0.76/0.74	0.85/0.83	0.85/0.83
6	vgg-ad-xx-do0-ta0-to0-e1	0.20/0.72	0.20/0.73	0.01/0.78	0.01/0.80
7	vgg-sg-xx-do0-ta0-to0-e1	0.61/0.59	0.61/0.59	0.69/0.67	0.69/0.67
8	eff-ad-xx-do0-ta0-to0-e1	0.54/0.53	0.54/0.53	0.61/0.60	0.61/0.61
9	eff-sg-xx-do0-ta0-to0-e1	0.74/0.73	0.74/0.74	0.71/0.72	0.71/0.72
10	r18-ad-da-do0-ta0-to0-e1	0.82/0.81	0.83/0.82	0.89/0.89	0.90/0.90
11	r18-ad-xx-do1-ta0-to0-e1	0.73/0.73	0.72/0.72	0.82/0.81	0.78/0.78
12	r18-ad-xx-do3-ta0-to0-e1	0.76/0.76	0.76/0.76	0.85/0.84	0.85/0.84
13	r18-ad-xx-do5-ta0-to0-e1	0.79/0.78	0.79/0.78	0.87/0.86	0.88/0.87
14	r18-ad-xx-do7-ta0-to0-e1	0.80/0.78	0.80/0.79	0.89/0.88	0.90/0.90
15	r18-ad-da-do3-ta0-to0-e1	0.83/0.83	0.85/0.84	0.91/0.90	0.92/0.92
16	r18-ad-da-do5-ta0-to0-e1	0.83/0.83	0.85/0.84	0.91/0.91	0.93/0.93
17	r18-ad-da-do7-ta0-to0-e1	0.83/0.82	0.84/0.84	0.92/0.91	0.94/0.93
18	r18-ad-da-do5-ta3-to0-e1	0.86/0.86	0.87/0.87	0.95/0.95	0.96/0.96
19	r18-ad-da-do5-ta5-to0-e1	0.86/0.85	0.87/0.86	0.95/0.95	0.96/0.95
20	r18-ad-da-do5-ta0-to3-e1	0.82/0.82	0.83/0.82	0.93/0.93	0.94/0.93
21	r18-ad-da-do5-ta0-to5-e1	0.83/0.82	0.83/0.83	0.94/0.93	0.94/0.93
22	r18-ad-da-do3-ta3-to0-e3	0.91/0.90	0.92/0.91	0.98/0.97	0.97/0.97
23	r18-ad-da-do3-ta3-to0-e5	0.93/0.92	0.94/0.92	**0.99/0.98**	0.98/0.97
24	r50-ad-da-do3-ta3-to0-e3	0.92/0.90	0.93/0.91	0.98/0.97	0.98/0.97
25	**r50-ad-da-do3-ta3-to0-e5**	**0.94/0.93**	**0.95/0.93**	**0.99/0.98**	**0.99/0.99**
26	r101-ad-da-do3-ta3-to0-e3	0.91/0.91	0.92/0.91	0.98/0.98	0.98/0.98
27	r101-ad-da-do3-ta3-to0-e5	0.93/0.91	0.95/0.91	**0.99/0.98**	**0.99**/0.98
28	vgg-sg-da-do3-ta3-to0-e3	0.91/0.90	0.91/0.90	0.98/0.97	0.98/0.97
29	vgg-sg-da-do3-ta3-to0-e5	0.93/0.91	0.93/0.91	**0.99/0.98**	**0.99**/0.98
30	eff1-sg-da-do3-ta3-to0-e3	0.92/0.91	0.92/0.91	0.96/0.97	0.96/0.97
31	eff1-sg-da-do3-ta3-to0-e5	0.93/0.92	0.93/0.92	0.98/0.97	0.98/0.97
32	eff4-sg-da-do3-ta3-to0-e3	0.92/0.91	0.92/0.91	0.96/0.96	0.96/0.96
33	eff4-sg-da-do3-ta3-to0-e5	0.93/0.92	0.93/0.92	0.96/0.96	0.96/0.95
34	incv3-sg-da-do3-ta3-to0-e3	0.89/0.88	0.89/0.88	0.96/0.96	0.96/0.96
35	incv3-sg-da-do3-ta3-to0-e5	0.91/0.91	0.91/0.91	0.97/0.97	0.97/0.97

Explanations. **h-inters20/30:** hardness intersection score of p_L averaged until epoch 20/30, values computed for 20% hardest samples. **spearm20/30:** class-wise spearman rank correlation for p_L averaged until epoch 20/30. First value is related to self-consistency, while second value is associated to consistency between original dataset and a subset of 75% its size.

Table 4.6: Results for design choice experiments for CIFAR-100 dataset.

ID	Setup Name	h-inters20	h-inters30	spearm20	spearm30
		self/sub75	self/sub75	self/sub75	self/sub75
1	r18-ad-xx-notpretrained	0.68/0.67	0.68/0.66	0.73/0.72	0.72/0.71
2	r18-ad-xx-do0-ta0-to0-e1	0.70/0.70	0.70/0.70	0.75/0.74	0.75/0.74
3	r18-sg-xx-do0-ta0-to0-e1	0.62/0.61	0.62/0.61	0.62/0.62	0.62/0.62
4	r50-ad-xx-do0-ta0-to0-e1	0.71/0.69	0.71/0.68	0.75/0.73	0.75/0.72
5	r101-ad-xx-do0-ta0-to0-e1	0.70/0.71	0.69/0.70	0.74/0.76	0.73/0.75
6	vgg-ad-xx-do0-ta0-to0-e1	0.70/0.68	0.70/0.67	0.78/0.74	0.79/0.75
7	vgg-sg-xx-do0-ta0-to0-e1	0.46/0.47	0.46/0.47	0.46/0.43	0.46/0.43
8	eff-ad-xx-do0-ta0-to0-e1	0.50/0.50	0.50/0.50	0.49/0.48	0.49/0.48
9	eff-sg-xx-do0-ta0-to0-e1	0.66/0.65	0.66/0.65	0.61/0.61	0.61/0.61
10	r18-ad-da-do0-ta0-to0-e1	0.81/0.80	0.81/0.80	0.87/0.86	0.89/0.88
11	r18-ad-xx-do1-ta0-to0-e1	0.74/0.73	0.74/0.73	0.82/0.81	0.82/0.81
12	r18-ad-xx-do3-ta0-to0-e1	0.79/0.77	0.79/0.78	0.87/0.86	0.88/0.87
13	r18-ad-xx-do5-ta0-to0-e1	0.79/0.79	0.81/0.80	0.88/0.88	0.90/0.90
14	r18-ad-xx-do7-ta0-to0-e1	0.75/0.79	0.74/0.80	0.83/0.87	0.84/0.89
15	r18-ad-da-do3-ta0-to0-e1	0.82/0.80	0.84/0.83	0.91/0.90	0.92/0.92
16	r18-ad-da-do5-ta0-to0-e1	0.80/0.80	0.82/0.82	0.89/0.89	0.91/0.92
17	r18-ad-da-do7-ta0-to0-e1	0.78/0.79	0.80/0.81	0.85/0.86	0.88/0.89
18	r18-ad-da-do3-ta3-to0-e1	0.84/0.84	0.86/0.85	0.94/0.93	0.95/0.94
19	r18-ad-da-do3-ta5-to0-e1	0.85/0.84	0.86/0.85	0.93/0.93	0.94/0.94
20	r18-ad-da-do3-ta0-to3-e1	0.82/0.82	0.83/0.83	0.92/0.91	0.93/0.92
21	r18-ad-da-do3-ta0-to5-e1	0.82/0.82	0.83/0.83	0.92/0.92	0.93/0.93
22	r18-ad-da-do3-ta3-to0-e3	0.90/0.90	0.91/0.90	0.97/0.97	0.98/0.97
23	**r18-ad-da-do3-ta3-to0-e5**	**0.93/0.91**	**0.93/0.91**	**0.98/0.98**	**0.98/0.98**
24	r50-ad-da-do3-ta3-to0-e3	0.89/0.86	0.88/0.84	0.91/0.92	0.90/0.90
25	r50-ad-da-do3-ta3-to0-e5	0.92/0.88	0.91/0.86	0.95/0.94	0.94/0.92
26	r101-ad-da-do3-ta3-to0-e3	0.91/0.88	0.91/0.88	0.96/0.94	0.97/0.94
27	r101-ad-da-do3-ta3-to0-e5	0.91/0.90	0.91/0.90	0.97/0.96	0.96/0.96
28	vgg-sg-da-do3-ta3-to0-e3	0.89/0.88	0.89/0.88	0.97/0.96	0.97/0.96
29	vgg-sg-da-do3-ta3-to0-e5	0.92/0.90	0.92/0.90	**0.98**/0.97	**0.98**/0.97
30	eff1-sg-da-do3-ta3-to0-e3	0.88/0.87	0.88/0.87	0.94/0.93	0.94/0.93
31	eff1-sg-da-do3-ta3-to0-e5	0.90/0.89	0.91/0.89	0.96/0.95	0.96/0.95
32	eff4-sg-da-do3-ta3-to0-e3	0.82/0.84	0.81/0.83	0.84/0.89	0.81/0.87
33	eff4-sg-da-do3-ta3-to0-e5	0.87/0.86	0.86/0.84	0.90/0.89	0.89/0.87
34	incv3-sg-da-do3-ta3-to0-e3	0.88/0.87	0.88/0.87	0.96/0.95	0.96/0.95
35	incv3-sg-da-do3-ta3-to0-e5	0.90/0.90	0.90/0.90	0.97/0.97	0.97/0.97

Explanations. **h-inters20/30:** hardness intersection score of p_L averaged until epoch 20/30, values computed for 20% hardest samples. **spearm20/30:** class-wise spearman rank correlation for p_L averaged until epoch 20/30. First value is related to self-consistency, while second value is associated to consistency between original dataset and a subset of 75% its size.

