The Effects of Regularization and Data Augmentation are Class Dependent

Anonymous Author(s)
Affiliation
Address
email

Abstract

Regularization is a fundamental technique to prevent over-fitting and to improve generalization performances by constraining a model's complexity. Current Deep Networks heavily rely on regularizers such as Data-Augmentation (DA) or weight-decay, and employ structural risk minimization, i.e. cross-validation, to select the optimal regularization hyper-parameters. In this study, we demonstrate that techniques such as DA or weight decay produce a model with a reduced complexity that is unfair across classes. The optimal amount of DA or weight decay found from cross-validation over all classes leads to disastrous model performances on some classes e.g. on Imagenet with a resnet50, the "barn spider" classification test accuracy falls from 68% to 46% only by introducing random crop DA during training. Even more surprising, such performance drop also appears when introducing uninformative regularization techniques such as weight decay. Those results demonstrate that our search for ever increasing generalization performance —averaged over all classes and samples— has left us with models and regularizers that silently sacrifice performances on some classes. This scenario can become dangerous when deploying a model on downstream tasks e.g. an Imagenet pre-trained resnet50 deployed on INaturalist sees its performances fall from 70% to 30% on class #8889 when introducing random crop DA during the Imagenet pre-training phase. Those results demonstrate that finding a correct measure of a model's complexity without class-dependent preference remains an open research question.

1 Introduction

Machine learning and deep learning aim at learning systems to solve as accurately as possible a given task at hand [LeCun et al., 1998, Bishop and Nasrabadi, 2006, Jordan and Mitchell, 2015]. This process often takes the form of (i) being given a finite dataset, a (differentiable) loss function, and a performance measure, (ii) splitting the dataset into train/valid/test sets to optimizing the system's parameters e.g. from gradient updates of the loss on the train set while cross-validating hyper-parameters using the valid set, and (iii) assessing the system's performance on the test set. **As the training set is finite, and the optimal design of the system is unknown, it is common to employ regularization during the optimization phase to reduce over-fitting** [Tikhonov, 1943, Tihonov, 1963] i.e. to decrease the system's performance gap between train set and test set samples [Simard et al., 1991, Chapelle et al., 2000, Bottou, 2012, Neyshabur et al., 2014]. Central to our study is the fact that hyper-parameter selection is done via cross-validation by maximizing the valid set performance with ad-hoc statistics e.g. the average accuracy over all samples for classes in classification tasks.

Cross-validation commonly involves many different types of regularization along with their "strengths" [Goodfellow et al., 2016, He et al., 2021]. Most variants of regularization take one of two forms:

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

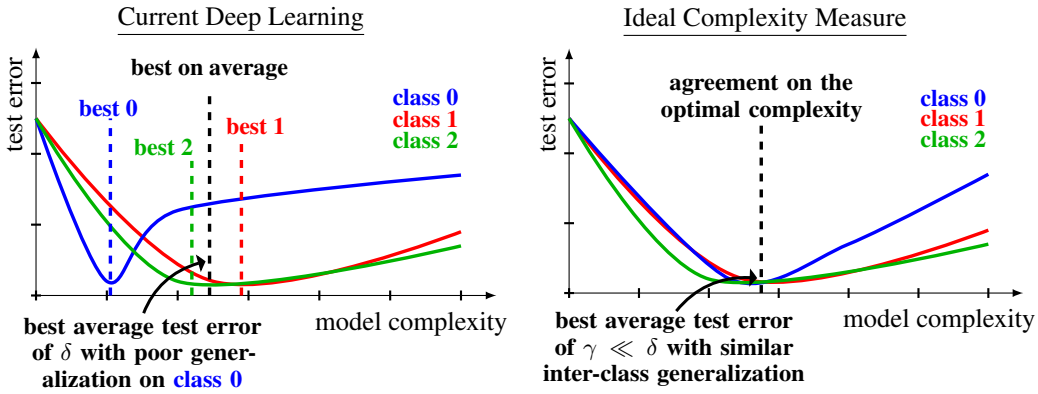

Figure 1: Structural risk minimization minimizes the empirical risk of several models with varying complexity, and selects the one offering the best compromise between under-fitting and over-fitting [Vapnik and Chervonenkis, 1974]. In deep learning, controlling the model's complexity is achieved by picking different DN architectures and/or by applying different levels and flavors of regularization. The key observation of our study is that **when the model complexity is calibrated by a regularizer such as DA (see figs. 2, 5 and 6) or weight-decay (see fig. 3), the class-conditional empirical risks do not align between classes i.e. cross-validation produces models that perform really well on the majority of classes but arbitrarily poorly on the others** as depicted on the left-hand-side. In an ideal setting where the control of the model's complexity is well aligned with the task and model, one would observe the right-hand-side scenario where the same model complexity is optimal for all classes.

Data-Augmentation (DA) and weight-decay. DA is a data-driven and informed regularization strategy that artificially increase the number of training samples [Shorten and Khoshgoftaar, 2019]. As opposed to most *explicit* regularizers e.g. Tikhonov regularization [Krogh and Hertz, 1991], also denoted as weight decay, DA's regularization is *implicit* as it is not a function of a model's parameter, but a function of the training samples [Neyshabur et al., 2014, Hernández-García and König, 2018, LeJeune et al., 2019]; although some DA strategies can be turned into explicit regularizers Balestriero et al. [2022]. Nevertheless, a key distinction between DA and weight decay is that DA tends to requires more domain knowledge to be successful than weight decay. Most —if not all— of current state-of-the-art employ such regularizers [Huang et al., 2018, Chen et al., 2020b, Liu et al., 2021, Tan and Le, 2021, Liu et al., 2022].

