The Effects of Regularization and Data Augmentation are Class Dependent

Randall Balestriero[1], Léon Bottou[1], and Yann LeCun[1,2]

[1]Meta AI Research, [2]NYU

{rbalestriero,leonb,ylecun}@fb.com

Codes, full result tables, and pre-trained model weights will be released soon

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 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 designing novel regularizers without class-dependent bias 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 being given a *finite training set* and a *performance measure*, optimizing the system's parameters e.g. from gradient updates, and assessing the system's performance on test set samples, i.e. samples that were not used during the system optimization. **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).

Data-Augmentation (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

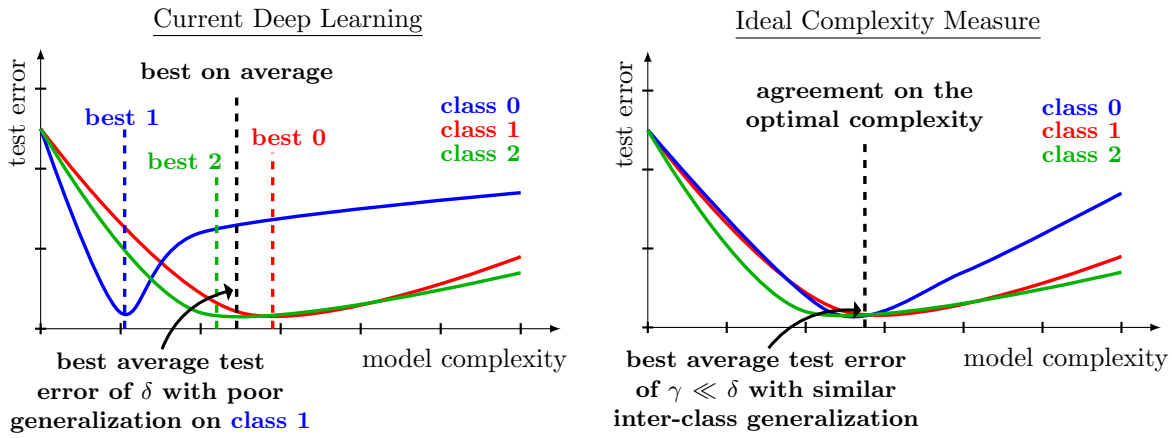

Figure 1: Structural risk minimization minimizes the empirical risk of several models with varying complexity, and selects the one offering the best compromise between under-fitting and over-fitting (Vapnik and Chervonenkis, 1974). In deep learning, controlling the model's complexity is achieved by picking 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 to 4) or weight-decay (see fig. 5), 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.

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 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 employing regularization such as DA or weight decay, a significant bias is introduced into the trained model**. In particular, **the regularized model exhibits strong per-class favoritism i.e. while the average test accuracy over all classes improves when employing regularization, it is at the cost of the model becoming arbitrarily inaccurate on some specific classes** as illustrated in fig. 1. After a brief theoretical justification on why and when DA can be the cause of bias (section 2.1), we propose a dedicated sensitivity analysis of the bias produced by different amounts of DA in sections 2.2 and 2.3, which is followed by a similar study dedicated to weight decay in section 2.4 and transfer learning in section 2.5. *We shall highlight that although we perform a class-level study, it is possible to refine this entire analysis at the sample-level.*

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. In fact, it is often beneficial to introduce bias through regularization if it results in a significant reduction of the estimator variance —when one aims to minimize the average empirical risk. This is one of the main reason behind the success of techniques such as ridge regression. However, what is potentially dangerous is that **the bias introduced by regularization treats classes differently, including on transfer learning tasks** as we will demonstrate in section 2.5. Those observations also support recent theoretical results tying a model's performance to its robustness and to DA, as we discuss in appendix A.