Table 4.7: Results for design choice experiments for TinyImagenet dataset.

ID	Setup Name	h-inters20	h-inters30	spearm20	spearm30
		self/sub75	self/sub75	self/sub75	self/sub75
1	r18-ad-xx-notpretrained	0.73/0.73	0.73/0.73	0.74/0.74	0.74/0.74
2	r18-ad-xx-do0-ta0-to0-e1	0.72/0.71	0.72/0.71	0.75/0.74	0.76/0.74
3	r18-sg-xx-do0-ta0-to0-e1	0.57/0.59	0.57/0.59	0.50/0.53	0.50/0.53
4	r50-ad-xx-do0-ta0-to0-e1	0.73/0.73	0.71/0.71	0.80/0.79	0.80/0.79
5	r101-ad-xx-do0-ta0-to0-e1	0.73/0.73	0.72/0.72	0.80/0.80	0.80/0.80
6	vgg-ad-xx-do0-ta0-to0-e1	0.20/0.20	0.20/0.20	0.00/0.00	0.00/0.00
7	vgg-sg-xx-do0-ta0-to0-e1	0.42/0.43	0.42/0.43	0.45/0.42	0.45/0.42
8	eff-ad-xx-do0-ta0-to0-e1	0.67/0.65	0.67/0.65	0.69/0.67	0.69/0.67
9	eff-sg-xx-do0-ta0-to0-e1	0.63/0.63	0.63/0.63	0.53/0.55	0.53/0.55
10	r18-ad-da-do0-ta0-to0-e1	0.75/0.74	0.76/0.74	0.83/0.82	0.85/0.83
11	r18-ad-xx-do1-ta0-to0-e1	0.74/0.73	0.74/0.73	0.79/0.78	0.80/0.79
12	r18-ad-xx-do3-ta0-to0-e1	0.77/0.77	0.77/0.76	0.85/0.84	0.85/0.84
13	r18-ad-xx-do5-ta0-to0-e1	0.81/0.79	0.80/0.78	0.90/0.89	0.90/0.89
14	r18-ad-xx-do7-ta0-to0-e1	0.83/0.82	0.84/0.81	0.93/0.91	0.93/0.92
15	r18-ad-da-do3-ta0-to0-e1	0.81/0.78	0.82/0.79	0.89/0.88	0.90/0.89
16	r18-ad-da-do5-ta0-to0-e1	0.81/0.81	0.83/0.82	0.90/0.90	0.92/0.91
17	r18-ad-da-do7-ta0-to0-e1	0.82/0.81	0.84/0.83	0.91/0.90	0.93/0.93
18	r18-ad-da-do3-ta3-to0-e1	0.83/0.83	0.82/0.83	0.93/0.92	0.93/0.93
19	r18-ad-da-do3-ta5-to0-e1	0.85/0.83	0.84/0.82	0.94/0.93	0.94/0.93
20	r18-ad-da-do3-ta0-to3-e1	0.83/0.81	0.82/0.80	0.93/0.92	0.93/0.92
21	r18-ad-da-do3-ta0-to5-e1	0.83/0.81	0.82/0.80	0.93/0.92	0.93/0.92
22	r18-ad-da-do3-ta3-to0-e3	0.90/0.88	0.90/0.88	0.97/0.96	0.97/0.96
23	**r18-ad-da-do3-ta3-to0-e5**	**0.93**/0.90	**0.92/0.90**	**0.98**/0.97	**0.98**/0.97
24	r50-ad-da-do3-ta3-to0-e3	0.90/0.86	0.89/0.86	0.97/0.95	0.97/0.95
25	r50-ad-da-do3-ta3-to0-e5	0.92/0.89	0.92/0.88	**0.98**/0.97	**0.98**/0.97
26	r101-ad-da-do3-ta3-to0-e3	0.90/0.89	0.90/0.88	0.97/0.96	0.97/0.96
27	**r101-ad-da-do3-ta3-to0-e5**	0.92/**0.91**	**0.92/0.90**	**0.98/0.98**	**0.98/0.98**
28	vgg-sg-da-do3-ta3-to0-e3	0.89/0.88	0.90/0.88	0.95/0.94	0.95/0.94
29	vgg-sg-da-do3-ta3-to0-e5	0.91/0.90	**0.92/0.90**	0.97/0.96	0.97/0.96
30	eff-sg-da-do3-ta3-to0-e3	0.88/0.88	0.88/0.88	0.92/0.91	0.92/0.91
31	eff-sg-da-do3-ta3-to0-e5	0.90/0.90	0.90/0.90	0.93/0.93	0.93/0.94
32	eff4-sg-da-do3-ta3-to0-e3	0.89/0.87	0.89/0.88	0.93/0.93	0.93/0.93
33	eff4-sg-da-do3-ta3-to0-e5	0.91/0.90	0.91/0.90	0.96/0.95	0.96/0.95
34	incv3-sg-da-do3-ta3-to0-e3	0.90/0.89	0.90/0.89	0.96/0.95	0.96/0.95
35	incv3-sg-da-do3-ta3-to0-e5	0.92/0.91	0.92/0.91	0.97/0.97	0.97/0.97

Explanations. **h-inters20/30:** hardness intersection score of p_L averaged until epoch 20/30, values computed for 20% hardest samples. **spearm20/30:** class-wise spearman rank correlation for p_L averaged until epoch 20/30. First value is related to self-consistency, while second value is associated to consistency between original dataset and a subset of 75% its size.

5 Informativeness of Samples

A common assumption in deep learning is the existence of a hierarchy of informativeness of samples [46]. According to this understanding, easy samples are more similar to each other and therefore more redundant than hard samples when learning from a dataset. In contrast, hard samples are attributed with more diverse features and a higher informative value for generalization. Good training results cannot be achieved without the contribution of these hard samples. Hypothesis 3 formalizes this assumption. It is a necessary prerequisite for justifying the focus on distinguishing hard from mislabeled samples in label noise research (see Hypothesis 1). The following chapter investigates the truthfulness of Hypothesis 3 by analyzing the relationship between sample difficulty and informative value.

5.1 Difficulty of Test Set

Before meaningful statements on the informative value of a sample can be made, it is essential to better understand the test set in terms of difficulty, since the capacity of a method to generalize well is determined by it. Significant differences in the notion of difficulty between the train and the test set may distort deductions from generalization performance on the test set. For example, an easier test set compared to the train set could make hard samples disproportionately more redundant, while a harder test set could make easier samples disproportionately more redundant. In order to analyze the relationship in terms of difficulty between the train and the test set, a joined hardness score is build. Specifically, the train and test samples are combined and trained jointly following the procedure of the best subset consistent training desgin choices for producing the hardness scores. Then, each class is divided into ten difficulty bins, according to the respective class specific *h-score* ranking. For each difficulty bin, the fraction of samples originally belonging to the test set is computed. Figure 5.1 shows the results for CIFAR-10 (left) , CIFAR-100 (middle) and TinyImageNet (right). It emphasizes that for the three datasets,

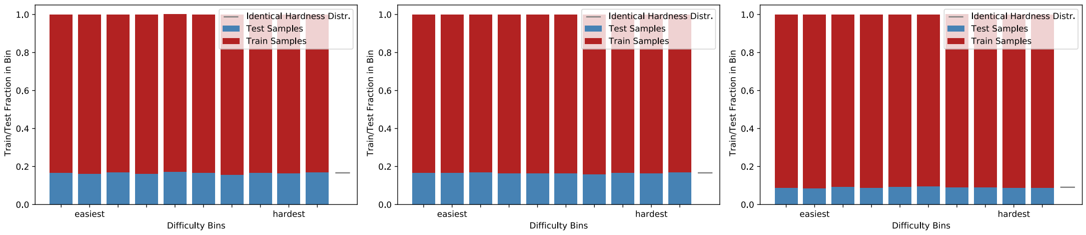


Figure 5.1: Joined *h-scores* of train and test set binned per class into ten difficulty categories for CIFAR-10 (**left**), CIFAR-100 (**middle**) and TinyImagenet (**right**).

the train-test split is performed in a difficulty preserving manner. The deviation to the expected number of test samples per bin is less than 30 per bin. Samples of each degree of difficulty are proportionally distributed to the train and test set.