In this paper, we will demonstrate that **when cross-validation is employed to select the regularization settings maximizing the validation performance, a significant bias is introduced into the trained model: the regularized model exhibits strong per-class favoritism i.e. while the average test performance is improved, it is at the cost of producing a model with significant performance drop on some of the classes** as illustrated in the schematic of fig. 1. For readers familiar with statistical estimation results e.g. the bias-variance trade-off [Kohavi et al., 1996, Von Luxburg and Schölkopf, 2011] or bayesian estimation e.g. Tikhonov regularization [Box and Tiao, 2011, Gruber, 2017], it should not be surprising that regularization produces bias (more details and background provided in appendix A). In fact, it is beneficial to introduce bias through regularization if it results in a significant reduction of the estimator variance —when one minimizes the average empirical risk. However, the potentially dangerous effect of regularization that this study brings forward is that **the bias introduced by regularization is class-dependent, including on transfer learning tasks**.

To thoroughly validate this observation, we propose a variety of controlled experiments in section 2. First, we carefully quantify the impact of DA, weight decay and dropout on the per-class performance of a model in sections 2.1 and 2.2, demonstrating that current deep learning finds itself in the scenario depicted on the left of fig. 1. Then, we consider the task of transfer learning in section 2.3 where it is again possible to identify again a per-class bias on the target dataset even though the regularization was applied on a different (source) training set. This latter scenario is particularly relevant in current times where it is common to deploy a large pre-trained model on a variety of tasks and raise an important issue: selecting the —on average— best performing pre-trained model can lead to catastrophic individual class performance even on for different downstream tasks.

Our next section 3 will aim at exploring possible explanations and solutions. First, we will provide a brief theoretical justification on why and when DA can be the cause of model bias (section 3.1)

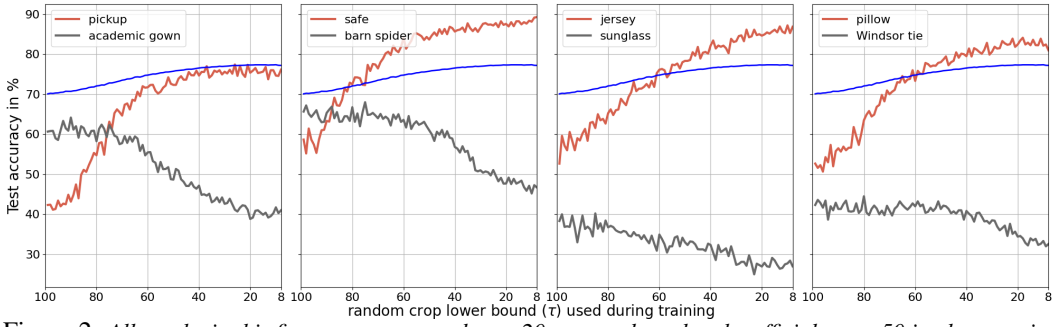

Figure 2: *All results in this figure are averaged over 20 runs and employ the official resnet50 implementation from PyTorch that we trained on Imagenet with horizontal flip and varying lower bound for the random crop DA*. During training, the lower bound on the amount of the original image kept in the random crops (x-axis) varies from 100% to 8% (commonly employed value), the upper bound is always 100%, and the test images are obtained from center crops. We observe that **training the same architecture but with varying random crop parameter (lower bound) provides greater average test accuracy (blue) but makes the per-class performance fall for some of the classes**. Samples for each class are provided in fig. 11, in the appendix.

regardless of the task and data at hand. This will shed light to a first possible issue: the DA parameters that make the transformed input preserve its label information vary depending on the class underlying statistics. In short, DA silently introduces class-imbalance in the training set. We propose a dedicated analysis of the label-preserving property of DA on different classes and models in section 3.2. We then take on the task of searching for a possible solution by first reviewing known theoretical studies quantifying the interplay between regularization and bias in section 3.3. Lastly, we propose some solutions of our own in section 3.4 built from the gained insights of section 3.2 using label-distillation and adaptive DA.

All the codebase used to train the various models and to generate the figures will be publicly released upon completion of the review process along with the pre-trained weights of all the models. For the reviewers, we currently provide in the supplementary material all summary statistics in .json files that are sufficient to reproduce the figures of the study and further explore the interplay between regularization and class-dependent bias.

2 Maximizing the Average Model Performance by Cross-Validation Silently Produce Poor Final Performances on a Minority of the Classes

The first part of our focuses on providing extensive empirical validation of fig. 1. We precisely quantify the amount of class-dependent bias caused by DA in various realistic scenarios in section 2.1; and extend our analysis to weight decay and dropout in section 2.2. Finally, we conclude by demonstrating how the bias introduced by regularization transfers to downstream tasks e.g. when deploying an Imagenet (*source*) trained model on the INaturalist (*target*) dataset in section 2.3; that scenario is key as it demonstrates the potential harm of selecting the best performing model —on average— on the source dataset which could turn out to also be the most biased model on the target dataset class of interest. In fact, it is crucial to remember that regularization, or any other form of structural risk minimization, improves generalization performances by increasing the bias of the estimator so that the estimator's variance is decreased by a greater amount. However, nothing guarantees the fairness of this bias i.e. for it to be equally distributed amongst the dataset classes.