2 Regularization Creates Class-Dependent Model Bias that can be Harmful even for Transfer Learning Tasks

The first part of our study focuses on DA, a technique that regularizes a model by introducing new training samples, derived from the observed ones. 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). We first provide in sections 2.1 and 2.2 some intuition on why DA can be a source of bias regardless of the task, dataset and model at hand. We then quantify the amount of bias caused by DA in various realistic scenarios in section 2.3; and extend our analysis to weight decay in section 2.4. 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 2.5; 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 When Data-Augmentation Creates Bias

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

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

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 $\mathcal{T}_\alpha$,**

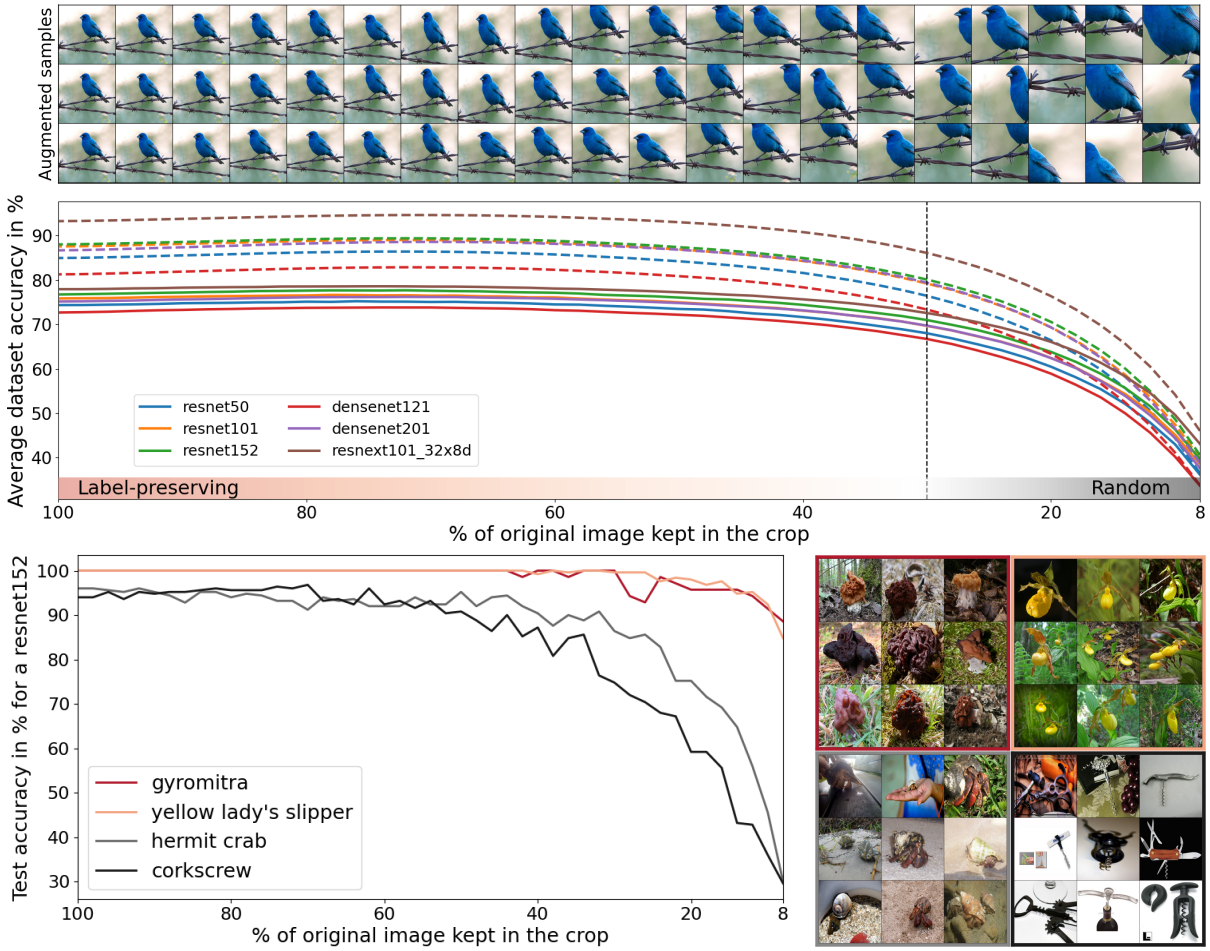

Figure 2: *All results in this figure employ official pretrained models from PyTorch with random crop DA.* We present examples of an augmented image of class "bird" (**top**) along with the average accuracy on the training set (dashed line) and test set (plain line) on Imagenet, using 6 popular architectures (**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. 10 and 11 and exhibit the same per-class behaviors.

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 2.2 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.