5.2 Informative Value w.r.t. Test Accuracy

In order to understand the contribution of samples of different degree of difficulty to the overall performance, we evaluate the test set accuracy on different subsets of the train set that are product of holding out samples belonging to a certain interval of difficulty. In the following, e.g., the set $D_{HO60\%-70\%}$represents the subset of the train set that excludes all samples that are harder than 60% of the samples of its respective class, but are easier than 30% of the samples of its respective class. In other words, belong to the fourth easiest difficulty bin under a split of 10 bins. As a measure of difficulty we adopt the *h-score* under the best subset consistent training design choice and the *c-score* proposed by Jiang et. al [33]. For each dataset, we conduct experiments for three different sizes of hold-out sets, namely intervals of 10%, 20% and 30% of the data. To better comprehend the influence of samples of different degrees of difficulty, the

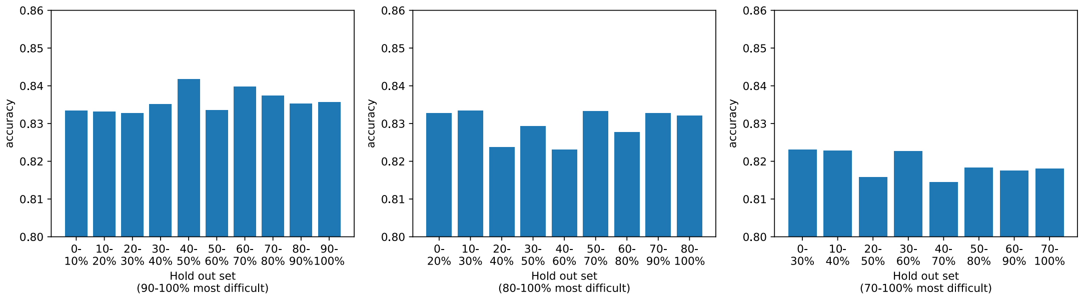


Figure 5.2: For the CIFAR-10 dataset, *h-score* for best setup with ResNet18 and SGD, set accuracy for train sets with hold-out intervals of 10% of the train set (**left**), 20% of the train set (**middle**) and 30% of the train set (**right**).

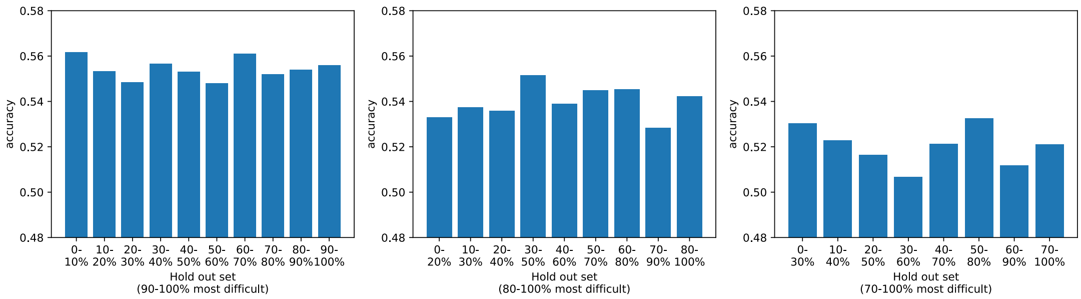


Figure 5.3: For the CIFAR-100 dataset, *h-score* for best setup with ResNet18 and SGD, test set accuracy for train sets with hold-out intervals of 10% of the train set (**left**), 20% of the train set (**middle**) and 30% of the train set (**right**).

subsets are produced by a sliding window of hold-out sets. Each score represents the average of five independent training runs of a ResNet18 on the same subset.

The *h-score* based results of CIFAR-10 and CIFAR-100 (see Figure 5.2 and Figure 5.3) do not reveal any significant differences in test set accuracy between holding-out harder or easier samples. While the min-max differences of scores does not vary by more than 0.02, no consistent pattern can be observed between holding out the same portion of samples inside a 10%, 20% or 30% hold-out interval. With respect to the test set accuracy, no hierarchy in informative value can be observed between easier and harder samples for the datasets CIFAR-10 and CIFAR-100. Further, the *c-score* based results for CIFAR-10 and CIFAR-100 coincide with these observations (see Appendix, Figure A4 and Figure A5). However, the results from TinyImagenet (see Figure 5.4) show minor differences in informative value that can be consistently observed over all three hold-out set sizes. The hardest samples have a slightly higher informative value than the other samples. The bigger the hold-out set at the top end of the hardest samples, the lower is the obtained accuracy on the test set in comparison to holding out other intervals of difficulty. While holding out 10% of the hardest samples leads to a difference of about 0.02 in accuracy, this difference increases up to 0.07 when holding out the hardest 30% of the data. Furthermore, a similar trend of smaller magnitude can be observed for the easiest samples. However, any interval of samples in the middle range of difficulty (20%-80%) did not reveal such effect. Instead of a linear correlation of difficulty and informative value, the hardest and the easiest samples have a slightly higher informative value than the rest of the samples for the TinyImagenet dataset. The informative value of medium hard samples is lower and constant.

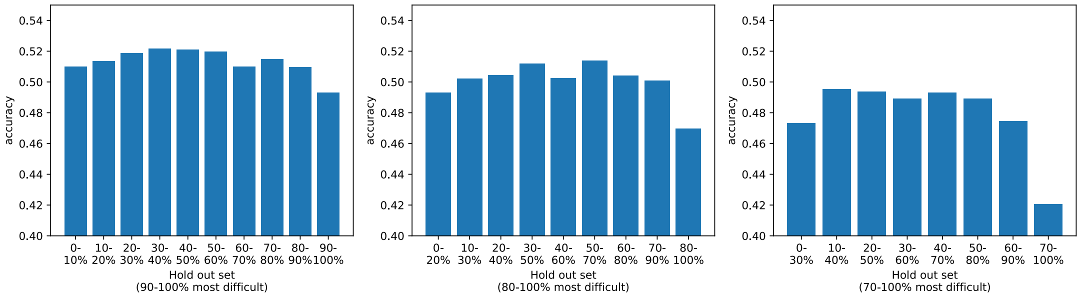


Figure 5.4: For the Tiny Imagenet dataset, *h-score* for best setup with ResNet18 and SGD, test set accuracy for train sets with hold-out intervals of 10% of the train set (**left**), 20% of the train set (**middle**) and 30% of the train set (**right**).

5.3 Informative Value w.r.t. Generalization

Studying the informative value of a group of samples by only observing the overall accuracy score on the test set bears some limitations for conclusions about its impact to generalization. Holding out an interval of samples belonging to a certain degree of difficulty in the train set could result in a decrease of predictive power for test samples with similar hardness. If only a minority of samples in the test set can be considered to be difficult (i.e., more diverse by its representation), worse predictive power on these rare hard samples would not be visible in the overall accuracy score. In contrast, a small decrease in predictive power for easier samples could be easily observed due to an over representation of these samples in the test set. Since the *h-score* is a relative metric of difficulty (e.g., different learning rates, dataset sizes, architectures or epochs until averaged can lead to different magnitudes of p_L), it is impossible to estimate the exact amount of hard samples in relation to easy samples. In order to investigate whether the above described situation occurs when evaluating the overall accuracy score on the test set, we further differentiate the evaluation of the test set by evaluating the performance of the hold-out models under different test set difficulty bins. Therefore, we make use of the *h-scores* of the test set produced in Section 5.1. We conduct experiments on subsets with hold-out sets according to a difficulty interval as described in Section 5.2, with the addition, that we evaluate each model by different bins of difficulty of the test set. The results in Figure 5.5, Figure 5.6 and Figure 5.7 evaluate the hold-out models for five test set bins with increasing hardness. The train sets used for this experiment comprise subsets with a hold-out set of 20% of the data. The lightest bars correspond to holding out the easiest samples, while the darkest bars correspond to holding out to the hardest samples. For CIFAR-10 and CIFAR-100 the hold-out models do not show different predictive capacity between each other for different degrees of difficulty of the test samples. The observation neglects the assumption that hard samples in the training curriculum are necessary to make more accurate predictions for hard (rare) test samples, and hence, are more important for generalization. In fact, the hardest samples from the train set of TinyImagenet do not improve predictive power on the hardest samples from the test set (see Figure 5.7). Instead, the results emphasize that the loss of generalization caused by holding-out the hardest samples of the train set tends to happen for easier samples of the test set. For example, the model trained without