2.1 Precisely Measuring the Per-Class Effect of Data-Augmentation with Controlled Experiments

This section aims at quantifying precisely the amount of downward or upward per-class performance shift that came as a result from using DA. We thus propose a sensitivity analysis by training a large collection of models with varying DA policies to precisely assess the relation between DA and class-dependent model bias.

DA samples have been known to sometimes disregard the semantic information of the original samples [Krizhevsky et al., 2012]. Nevertheless, DA remains applied universally, and fearlessly across tasks and datasets [Shorten and Khoshgoftaar, 2019] as it provides significant performance improvements, even in semi-supervised and unsupervised settings Guo et al. [2018], Xie et al. [2020], Misra and Maaten [2020]. First, we propose in fig. 2 a sensitivity analysis by training the same

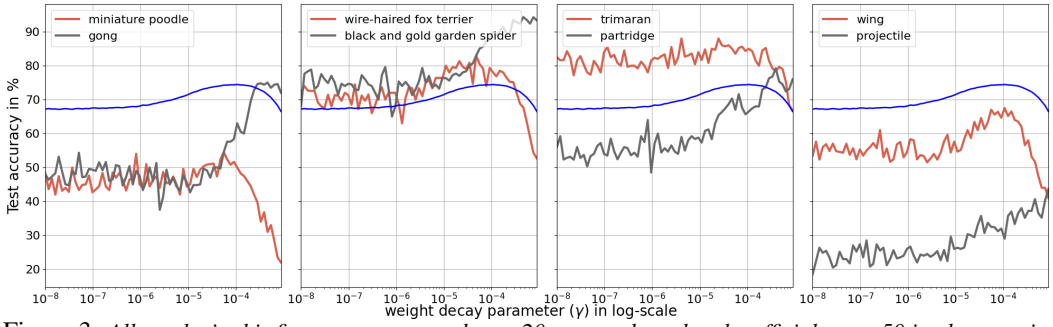

Figure 3: *All results in this figure are averaged over 20 runs and employ the official resnet50 implementation from PyTorch that we trained on Imagenet with varying weight decay parameter.* A very surprising result that we report here is that **the class-dependent bias that we observed from DA also occurs with one of the most fundamental and uninformed regularizer: weight decay**. We report the per-class performance when only employed horizontal flip as DA and a varying weight decay parameter, and we observe clear distinct behaviors between different classes. Samples for each class are provided in fig. 12, in the appendix. We provide the same figure for DenseNet121 model in fig. 15.

architecture on Imagenet with varying DA policies. In particular, we consider a given DA (random crop in this case) and we vary the support of the parameter α which represents how much of the original image is kept in the crop. We train DNs using $\alpha \in [100, \tau]$ with τ varying from 100 to 8 and for each case, we report our metrics averaged over 20 trained models. We observe a clear relation between **increase in the strength of the DA, increase in the average test accuracy overall classes, and decrease in some per-class test accuracies**. For example, on a resnet50 Imagenet setting, the accuracy on the "academic gown" class goes from 62% to 40% steadily as τ decreases. We defer the same experiment but using weight decay in the next section 2.2.

To further convey our claim, we now propose a formal statistical test [Neyman and Pearson, 1933, Fisher, 1955] on the hypothesis that the per-class accuracy is significantly lower when DA is applied for those classes. A test of significance is a formal procedure for comparing observed data with a claim or hypothesis. In our case we aim to test if the mean accuracy on class y of a model trained without DA is greater than the one obtained with DA. Due to the stochastic optimization process, this accuracy is a random variable even when the dataset is not changed. Hence we define that random variable by $\mathrm{Accu}_y(\mathbb{X})$ and our null hypothesis by $H_0 = \mathbb{E}[\mathrm{Accu}_y(\mathbb{X})] < \mathbb{E}[\mathrm{Accu}_y(\mathbb{X}_{\mathrm{DA}})]$ with $\mathbb{X}$ the original dataset and $\mathbb{X}_{\mathrm{DA}}$ the DA dataset using random crop with $\alpha \in [100, 8]$. A one-sided t-test is used and the form that does not assume equal variances is known as Welch's t-test [Welch, 1947] with statistic $t = \frac{\hat{\mu}(\mathbb{X}_1) - \hat{\mu}(\mathbb{X}_2)}{\sqrt{\hat{\sigma}^2(\mathbb{X}_1) - \hat{\sigma}^2(\mathbb{X}_2)}}$, and with degrees of freedom $\nu = \left(\frac{\hat{\sigma}^2(\mathbb{X}_1)}{N_1} + \frac{\hat{\sigma}^2(\mathbb{X}_2)}{N_2}\right)^2 \left(\frac{\hat{\sigma}^4(\mathbb{X}_1)}{N_1^2(N_2-1)} + \frac{\hat{\sigma}^4(\mathbb{X}_2)}{N_2^2(N_1-1)}\right)^{-1}$. We obtain that there is enough evidence to reject H_0 with 95% confidence for 4.4% of all the classes, and with 99% confidence for 2.1% of all the classes. Hence there is sufficient evidence to say that the per-class test accuracies is not increased when introducing DA for 4.4% of the 1000 Imagenet classes. Note that one could formulate this problem in term of average treatment effect, where the treatment is the application of DA and the outcome is the accuracy on class y of the trained model. Doing so, one could measure the per-sample bias from the Conditional Treatment Effect, however we limit ourselves to a per-class study and leave such fine-grained analysis for future work. The next section 2.2 proposes to reproduce those experiments but considering weight decay as a regularizer instead of DA.