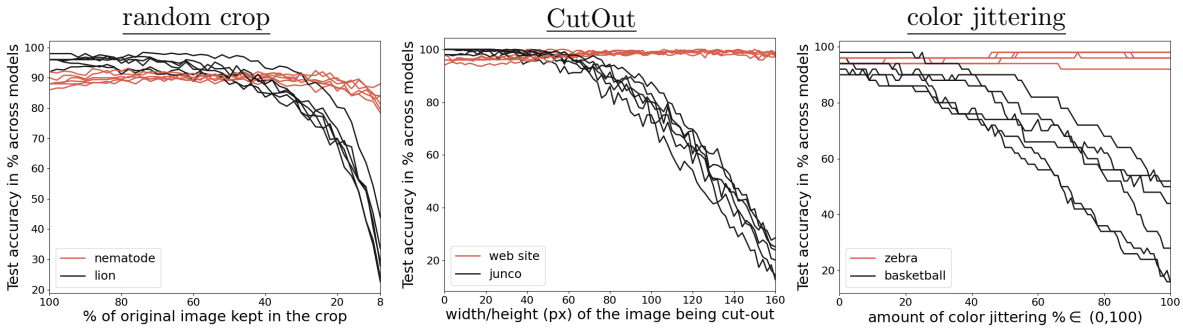

Figure 3: *All results in this figure employ official pretrained models from PyTorch.* Reprise of the bottom left of fig. 2 for three different DAs (**each column**) and using the same 6 popular architectures (resnet50, resnet101, resnet152, densenet121, densenet201, resnext101-32x8d) (**different lines**). 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**.

2.2 The Same Data-Augmentation can be Label-Preserving or Not Between Different Classes

In the previous section 2.1 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 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. 2). 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. 2) 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. 3 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. We propose in the next section to precisely quantify the impact of DA on each dataset class.

2.3 Measuring the Average Treatment Effect of Data-Augmentation on Models' Class-Dependent Bias

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

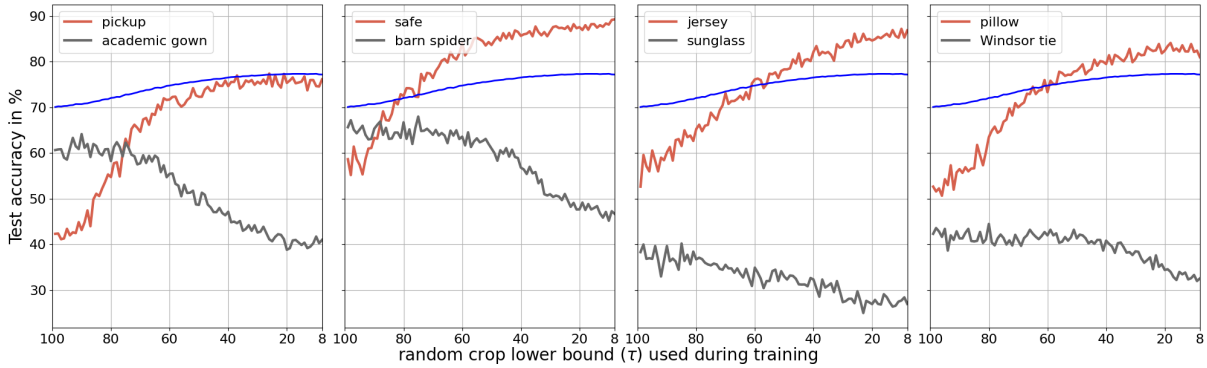

Figure 4: *All results in this figure are averaged over 20 runs and employ the official resnet50 implementation from PyTorch that we trained on Imagenet with horizontal flip and varying lower bound for the random crop DA.* During training, the lower bound on the amount of the original image kept in the 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. 8, in the appendix.