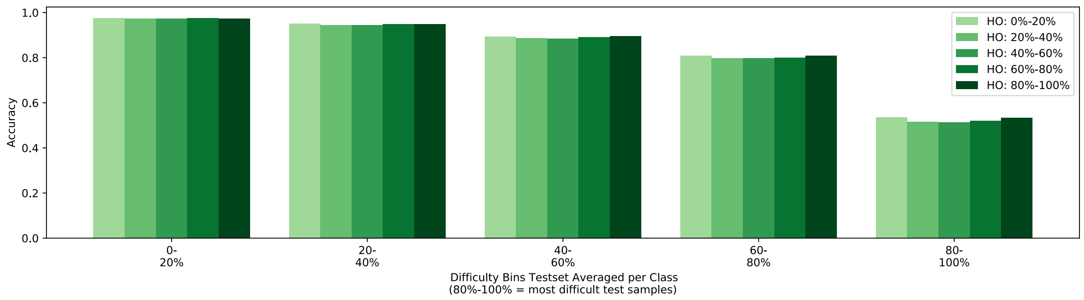


Figure 5.5: For the CIFAR-10 dataset, Hold-out training sets evaluated on five different difficulty bins of test set according to *h-score* for ResNet18 and SGD.

the hardest samples achieves an accuracy of less than 0.07 for the 20% easiest samples of the test set compared to other hold-out models, with increasing difficulty of the test sample bins the difference in accuracy diminishes. In general, it becomes apparent that in none of the datasets hard training samples provide additional informative value for classifying hard test samples. For CIFAR-10 and CIFAR-100 holding out the hardest samples during training does not damage generalization performance at all, i.e., the accuracy on different difficulty bins of the test set is affected in the same manner as for other hold-out training sets. In contrast, for TinyImagenet the decrease in performance is the most significant for easy test samples when holding out the hardest train samples.

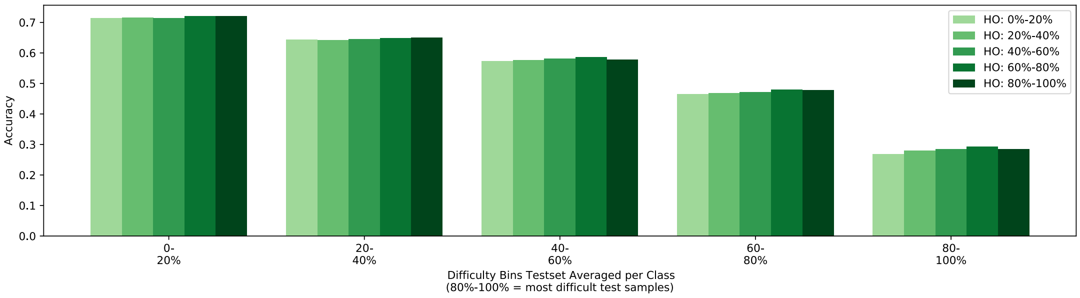


Figure 5.6: For the CIFAR-100 dataset, Hold-out training sets evaluated on five different difficulty bins of test set according to *h-score* for ResNet18 and SGD.

For the

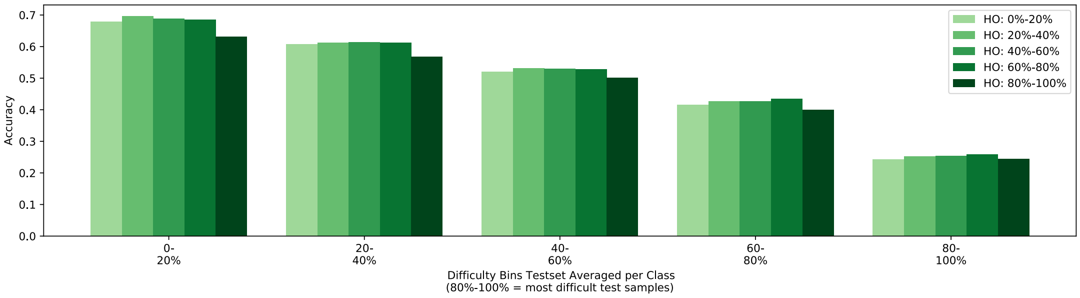


Figure 5.7: For the TinyImagenet dataset, Hold-out training sets evaluated on five different difficulty bins of test set according to *h-score* for ResNet18 and SGD.

5.4 Influence of H-Score Ensembling on Correlation with Informative Value

A possible explanation for a nonlinear (TinyImagenet) and an uncorrelated (CIFAR-10, CIFAR-100) relationship between the informative value and the *h-score* could lay in the fact that neural networks can reach multiple local optima via different learning paths. Even though the scores of the *spearman rank correlation* indicate a highly consistent learning order (see row 23 in Tables 4.5-4.7), the variance of the iteration in which individual samples are learned in different training runs might still spread a cluster of similar samples over multiple difficulty bins, such that excluding individual difficulty bins does not exclude all information from one cluster. The remaining samples still have enough capacity to represent the relevant features of the clusters samples for generalization. The training design choices for the highest subset consistent *h-score* include the ensembling of the *h-score* ranks of multiple training runs under the same setup. In order to examine the variance of individual samples in different learning paths, we calculate the difference of the min-max rank of samples per class for different training runs under the same setup. The results for CIFAR-10, CIFAR-100 and TinyImagenet are shown in Figure 5.8, Figure 5.9 and Figure 5.10. Each bar of the chart represents the samples of the respective difficulty bin according to the ensembled *h-score* and indicates the number of samples that have a min-max difference in the ranks of five training runs that is smaller than 10% of the size of the class. For CIFAR-10 it corresponds to being smaller than 500 (classes are of size 5000), for CIFAR-100 and TinyImagenet it corresponds to being smaller than 50 (classes are of size 500). The variance in the learning paths of these samples is small enough that they appear in the same difficulty bin consistently. The results for all three datasets show the same same tendency for both optimizer AdamW (left) and SGD (right). While the learning paths tend to start with the same easy samples and end with the same hard samples (most of them are learned inside the interval of one difficulty bin), the differences in the ranks for medium hard samples is big enough to spread over multiple difficulty bins. It can be potentially influence the indifference in informative value for medium difficult samples. Another observation reveals that the difficulty of the dataset (w.r.t accuracy and number of classes) lets the SGD optimizer produce less consistent results. The SGD optimizer produces the highest consistency for CIFAR-10 and the lowest for TinyImagenet.

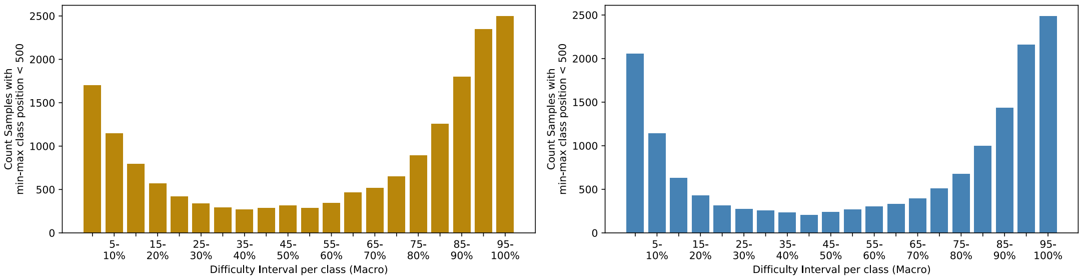


Figure 5.8: For the CIFAR-10 dataset, Min-max difference per sample class-wise in five h-score ranks of the same training setup that is smaller than 10% of the class size, binned by ensembled *h-score*. **Left:** ResNet18 and AdamW. **Right:** ResNet18 and SGD.

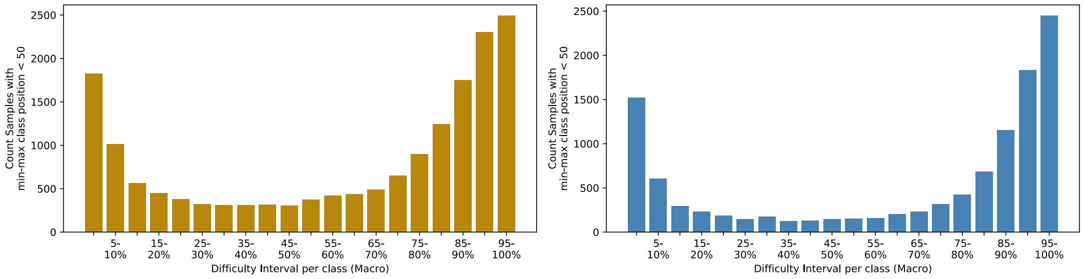


Figure 5.9: For the CIFAR-100 dataset, Min-max difference per sample class-wise in five h-score ranks of the same training setup that is smaller than 10% of the class size, binned by ensembled *h-score*. **Left:** ResNet18 and AdamW. **Right:** ResNet18 and SGD.