2.2 Uninformed Weight Decay and Dropout Also Create Class-Dependent Model Bias

We quantified in section 2.1 how much per-class bias was produced by DA. As per the arguments given in sections 3.1 and 3.2, it would be natural to assume that what makes DA responsible for creating class-dependent bias in DNs is our misfortune in defining correct augmentation policies, and thus, that other regularization techniques that are uninformed e.g. weight decay would behave differently. The goal of this section is to demonstrate that weight decay also suffers from the same class-dependent behavior indicating that designing a regularizer for DNs that is fair across classes might require novel innovative solutions.

The basic formulation of weight-decay consists in having any loss to be minimized $\mathcal{L}$ and add to it an additional term denoted as $\gamma\|\theta\|_2^2$ where θ collects the model's parameters and γ is the strength of the regularization (proportional to the restriction on the trained model complexity). In general, we do not

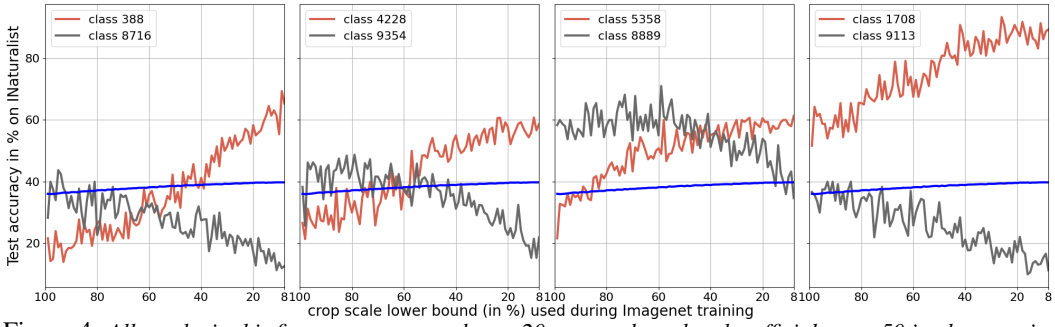

Figure 4: *All results in this figure are averaged over 20 runs and employ the official resnet50 implementation from PyTorch that we pre-trained on Imagenet with varying random crop lower bound and then transferred to INaturalist with frozen backbone weights (only the linear classifier is tuned).* We observe that even when the DA is applied on a different dataset during a pre-training phase, the trained model incorporates an inherent bias that transfers to downstream tasks i.e. **when deploying a model in a transfer learning task, selecting the one with best average test accuracy on the source dataset might result in deploying a model with the most biased against the classes of interest in the target dataset**. Samples for each class are provided in fig. 16, in the appendix.

incorporate the bias term(s) within θ as this would directly remove the ability of the model to learn the natural class prior [Hastie et al., 2009]. As a result, and throughout this study, we consider θ to incorporate all the DN parameters except for the ones of batch-normalization layers, as commonly done in deep learning [Leclerc et al., 2022]. We report in fig. 3 the per-class performance of a resnet50 trained on Imagenet with varying weight decay coefficient γ (as was done for DA in fig. 2) and we observe that different classes have different test accuracy sensitivities to variations in γ. Some will see their generalization performance increase, while others will have decreasing generalization performances. That is, **even for uninformative regularizers such as weight decay a per-class bias is introduced, reducing performances for some of the classes**. Although weight-decay is one of the most popular regularizer that is uninformed on the data and task at hand, recent studies have demonstrated that techniques such as model pruning —which can be seen as a post-training model complexity reduction i.e. regularization— also produce increased bias towards under-represented features [Hooker et al., 2019, 2020]. We further confirm such findings in figs. 8 and 9 for dropout where the same per-class trend is observed.

The next section 2.3 proposes to study the scenario of introducing a pre-trained model, on a different downstream task. We will demonstrate that even in this setting, the class-dependent effect of our current regularizers remains present and unfair towards specific downstream classes.

2.3 The Class-Dependent Bias Transfers to Other Downstream Tasks

The last experiment we propose is to quantify the amount of class-dependent bias that transfers to other downstream tasks, a common situation in transfer learning and in system deployment to the real world [Pan and Yang, 2009]. We thus want to measure how regularization applied during the pre-training phase on a *source* dataset impacts the per-class accuracy of that model on the *target* dataset.

In order to keep the setting similar to section 2.1, we adopt a resnet50 model with random crop DA. That model is pre-trained on Imagenet dataset (source) with varying value of τ (random crop lower bound) and then, the trained model is transferred to the INaturalist dataset [Van Horn et al., 2018] (target) that consists of 10,000 classes. When transferring the model to INaturalist, the parameters are kept frozen, and only a linear classifier is trained on top of it. We report in fig. 4 the performance of the trained models with varying τ on different INaturalist classes. We observe once again that the best resnet50 —on average— is not necessarily the one that should be deployed as there exists a strong per-class bias that varies with τ. As a result, picking the best performing model from a source dataset to a target dataset, might leave the pipeline to perform poorly since that model might also be the one that is the most biased against the class of interest in the target dataset.

This result should motivate the design of novel regularizers that do not reduce performances between classes at different regimes. Additionally, due to the cost of training multiple models with varying regularization settings, one might wonder on the possible alternative solutions to detect trends such as shown in fig. 4 only when given a single pre-trained model.