of models with varying DA policies to precisely assess the relation between DA and class-dependent model bias.

First, we propose in fig. 4 a sensitivity analysis by training the same 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 2.4.

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 $\text{Accu}_y(\mathbb{X})$ and our null hypothesis by $H_0 = \mathbb{E}[\text{Accu}_y(\mathbb{X})] < \mathbb{E}[\text{Accu}_y(\mathbb{X}_{\text{DA}})]$ with $\mathbb{X}$ the original dataset and $\mathbb{X}_{\text{DA}}$ 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

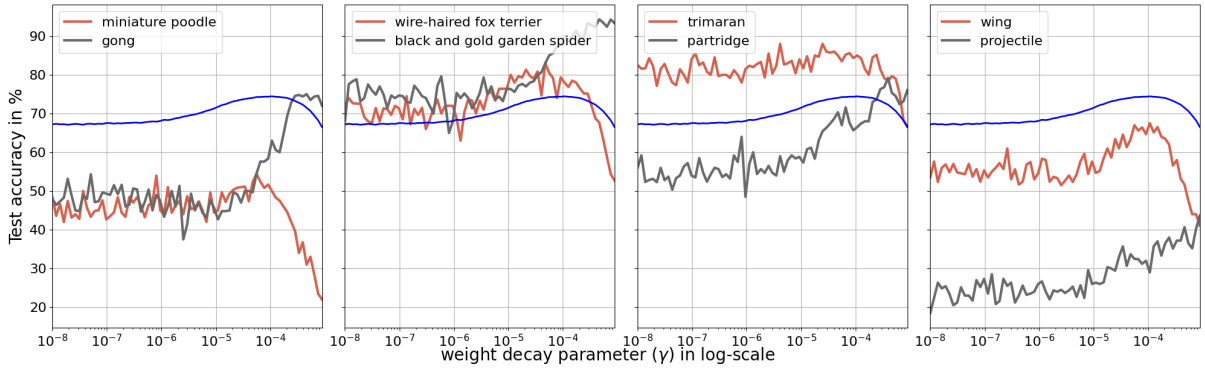

Figure 5: *All results in this figure are averaged over 10 runs and employ the official resnet50 implementation from PyTorch that we trained on Imagenet with varying weight decay parameter.* A very surprising result that we report here is that **the class-dependent bias that we observed from DA also 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. 9, in the appendix. We provide the same figure for DenseNet121 model in fig. 12.

such fine-grained analysis for future work. The next section 2.4 proposes to reproduce those experiments but considering weight decay as a regularizer instead of DA.

2.4 Uninformed Weight Decay Also Creates Class-Dependent Model Bias

We quantified in section 2.3 how much per-class bias was produced by DA. As per the arguments given in sections 2.1 and 2.2, 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 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. 5 the per-class performance of a resnet50 trained on Imagenet with varying weight decay coefficient γ (as was done for DA in fig. 4) 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**. The next section 2.5 proposes to quantifying how much bias transfers to different downstream tasks.