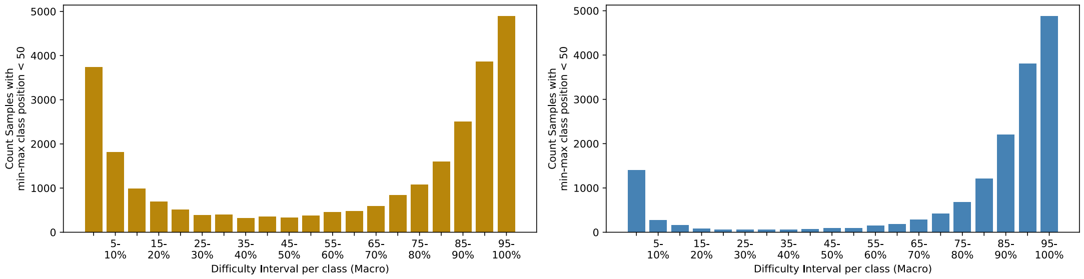


Figure 5.10: For the TinyImagenet dataset, Min-max difference per sample class-wise in five h-score ranks of the same training setup that is smaller than 10% of the class size, binned by ensembled *h-score*. **Left:** ResNet18 and AdamW. **Right:** ResNet18 and SGD.

5.5 Conclusions

This chapter partially rejected Hypothesis 3 that assumed a correlation between the *h-score* of a sample and its informative value. None of the three datasets showed a linear correlation. While CIFAR-10 and CIFAR-100 do not show any correlation, the easiest and hardest samples of TinyImagenet have marginally higher informative values than medium hard samples. The conclusions drawn from the experiments on the informative value are dataset depended and need to be further distinguished.

CIFAR-10 and CIFAR-100, the two most common datasets for ablation studies on artificial label noise, do not rely on hard samples more than on easy samples for good generalization performance. Considering the fact that hard and noisy-labeled samples are learned simultaneously at later stages of the training, a particular focus on learning from these hard samples is not justifiable given the equal importance of easy and hard samples for generalization. In fact, in opposition to Hypothesis 1, the results suggest that a particular focus on easy samples might be more beneficial for good generalization performance, since samples learned at earlier stages tend to be free of label noise while providing the same informativeness as any other sample. The ability of semi-supervised learning methods that achieve compatible performance to fully supervised method by only using a small subset of the labels (see Table 6.1 in Chapter 6) suggests retaining a mostly clean, easy subset could be more beneficial than being good in distinguishing between hard and noisy-labeled samples for CIFAR-10 and CIFAR-100.

In contrast, the results on the informativeness of hard samples for TinyImagenet point towards a different direction. Although the *h-score* does not provide a hierarchical ranking of importance for generalization, the 20% hardest samples are less redundant for generalization than samples from any other difficulty interval. It suggests that structural differences in the nature of the CIFAR datasets and TinyImagenet dataset exist. A lack of class diversity and dataset complexity could be a possible explanation for the partial rejection of Hypothesis 3. Label noise research would profit from introducing new benchmarks with artificial label noise induction on more complex datasets than CIFAR-100, such as TinyImagenet or Imagenet. Currently, the only datasets of higher complexity that are being studied in the literature contain label noisy naturally, and are therefore only evaluated for one fix, but unknown noise rate that is estimated to be 20% for WebVision [42] and 37.5% for Clothing1M [70].

While the development of noise robust methods for new benchmarks on more complex datasets with varying artificial noise rates (e.g., TinyImagenet or Imagenet) could benefit from knowledge about hard samples with higher informative value, the existing benchmarks on CIFAR-10 and CIFAR-100 do not profit from explicit knowledge on hard samples. Due to the indifference of informative value over all levels of difficulty, a focus on a clean subset combined with state-of-the-art semi-supervised learning methods appears to be more promising. Chapter 6 integrates the findings on the informativeness of samples into a modularized approach for competing with state-of-the-art methods for existing benchmarks, CIFAR-10 and CIFAR-100.

6 Application of Discoveries in Noise Robust Training

This section incorporates the observations of sample informativeness into label noise research by illustrating the effectiveness of semi-supervised learning methods under label noise. In particular, a naive two-stage approach for learning from noisy-labeled data is proposed. The observed constant informative value of samples independent of its difficulty for CIFAR-10 and CIFAR-100 indicates a rejection of Hypothesis 1. Due to the equal relevance of samples, the identification of hard samples, which can be easily mistaken with noisy-labeled samples, appears to be redundant. Instead, considering recent advancements in semi-supervised learning, selecting an easy and therefore mostly clean subset that is interpreted as labeled during training of a semi-supervised method seems to be a more effective approach to noise robust learning. Table 6.1 compares state-of-the-art results of supervised and semi-supervised methods with noise robust approaches evaluated under 20% label noise. State-of-the-art methods under partial supervision achieve higher accuracy with less than 20% of the labels than noise robust methods under 20% label noise (i.e., 80% clean samples). If the results on sample informativeness are transferable to semi-supervised settings, Table 6.1 emphasizes that semi-supervised learning techniques trained with easy, mostly clean samples play a crucial role for achieving competitive results under label noise. In order to show the superiority of partial supervision, we follow the naive approach of using the proposed *h-score* rankings to identify easy (i.e., less noisy) samples that are forwarded to a semi-supervised learning method. In particular, due to the lowest relative noise rates among the easy samples (see Figure 4.13), we identify easy samples by *h-scores* that are produced by an EfficientNetB4 optimized with SGD. The SSL backbone is chosen to be EnAET [66], a former state-of-the-art method. Limited computational resources impeded the use of superior methods

Table 6.1: Comparison of supervised, semi-supervised and noise robust state-of-the art methods for CIFAR-10 and CIFAR-100.

Method	*CIFAR-10*			*CIFAR-100*		
	SL	SSL (4k)	Noise (20%)	SL	SSL (10k)	Noise (20%)
EffNet-L2+SAM [19]	99.7	-	-	96.1	-	-
LaplaceNet [55]	-	97.1	-	-	78.9	-
FlexMatch [74]	-	96.1	-	-	81.1	-
EnAET [66]	-	95.8	-	-	73.1	-
ELR [44]	-	-	92.1	-	-	74.7
DivideMix [40]	-	-	96.1	-	-	77.3

such as FlexMatch [74] that were developed more recently.

The lack of benchmarks under artificial label noise for TinyImagenet prevents meaningful conclusions from evaluating the 2-stage approach for this dataset. Therefore, this thesis only follows the standard benchmarks in label noise research and evaluates the proposed approach for CIFAR-10 and CIFAR-100. Further considerations regarding the divergent results of TinyImagenet for sample informativeness and its implication for label noise research are discussed in Chapter 7.

6.1 Experiments with CIFAR-10 and CIFAR-100

The scores of the proposed approach reflect the best result of a broad grid-search with respect to the optimal amount of samples that are considered as labeled during semi-supervised training. Due to limited computational resources the results are reported for only one training run per noise rate. The results have to be understood as an illustration of the potentials of semi-supervision under label noise rather than an empirically optimized method towards the final task of noise robustness. Further, it has to be taken into account that the backbones of the methods are similar but not equal. The non-SSL methods used a ResNet34 backbone [26], the SSL-based methods used a PreAct-ResNet18 [27], while the EnAET is designed to work with a WResNet28-2 [73]. Figure 6.2 shows that our proposed naive two-stage approach is able to outperform any non-SSL noise robust method. The higher the label noise the better our pro-

Table 6.2: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 with symmetric label noise. The results are obtained from the respective paper. The methods SL and ELR are non-SSL, the methods ELR+ and DivideMix are SSL-based. The subscripts under our approach describe the selected amount of samples interpreted as labeled and the corresponding relative noise rate among them.