3 Understanding Why and When Regularization Produces Models With Class-Dependent Preferences

The first part of our study from section 2 empirically validated that current regularization methods do not produce the same amount of complexity control over different classes, resulting in some classes becoming poorly predictable from a model that has been trained with regularization hyper-parameters maximizing the average valid set performances. We now provide in sections 3.1 and 3.2 some intuition on why DA can be a source of bias regardless of the task, dataset and model at hand. Then, section 3.3 reviews existing works trying to confront regularization and model bias, and as we will see, there does not exist yet a unified solution nor analysis that not only explains but also corrects the behavior observed in section 2. Nevertheless, we attempt in section 3.4 to explore a few possible solutions.

3.1 A Data-Augmentation Policy That is Not Label-Preserving For All Classes Will Create Class-Imbalance Performances

To provide a simple explanation on how DA causes bias in a trained model, we propose the following derivation that holds for any signal e.g. timeseries, images, videos. Without loss of generality [Hui and Belkin, 2020] we will consider here the ℓ_2 loss, although the same derivation carries out with any desired metric.

Dataset notations. Given a sample $\boldsymbol{x} \in \mathcal{X}$ with $\mathcal{X} \subset \mathbb{R}^D$, we consider $\boldsymbol{y} \triangleq f^*(\boldsymbol{x})$ to be the ground-truth target value. Hence our hope is to learn an approximator f_θ that is as close as possible to f^* everywhere in $\mathcal{X}$, although we only observe a finite training set $\mathbb{X} \triangleq \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \dots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$. **Data-Augmentation notations.** Additionally, one employs a DA policy $\mathcal{T} : \mathbb{R}^D \times \mathcal{K} \mapsto \mathbb{R}^D$ such that given a transformation parameter $\alpha \in \mathcal{K}$, $\mathcal{T}_\alpha(\boldsymbol{x})$ produces the transformed view of $\boldsymbol{x}$. Often, one also defines a density p on $\mathcal{K}$ that helps in sampling transformation parameters that are a priori known to be the most useful.

Theorem 1. Whenever the transformations produced by $\mathcal{T}_\alpha, \forall\alpha$ do not respect the level-set of f^*, and whenever the model has enough capacity to minimize the training loss, the DA will create irreducible bias in f_θ as in

$$\underbrace{\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{X}} \mathbb{E}_\alpha \left[\|\boldsymbol{y} - f^*(\mathcal{T}_\alpha(\boldsymbol{x}))\|_2^2\right] > 0}_{=\,0 \text{ iff the DAs of } \boldsymbol{x} \text{ are on the same level-set of } f^*} \text{ and } \underbrace{\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{X}} \mathbb{E}_\alpha \left[\|\boldsymbol{y} - f_\theta(\mathcal{T}_\alpha(\boldsymbol{x}))\|_2^2\right] = 0}_{\text{zero training error}} \Longrightarrow \text{ biased } f_\theta. \quad (1)$$

The main idea of the proof, provided in appendix B, is to show that if a transformation does not move samples on the level-set of the true function (left-hand-side of eq. (1)), then f_θ will learn a different level (since it has 0 training error), and thus $\|f^* - f_\theta\| > 0$ i.e. f_θ is biased regardless of the training set.

Whenever the left-hand-side of eq. (1) is 0, the DA is denoted as *label-preserving* [Cui et al., 2015, Taylor and Nitschke, 2018]. From the above, we see that **unless the target $\boldsymbol{y}$ associated to $\mathcal{T}_\alpha(\boldsymbol{x})$ is modified accordingly to encode the shift in the target function level-set produced by $\mathcal{T}_\alpha$, any DA that is not label-preserving will introduce a bias**. Some DAs propose to incorporate label transformation i.e. not only $\boldsymbol{x}$ but also $\boldsymbol{y}$ is augmented to better inform on the uncertainty that has been added into $\mathcal{T}_\theta(\boldsymbol{x})$. This is for example the case for MixUp [Zhang et al., 2017], ManifoldMixUp [Verma et al., 2019], CutMix [Yun et al., 2019] and their extensions.

Our goal in the next section 3.2 is to demonstrate how DAs such as random crop, color jittering, or CutOut are only label preserving for some values of α that vary with the sample class. As a consequence, while the use of the DA improves the average test performance, it is at the cost of a significant reduction in performance for some of the classes.

3.2 The Same Data-Augmentation can be Label-Preserving or Not Between Different Classes Introducing Class-Imbalance During Training

In the previous section 3.1 we provided a general argument on the sufficient conditions for DA to produce a biased model. We hope in this section to provide a more concrete example that applies to current DN training. To that end, we will demonstrate that a DA can be label-preserving or not depending on the sample's class, hence, since the same DA policy is employed for all classes, the