Dataset	Method	*Symmetric Label Noise*					
		20%	40%	50%	60%	80%	90%
CIFAR-10	SL [67]	89.3	87.1	-	82.8	68.1	-
	ELR [44]	92.1	91.4	-	88.9	80.7	-
	ELR+ [44]	95.8	-	94.8	-	**93.3**	**78.7**
	DivideMix [40]	96.1	-	94.6	-	93.2	76.0
	Ours	**96.5**	95.7	**95.1**	94.1	87.5	9.6
	(labels / relative noise rate)	(40k / 2%)	(30k / 5%)	(25k / 7%)	(10k / 2%)	(5k / 19%)	(5k / 91%)
CIFAR-100	SL [67]	70.4	62.3	-	54.8	25.9	-
	ELR [44]	74.7	68.4	-	60.1	30.3	-
	ELR+ [44]	77.6	-	73.6	-	**60.8**	**33.4**
	DivideMix [40]	77.3	-	**74.6**	-	60.2	31.5
	Ours	**78.1**	74.8	72.9	70.0	53.9	20.9
	(labels / relative noise rate)	(40k / 3%)	(30k / 7%)	(25k / 9%)	(20k / 12%)	(10k / 36%)	(2k / 60%)

posed method is in comparison to the state-of-the-art approaches of this kind, ELR [44] and SL [67]. In particular, the effect is visible for the more complex dataset CIFAR-100 consisting of more classes and less instances per class. Comparing the results with the best noise robust methods that comprise semi-supervised techniques, it becomes apparent that the naive approach is competitive. It outperforms state-of-the-art results under 20% label noise for both CIFAR datasets, and under 50% label noise for CIFAR-10. However, with increasing noise rates the superiority of the methods DivideMix [40] and ELR+ [44] (an SSL enhanced version of the basic framework) is observable. Nonetheless, while the selection procedure of the clean subset is not optimized towards noise robustness and more powerful SSL backbones exist, the results suggest that a key component of modern noise robust methods are semi-supervised learning techniques. In fact, the results emphasize that the key challenge of noise robust learning is to distinguish easy from noisy-labeled samples. A particular identification of hard samples is not necessary. Instead, the task of identifying easy, clean samples without confounding them for noisy-labeled samples plays a crucial role for final performance of semi-supervised methods.

7 Conclusion and Discussion

By taking into account sample-based hardness scores, this thesis provided a new perspective on learning with noisy labels. In particular, the hypothesis that noise robust methods need to be able to distinguish hard from noisy-labeled samples in order to achieve state-of-the-art results was investigated. As one of the necessary conditions, it was proven that samples can be ranked consistently according to their difficulty in implicit curricula of neural networks. However, the second condition that the informative value of a sample correlates with its degree of difficulty was partially rejected. While datasets CIFAR-10 and CIFAR-100 did not show any correlation between sample difficulty and informativeness, the more complex dataset TinyImagenet showed a non-linear correlation. For TinyImagenet it was found that the easiest and the hardest samples are associated to higher informative value for generalization performance than medium hard samples. The implications for noisy label research derived by the observations of this thesis are of two kind.

For methods evaluated on the standard benchmarks CIFAR-10 and CIFAR-100 hard samples do not contribute more to generalization than easy samples. Considering the propensity of neural networks to learn hard and noisy-labeled samples simultaneously at later stages of the training, a particular focus on hard samples is not justifiable by higher informativeness. Instead, identifying easier samples that tend to be less noisy seems to be more beneficial. Hypothesis 1 was rejected for these datasets. Further, the observed indifference in informativeness emphasizes current trends in noise robust learning that make use of semi-supervised learning techniques by considering hard and noisy-labeled samples as unlabeled [40] [44] . Following the observations on sample informativeness for CIFAR-10 and CIFAR-100, this thesis illustrated the effectiveness of semi-supervised learning methods in noise robust training. A naive two-stage learning approach that used the established *h-scores* for clean (i.e., easy) sample detection combined with a semi-supervised learning method that interpreted the easy samples as labeled and the hard samples as unlabeled was proposed. With no explicit strategy for noise robustness it outperformed any non-SSL noise robust approach and produced competitive results in comparison to state-of-the-art SSL-based approaches. The results further underline the superiority of semi-supervised approaches for solving the task of noisy labels for CIFAR-like datasets.

On the other hand, the results of informativeness of hard samples in TinyImagenet suggest that with increasing dataset complexity, sample difficulty has an impact on its informativeness. However, these trends affect both very easy and very hard samples. To better understand these dynamics, more complex datasets such as ImageNet need to be studied in label noise literature. The first indications of structural differences in between easier datasets like CIFAR-10 and CIFAR-100 and TinyImagenet suggest that current standard benchmarks for ablation studies in noise robust learning are not sufficient. Complex datasets like Imagenet need to be additionally studied under varying artificial noise rates. If the observation of higher informativeness of easy and hard samples can be transferred to Imagenet, a re-evaluation of the learning dynamics of state-of-the-art methods might be necessary.

A main direction of future works includes the optimization of the proposed two-stage approach. For example, the semi-supervised stage could be optimized towards noise robustness by replacing the clear split between labeled and unlabeled samples by a continuous approach that considers samples as partially labeled and unlabeled at the same time. Such hybrid soft-assignments could be tied to sample difficulty eventually converging to fully unlabeled samples for samples of very high difficulty. Further improvements could be achieved by replacing the semi-supervised learning backbone EnAET [66] with superior approaches like FixMatch [58] or SimCLR [11], that was initially developed for representation learning. Limiting computational resources impeded an application of these methods. On the other hand, the clean sample detection module could be enhanced by various techniques. As Chen et al. [12] show, most of the memorization of noisy labels happens in the final softmax layer. Inspired by the *prediction depth* proposed by Baldock et al. [4], additional information for detecting clean samples could be derived by elaborating information from the activations of the hidden layer in neural networks. Furthermore, a comprehensive study of a wider range of network families, optimization procedure and regularization strategies could lead to improved *h-score* rankings for selecting easy samples. Another possible technique to improve the selection of easy and clean samples, in particular under high noise rates, consists of an iterative self-cleaning procedure on the subset of easy samples. As Figure 4.12 (left) emphasizes, under 20% label noise the *h-score* is able provide a clear decision boundary between clean and noisy-labeled samples. A self-cleaning procedure under high noise rates would start with building an initial *h-score* ranking over the full dataset. Then, a new *h-score* is produced for the easiest samples only, ideally leading to a better separation of clean and noisy-labeled samples for the subset due to a smaller rate of label noise. Now, at each iteration, while the hardest (mostly noisy-labeled) samples are filtered out from the subset, a small fraction of samples belonging to the next highest difficulty bin from the initial *h-score* ranking is added. The construction of new *h-scores* on the updated subset follows. This procedure can be repeated until empirically determined thresholds are reached. Another improvement of the clean sample detection module could be achieved by a procedure which we call duplicate-sample trick. Additional information for selecting clean samples may be derived by adding copies of samples with a new label that is assigned to the most probable class next to the annotated class. It is assumed that newly added duplicates with correct labels will be learned at earlier stages than their noisy-labeled counterparts. By comparing the *h-score* rank distance of sample-pairs, clean and noisy labels can be identified. This procedure may be applied for samples gathered around the decision boundary between the clean and noisy subset (labeled and unlabeled respectively) passed on to the semi-supervised method.

The other direction of follow up works relate to structures and regularities in computer vision datasets. Besides in-depth studies on the differences between CIFAR datasets and more complex datasets, further experiments on individual class distributions and representativeness could provide a more fundamental understanding of which class properties impede respectively favor generalization. A particular interesting aspect of the informativeness of samples belonging to TinyImagenet that is worth investigating, relates to the observation that, while hard samples have a higher informative value than samples of other degrees of difficulty, their main contribution to better performance is realized on easy samples. A deeper comprehension of this phenomena might reveal interesting insights on class representation hierarchies in deep neural networks, the notion of a difficult sample and its relevance in deep learning curricula. Generally, understanding common benchmark datasets better can have positive impacts on the development of architectures or regularization and optimization procedures in deep learning.

Bibliography

[1] Chirag Agarwal, Daniel D'souza, and Sara Hooker. "Estimating example difficulty using variance of gradients". In: *arXiv preprint arXiv:2008.11600*. 2020.

[2] Devansh Arpit et al. "A closer look at memorization in deep networks". In: *International Conference on Machine Learning*. PMLR, 2017, pp. 233–242.

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. "Learning with pseudo-ensembles". In: *Advances in neural information processing systems*. Vol. 27. 2014, pp. 3365–3373.

[4] Robert JN Baldock, Hartmut Maennel, and Behnam Neyshabur. "Deep Learning Through the Lens of Example Difficulty". In: *arXiv preprint arXiv:2106.09647*. 2021.

[5] Yoshua Bengio et al. "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 41–48.

[6] David Berthelot et al. "Mixmatch: A holistic approach to semi-supervised learning". In: *arXiv preprint arXiv:1905.02249*. 2019.

[7] David Berthelot et al. "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring". In: *arXiv preprint arXiv:1911.09785*. 2019.