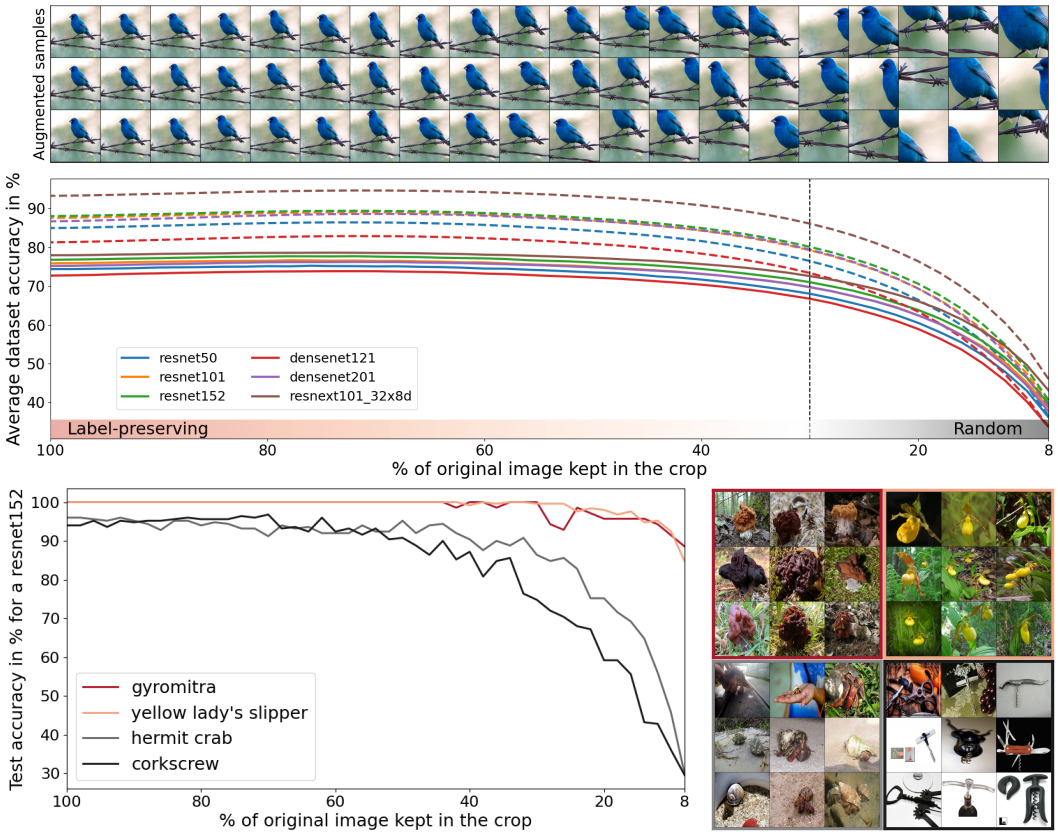

Figure 5: *All results in this figure employ official pretrained models from PyTorch with random crop DA.* We present examples of an augmented image of class "bird" (**top**) along with the average accuracy on the training set (dashed line) and test set (plain line) on Imagenet, using 6 popular architectures (**middle**). The random crop DA seems to loose its label-preserving property when less than 30% of the image is kept in the crop. However, when looking at per-class performances we observe an entirely different story where **random crop DA can be label-preserving with only 8% of the original image for some classes, while for other classes the label information starts to reduce at around 50%** as reported at the bottom along with 9 images of the corresponding classes. The CutOut and ColorJitter cases are presented in figs. 13 and 14 and exhibit the same per-class behaviors.

augmented dataset will exhibit a class-imbalance in favor of the classes for which the DA is most label-preserving.

To measure by how much a given DA, $\mathcal{T}_\alpha$, is label-preserving, we propose to take 6 popular architectures that are pre-trained on Imagenet [Deng et al., 2009] from the official PyTorch [Paszke et al., 2019] repository, and to evaluate their accuracy performances for varying DA settings (top of fig. 5). We observe that when considering the dataset as a whole, it is possible to identify a DA regime as a function of α for which the amount of information present in $\mathcal{T}_\alpha(\boldsymbol{x})$ becomes insufficient to predict the correct label on average. But more interestingly, we also take the per-class accuracy performance (bottom of fig. 5) and observe that for some classes, any level of transformation α can produce augmented samples with enough information to be correctly classified, while other classes see their samples become unpredictable as soon as $\mathcal{T}_\alpha$ moves away from the identity mapping. Note that we report test set performances ensuring that the observed performances are not due to ad-hoc memorization.

To further ensure that the observed relation between label-preservation, sample class, and amount of transformation α is sound, we provide in fig. 6 the per-class test accuracy on different models, all exhibit the same trends. In short, we identify that **when creating an augmented dataset by applying the same DA across classes, the number of per-class samples that actually contain enough information about their true labels will become largely imbalance between classes, even if the original dataset was balanced**. Any model trained on the augmented dataset will thus focus on the classes for which the DA is the most label-preserving.

Figure 6: *All results in this figure employ official pretrained models from PyTorch.* Reprise of the bottom left of fig. 5 for three different DAs (**each column**) and using the same 6 popular architectures (resnet50, resnet101, resnet152, densenet121, densenet201, resnext101-32x8d) (**different lines**). We observe that **across DAs, different architectures agree on the label-preserving regimes for $\mathcal{T}_\alpha$ i.e. even an ensemble of model would not reduce the class-dependent bias of the final prediction**.

Beyond the above intuitive and natural understand we obtained in term of data-augmentation, there exists theoretical studies that we propose to summarize below

3.3 Do Existing Studies Provide Answers Into the Inter-Play Between Regularization and Class-Dependent Bias?

We explored in section 3.2 a possible explanation of the class-dependent bias we observed in section 2 hinting at the need to use class-dependent DA. We propose here to summarize existing studies that are attempting to better understand the impact of DA and regularization in general onto a model's performance and bias. Then, section 3.4 will explore their application.