[8] Jeff A Bilmes and others. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models". In: *International Computer Science Institute*. Vol. 4. 1998, p. 126.

[9] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. "Distribution density, tails, and outliers in machine learning: Metrics and applications". In: *arXiv preprint arXiv:1910.13427*. 2019.

[10] Pengfei Chen et al. "Understanding and utilizing deep neural networks trained with noisy labels". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 1062–1070.

[11] Ting Chen et al. "Big self-supervised models are strong semi-supervised learners". In: *arXiv preprint arXiv:2006.10029*. 2020.

[12] Yingyi Chen et al. "Boosting Co-teaching with Compression Regularization for Label Noise". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2688–2692.

[13] *CIFAR website*. URL: `https://www.cs.toronto.edu/~kriz/cifar.html`.

[14] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning*. Vol. 20. Publisher: Springer. 1995, pp. 273–297.

[15] Ekin D Cubuk et al. "Autoaugment: Learning augmentation policies from data". In: *arXiv preprint arXiv:1805.09501*. 2018.

[16] Ekin D Cubuk et al. “Randaugment: Practical automated data augmentation with a reduced search space”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.

[17] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[18] Edgar C Fieller, Herman O Hartley, and Egon S Pearson. “Tests for rank correlation coefficients. I”. In: *Biometrika*. Vol. 44. Publisher: JSTOR. 1957, pp. 470–481.

[19] Pierre Foret et al. “Sharpness-aware minimization for efficiently improving generalization”. In: *arXiv preprint arXiv:2010.01412*. 2020.

[20] Aritra Ghosh, Himanshu Kumar, and PS Sastry. “Robust loss functions under label noise for deep neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. Issue: 1. 2017.

[21] Yves Grandvalet, Yoshua Bengio, and others. “Semi-supervised learning by entropy minimization.” In: *CAP*. Vol. 367. 2005, pp. 281–296.

[22] Sheng Guo et al. “Curriculumnet: Weakly supervised learning from large-scale web images”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 135–150.

[23] Bo Han et al. “Co-teaching: Robust training of deep neural networks with extremely noisy labels”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper/2018/file/a19744e268754fb0148b017647355b7b-Paper.pdf`.

[24] Jiangfan Han, Ping Luo, and Xiaogang Wang. “Deep Self-Learning From Noisy Labels”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[25] Yan Han et al. “Learning from Noisy Labels via Discrepant Collaborative Training”. In: *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2020.

[26] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[27] Kaiming He et al. “Identity mappings in deep residual networks”. In: *European conference on computer vision*. Springer, 2016, pp. 630–645.

[28] Dan Hendrycks et al. “Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf`.

[29] Tin Kam Ho. “Random decision forests”. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, 1995, pp. 278–282.

[30] Sara Hooker et al. “What do compressed deep neural networks forget?” In: *arXiv preprint arXiv:1911.05248*. 2019.

[31] Lang Huang, Chao Zhang, and Hongyang Zhang. “Self-adaptive training: beyond empirical risk minimization”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

[32] Lu Jiang et al. “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels”. In: *International Conference on Machine Learning*. PMLR, 2018, pp. 2304–2313.

[33] Ziheng Jiang et al. “Characterizing structural regularities of labeled data in overparameterized models”. In: *arXiv preprint arXiv:2002.03206*. 2020.

[34] Alex Krizhevsky, Geoffrey Hinton, and others. “Learning multiple layers of features from tiny images”. In: Publisher: Citeseer. 2009.

[35] Jan M. KÃ¶hler, Maximilian Autenrieth, and William H. Beluch. “Uncertainty Based Detection and Relabeling of Noisy Image Labels.” In: *CVPR Workshops*. 2019, pp. 33–37. URL: `http://openaccess.thecvf.com/content_CVPRW_2019/html/Uncertainty_and_Robustness_in_Deep_Visual_Learning/Kohler_Uncertainty_Based_Detection_and_Relabeling_of_Noisy_Image_Labels_CVPRW_2019_paper.html`.

[36] Samuli Laine and Timo Aila. “Temporal ensembling for semi-supervised learning”. In: *arXiv preprint arXiv:1610.02242*. 2016.

[37] Ya Le and Xuan Yang. “Tiny imagenet visual recognition challenge”. In: *CS 231N*. Vol. 7. 2015, p. 3.

[38] Dong-hyun Lee. “Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”. In.

[39] Kuang-Huei Lee et al. “CleanNet: Transfer Learning for Scalable Image Classifier Training With Label Noise”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. _eprint: 1711.07131. June 2018.

[40] Junnan Li, Richard Socher, and Steven CH Hoi. “DivideMix: Learning with Noisy Labels as Semi-supervised Learning”. In: *International Conference on Learning Representations*. 2019.

[41] Junnan Li et al. “Learning to learn from noisy labeled data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5051–5059.

[42] Wen Li et al. “Webvision database: Visual learning and understanding from web data”. In: *arXiv preprint arXiv:1708.02862*. 2017.

[43] Yuncheng Li et al. “Learning from Noisy Labels with Distillation”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. _eprint: 1703.02391. Oct. 2017.

[44] Sheng Liu et al. “Early-Learning Regularization Prevents Memorization of Noisy Labels”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

[45] Ilya Loshchilov and Frank Hutter. “Fixing weight decay regularization in adam”. In: 2018.

[46] Ilya Loshchilov and Frank Hutter. “Online batch selection for faster training of neural networks”. In: *arXiv preprint arXiv:1511.06343*. 2015.

[47] Eran Malach and Shai Shalev-Shwartz. “Decoupling ”when to update” from ”how to update””. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/58d4d1e7b1e97b258c9ed0b37e02d087-Paper.pdf`.

[48] Karttikeya Mangalam and Vinay Uday Prabhu. “Do deep neural networks learn shallow learnable examples first?” In: 2019.

[49] Takeru Miyato et al. “Virtual adversarial training: a regularization method for supervised and semi-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence*. Vol. 41. Publisher: IEEE. 2018, pp. 1979–1993.

[50] Duc Tam Nguyen et al. “SELF: Learning to Filter Noisy Labels with Self-Ensembling”. In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=HkgsPhNYPS`.

[51] Kento Nishi et al. “Augmentation strategies for learning with noisy labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8022–8031.

[52] Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang. “Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels”. In: _eprint: 1705.01936. 2017.

[53] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. “Mutual exclusivity loss for semi-supervised deep learning”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1908–1912.

[54] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. “Regularization with stochastic transformations and perturbations for deep semi-supervised learning”. In: *Advances in neural information processing systems*. Vol. 29. 2016, pp. 1163–1171.

[55] Philip Sellars, Angelica I Aviles-Rivero, and Carola-Bibiane SchÃ¶nlieb. “LaplaceNet: A Hybrid Energy-Neural Model for Deep Semi-Supervised Classification”. In: *arXiv preprint arXiv:2106.04527*. 2021.

[56] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556*. 2014.

[57] Leslie N Smith and Nicholay Topin. “Super-convergence: Very fast training of neural networks using large learning rates”. In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Vol. 11006. International Society for Optics and Photonics, 2019, p. 1100612.

[58] Kihyuk Sohn et al. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *arXiv preprint arXiv:2001.07685*. 2020.

[59] Guocong Song and Wei Chai. "Collaborative Learning for Deep Neural Networks". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 1832–1841. URL: `http://papers.nips.cc/paper/7454-collaborative-learning-for-deep-neural-networks.pdf`.

[60] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research*. Vol. 15. Publisher: JMLR. org. 2014, pp. 1929–1958.

[61] Sainbayar Sukhbaatar et al. "Training convolutional networks with noisy labels". English (US). In: Jan. 2015.

[62] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[63] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[64] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *arXiv preprint arXiv:1703.01780*. 2017.

[65] Mariya Toneva et al. "An empirical study of example forgetting during deep neural network learning". In: *arXiv preprint arXiv:1812.05159*. 2018.

[66] Xiao Wang et al. "EnAET: A self-trained framework for semi-supervised and supervised learning with ensemble transformations". In: *IEEE Transactions on Image Processing*. Vol. 30. Publisher: IEEE. 2020, pp. 1639–1647.

[67] Yisen Wang et al. "Symmetric cross entropy for robust learning with noisy labels". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 322–330.

[68] Hongxin Wei et al. "Combating noisy labels by agreement: A joint training method with co-regularization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13726–13735.

[69] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. "When Do Curricula Work?" In: *arXiv preprint arXiv:2012.03107*. 2020.

[70] Tong Xiao et al. "Learning from massive noisy labeled data for image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2691–2699.

[71] Qizhe Xie et al. "Unsupervised data augmentation for consistency training". In: *arXiv preprint arXiv:1904.12848*. 2019.