Input independent DA exacerbates the bias already present in a dataset. Especially relevant to our results is a recent result of Xu et al. [2020]. In this work, it was theorized that when the underlying dataset contains inherent biases, training on the original data is more effective than employing an i.i.d. DA policy, i.e. applying the same random augmentation to all samples/classes, to produce an unbiased model. In short, the DA exacerbates the already present biases and makes the trained model further away from the unbiased optimum. The difficulty of this result lies in defining bias for real images. As per our experiments from section 3, we observe that bias can take many form e.g. one class might naturally represent its object always under the same angle. This is particularly true say for boats which are rarely captured upside-down. Hence, simply having classes with different natural statistics could be enough for Xu et al. [2020] to prohibit the use of DA in current datasets. Furthermore, Raghunathan et al. [2020] obtained a surprising result combining both DA and regularization. In that case, it was found that the minimum norm interpolant on the original + DA dataset could have a larger standard error than the minimum norm interpolant on the original dataset alone, even when using label-preserving DA. However, this phenomenon only occurs as long as the model remains over-parametrized, even when considering the original+DA dataset.

Input dependent DA can reduce sample/class bias. Recall that section 3.2 brought forward one possible explanation on how DA can be the source of training-set by introduce class-imbalance due to the same DA being label-preserving for some classes and not for others. From this, a direct solution would be to adapt the "strength". This solution, formalized in Xu et al. [2020], consists in measuring the bias of a model and adapt the DA policy accordingly to correct it. This has been done in different flavors e.g. in McLaughlin et al. [2015], Iosifidis and Ntoutsi [2018], Jaipuria et al. [2020]. One limitation of this direction is that it requires to estimate a model bias and adapt the DA accordingly which can be challenging for large scale models.

Learned DA e.g. from a GAN can produce even more class-dependent bias. One natural extension of the hands-on adaptivity of a DA to a measured model bias would be to learn a DA to maximize a model's performance, for example. This line of work has led to many learn DA policies e.g. using Generative Adversarial Networks [Hu and Li, 2019]. Yet, it has been shown that learning a DA to maximize some aggregated measure of performance will produce even more bias in a model [Hu and Li, 2019]. In fact, and as per the controlled experiments from section 2, the learned DA will entirely disregard a minority of the classes if it means that the produced DA can drastically improve performances on all others, effectively maximizing the average performances.

Inherent tradeoff between DA and model robustness. In addition to the implication of DA into bias, there exists an intertwined relationship between DA and model robustness. In fact, even assuming the use of perfectly adapted DAs, there exists an *inherent tradeoff* between accuracy and

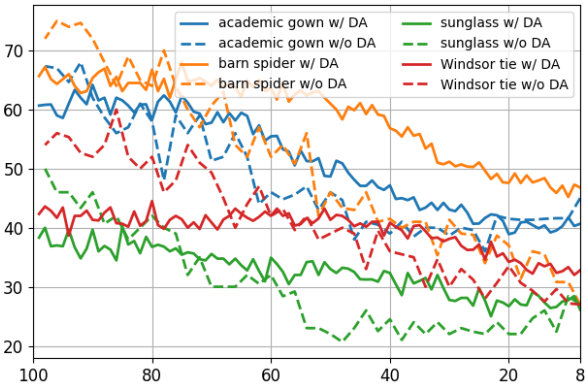

Figure 7: *All results in this figure are averaged over 5 runs and employ the official resnet50 implementation from PyTorch that we trained on Imagenet with horizontal flip and varying lower bound for the random crop DA without applying random crop at all for the* 20 *classes that suffered the most from it i.e. we apply class-dependent DA.* Reprise of fig. 2 but now implementing the class-dependent DA as prescribed in section 3.3. Although such strategy certainly suffices in some cases, we observe here that the bias of the model introduced from random crop spills-over to the classes that do not benefit from that DA, even though such classes never received that DA during training.

robustness that holds even in the infinite data limit [Tsipras et al., 2018, Fawzi et al., 2018, Zhang et al., 2019]. For example, Min et al. [2021] proved in the robust linear classification regime that (i) more data improves generalization in a weak adversary regime, (ii) more data can improve generalization up to a point where additional data starts to hurt generalization in a medium adversary regime, and that (iii) more data immediately decreases generalization error in a strong adversary regime.

3.4 Class-Dependent Data-Augmentation Seems Insufficient for Performance Recovery

We observed in section 2 that DA could lead to disastrous per-class performances on a minority of classes. We now propose to implement one solution from section 3.3 that consists in simply not applying the harmful DA to the classes suffering from it.

Solving the per-class impact of DA is crucial even beyond the per-class harmful effects. In fact, as an illustrative example, a standard resnet50 on Imagenet reaches 77.14% top-1 with a random crop lower bound of 8%. Using cross-validation we could reach 77.29 with a random crop lower bound of 10% on all classes. Yet, and most interestingly, if one were able to get the best per-class performance, one could reach 79.37. We thus explore one solution from section 3.3 that consists in not applying the random crop DA onto the classes whose test performances degrade with it. We report in fig. 7 how such strategy is not sufficient to guarantee that the class performance is not impacted. In fact, we observe that applying this DA only to the classes that benefit from it is enough to bias the model and degrade the performances on some classes at the same pace than when applying the DA to all classes unconditionally.