[72] Kun Yi and Jianxin Wu. “Probabilistic end-to-end noise correction for learning with noisy labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7017–7025.

[73] Sergey Zagoruyko and Nikos Komodakis. “Wide residual networks”. In: *arXiv preprint arXiv:1605.07146*. 2016.

[74] Bowen Zhang et al. “FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling”. In: _eprint: 2110.08263. 2021.

[75] Hongyi Zhang et al. “mixup: Beyond Empirical Risk Minimization”. In: *International Conference on Learning Representations*. 2018. URL: `https://openreview.net/forum?id=r1Ddp1-Rb`.

[76] Zhilu Zhang and Mert Sabuncu. “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf`.

Appendix

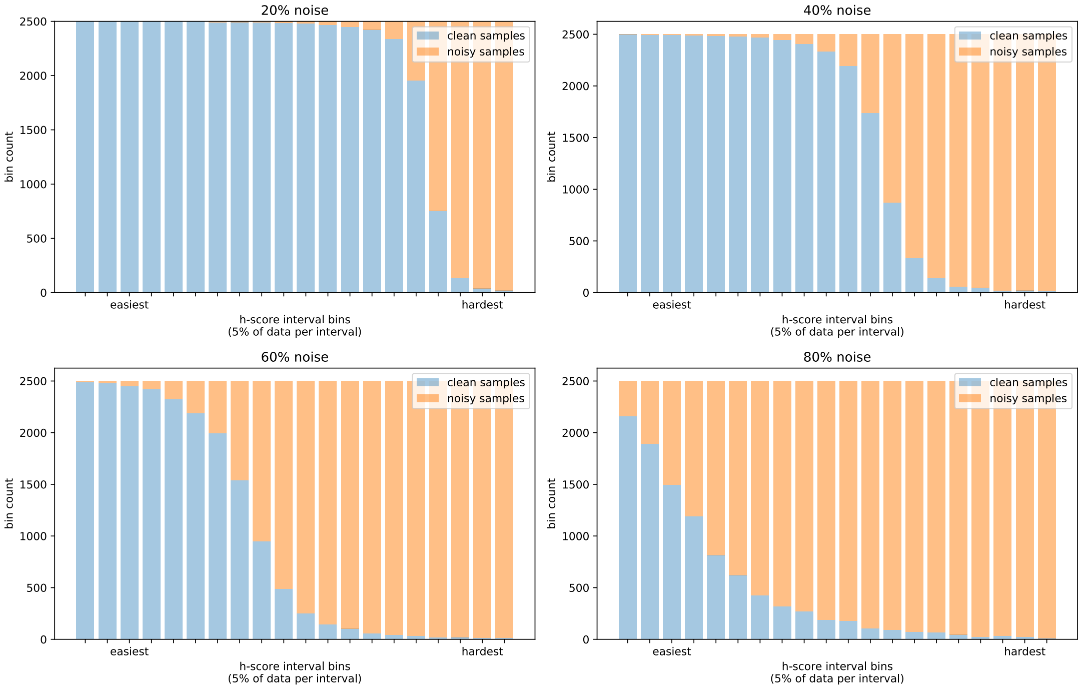


Figure A1: For the CIFAR-10 dataset, The distribution of clean and noisy-labeled samples to difficulty bins under different noise rates for EfficientNetB4 and SGD optimizer.

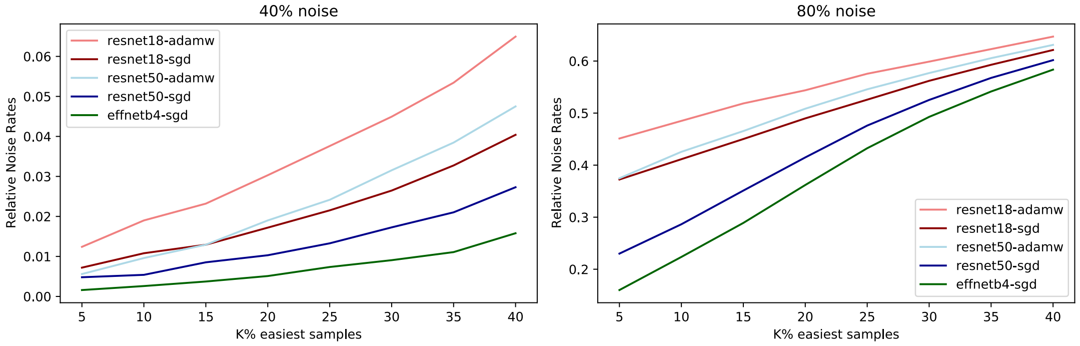


Figure A2: For the CIFAR-100 dataset, relative noise rates among top K% easiest samples according to *h-score*. **Left:** under 40% label noise. **Right:** under 80% label noise.

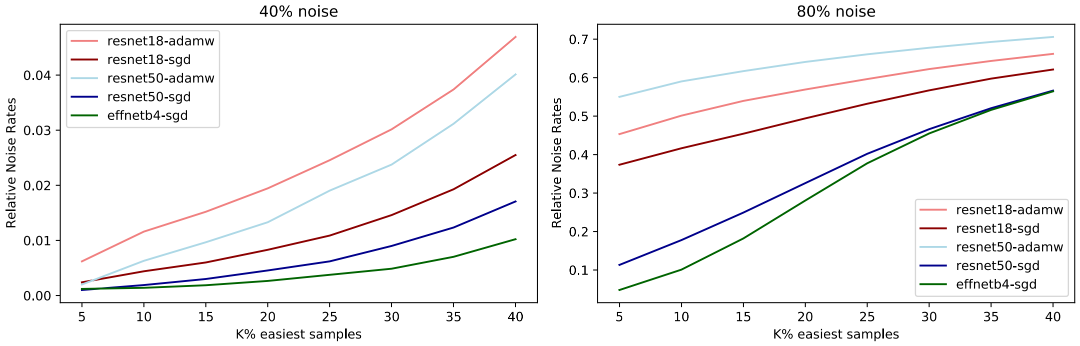


Figure A3: For the TinyImagenet dataset, relative noise rates among top K% easiest samples according to *h-score*. **Left:** under 40% label noise. **Right:** under 80% label noise.

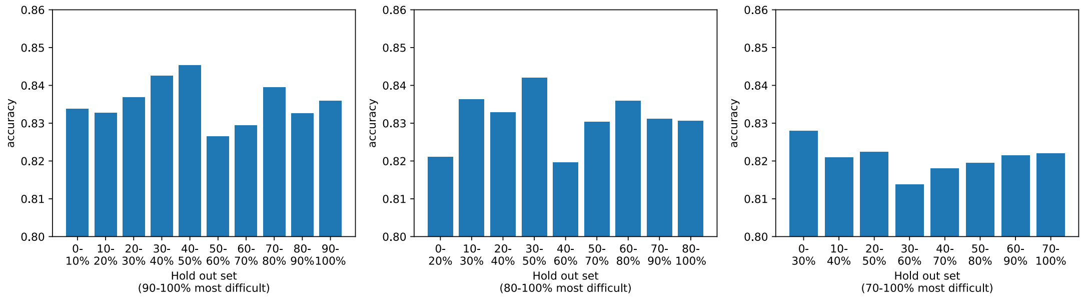


Figure A4: For the CIFAR-10 dataset, *c-score* for best setup with ResNet18 and SGD, set accuracy for train sets with hold-out intervals of 10% of the train set (**left**), 20% of the train set (**middle**) and 30% of the train set (**right**).

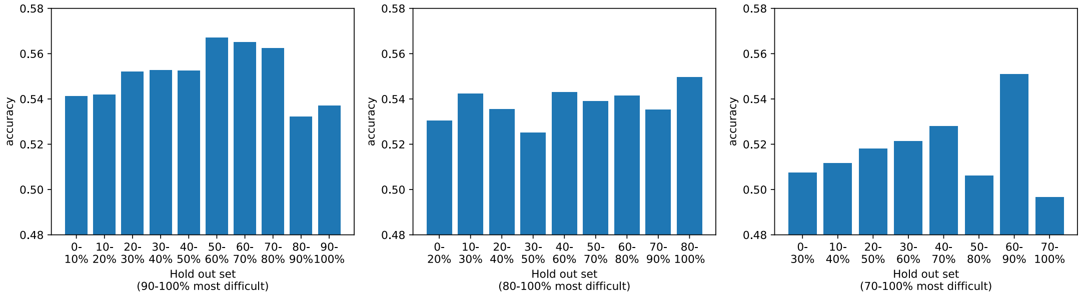


Figure A5: For the CIFAR-100 dataset, *c-score* for best setup with ResNet18 and SGD, set accuracy for train sets with hold-out intervals of 10% of the train set (**left**), 20% of the train set (**middle**) and 30% of the train set (**right**).