As a result, we observe that no readily and easily implemented solution provides us with a strategy to prevent deep learning to fall into the scenario depicted on the left of fig. 1. Those findings however motivate the search of novel model complexity controls that is fair among classes. From a more theoretical viewpoint, it might also be possible to better understand if even such a fair per-class model complexity could exist, which is not clear as natural image classes tend to have inherently different statistics.

4 Conclusions

We proposed in this study to understand the impact of regularization, in particular data-augmentation and weight decay, into the final performances of a deep network. We obtained that the use of regularization increases the average test performances at the cost of significant performance drops on some specific classes. By focusing on maximizing aggregate performance statistics we have produced learning mechanisms that can be potentially harmful, especially in transfer learning tasks. In fact, we have also observed that varying the amount of regularization employed during pre-training of a specific dataset impacts the per-class performances of that pre-trained model on different downstream tasks e.g. going from Imagenet to INaturalist. Lastly, commonly prescribed solutions e.g. class-dependent data-augmentation do not seem to help indicating that the sole use of an augmentation on some classes is enough to bias the model on all classes. Hence, there remains a vast research area to explore in order to turn deep learning model selection from the current regime to a more ideal one (left to right of fig. 1).

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Randall Balestriero, Ishan Misra, and Yann LeCun. A data-augmentation is worth a thousand samples: Exact quantification from analytical augmented sample moments. *arXiv preprint arXiv:2202.08325*, 2022.

Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, 13, 2000.

Shuxiao Chen, Edgar Dobriban, and Jane Lee. A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33:21321–21333, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine learning*, 107(3):481–508, 2018.

Ronald Fisher. Statistical methods and scientific induction. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(1):69–78, 1955.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Marvin HJ Gruber. *Improving efficiency by shrinkage: The James-Stein and ridge regression estimators*. Routledge, 2017.

Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. Deep embedded clustering with data augmentation. In *Asian conference on machine learning*, pages 550–565. PMLR, 2018.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*, 2018.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

Mengxiao Hu and Jinlong Li. Exploring bias in gan-based data augmentation for small samples. *arXiv preprint arXiv:1905.08495*, 2019.

Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2752–2761, 2018.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

Maximilian Ilse, Jakub M Tomczak, and Patrick Forré. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR, 2021.

Vasileios Iosifidis and Eirini Ntoutsi. Dealing with bias via data augmentation in supervised learning scenarios. *Jo Bates Paul D. Clough Robert Jäschke*, 24, 2018.

Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa, Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, and Vidya N Murali. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 772–773, 2020.

Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349 (6245):255–260, 2015.

Marius Kloft, Ulf Brefeld, Pavel Laskov, Klaus-Robert Müller, Alexander Zien, and Sören Sonnenburg. Efficient and accurate lp-norm multiple kernel learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL `https://proceedings.neurips.cc/paper/2009/file/d516b13671a4179d9b7b458a6ebdeb92-Paper.pdf`.

Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. ffcv. `https://github.com/libffcv/ffcv/`, 2022.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Daniel LeJeune, Randall Balestriero, Hamid Javadi, and Richard G Baraniuk. Implicit rugosity regularization via data augmentation. *arXiv preprint arXiv:1905.11639*, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. Data-augmentation for reducing dataset bias in person re-identification. In *2015 12th IEEE International conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2015.

Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. In *Uncertainty in Artificial Intelligence*, pages 129–139. PMLR, 2021.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ekachai Phaisangittisagul. An analysis of the regularization between l2 and dropout in single hidden layer neural network. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 174–179. IEEE, 2016.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. *Advances in neural information processing systems*, 4, 1991.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547. IEEE, 2018.

Andrei Nikolajevits Tihonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math.*, 4:1035–1038, 1963.

Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Giorgio Valentini and Thomas G Dietterich. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *Journal of Machine Learning Research*, 5(Jul):725–775, 2004.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

VN Vapnik and A Ya Chervonenkis. The method of ordered risk minimization, i. *Avtomatika i Telemekhanika*, 8:21–30, 1974.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

Ulrike Von Luxburg and Bernhard Schölkopf. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*, volume 10, pages 651–706. Elsevier, 2011.

Bernard L Welch. The generalization of 'student's'problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 1947.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

Yi Xu, Asaf Noy, Ming Lin, Qi Qian, Hao Li, and Rong Jin. Wemix: How to better utilize data augmentation. *arXiv preprint arXiv:2010.01267*, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [N/A] We do not propose a novel method that requires such discussion
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We carefully described why the observations we have made can be dangerous for real world applications
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Our only "theoretical result" consists in a formal statistical test for which we precisely describe our settings and statistics
   (b) Did you include complete proofs of all theoretical results? [N/A] No theoretical result requiring proofs was provided
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include summary statistics in the supplementary material which is enough to validate our claims. The full codebase and all the saved models will be released upon completion of the review process (this includes almost a thousand pre-trained resnet50s)
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specify within each caption the key experimental setups, additional training details are provided in the appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] we averaged all of our runs over a significant amount of realizations (20) and perform a formal statistical test on the significance of our results
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We provide this in the beginning of the appendix. Note that we will be providing the saved models and summary statistics to make the results easily reproducible even without GPU or computation ressources.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] For Imagenet and INaturalist
   (b) Did you mention the license of the assets? [Yes] the full code will be released with a NC by NC license on GitHub
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] we include summary result files that contain all the statistics to reproduce the main claims of the paper in the supplementary material. The full codebase along with pre-trained models will be released upon completion of the review process.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We are only using the Imagenet and INaturalist datasets
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We are only using the Imagenet and INaturalist datasets
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]