2.5 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

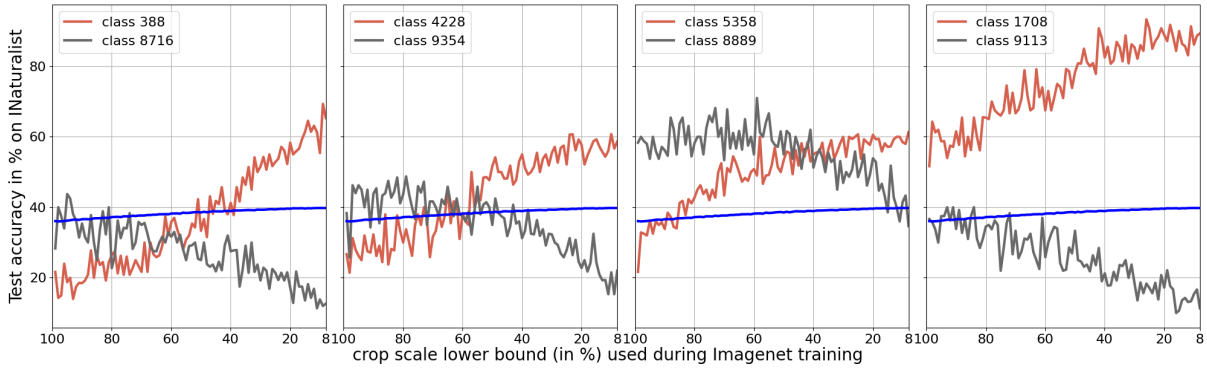

Figure 6: *All results in this figure are averaged over 10 runs and employ the official resnet50 implementation from PyTorch that we pre-trained on Imagenet with varying random crop lower bound and then transferred to INaturalist with frozen backbone weights (only the linear classifier is tuned).* We 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. 13, in the appendix.

(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 2.3, 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. 6 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. 6 only when given a single pre-trained model.

3 Conclusion

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.

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Balestriero, R., Misra, I., and LeCun, Y. (2022). A data-augmentation is worth a thousand samples: Exact quantification from analytical augmented sample moments. *arXiv preprint arXiv:2202.08325*.

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

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

Box, G. E. and Tiao, G. C. (2011). *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. (2000). Vicinal risk minimization. *Advances in neural information processing systems*, 13.

Chen, S., Dobriban, E., and Lee, J. (2020a). A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33:21321–21333.

Chen, X., Fan, H., Girshick, R., and He, K. (2020b). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

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

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

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

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

Gruber, M. H. (2017). *Improving efficiency by shrinkage: The James-Stein and ridge regression estimators*. Routledge.

Guo, X., Zhu, E., Liu, X., and Yin, J. (2018). Deep embedded clustering with data augmentation. In *Asian conference on machine learning*, pages 550–565. PMLR.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hernández-García, A. and König, P. (2018). Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*.

Hu, M. and Li, J. (2019). Exploring bias in gan-based data augmentation for small samples. *arXiv preprint arXiv:1905.08495*.

Huang, G., Liu, S., Van der Maaten, L., and Weinberger, K. Q. (2018). Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2752–2761.

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

Ilse, M., Tomczak, J. M., and Forré, P. (2021). Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR.

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

Jaipuria, N., Zhang, X., Bhasin, R., Arafa, M., Chakravarty, P., Shrivastava, S., Manglani, S., and Murali, V. N. (2020). Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 772–773.

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

Kloft, M., Brefeld, U., Laskov, P., Müller, K.-R., Zien, A., and Sonnenburg, S. (2009). Efficient and accurate lp-norm multiple kernel learning. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

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

Leclerc, G., Ilyas, A., Engstrom, L., Park, S. M., Salman, H., and Madry, A. (2022). ffcv. `https://github.com/libffcv/ffcv/`.

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

LeJeune, D., Balestriero, R., Javadi, H., and Baraniuk, R. G. (2019). Implicit rugosity regularization via data augmentation. *arXiv preprint arXiv:1905.11639*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*.

McLaughlin, N., Del Rincon, J. M., and Miller, P. (2015). 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.

Min, Y., Chen, L., and Karbasi, A. (2021). 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.

Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

Neyman, J. and Pearson, E. S. (1933). 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.

Neyshabur, B., Tomioka, R., and Srebro, N. (2014). In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*.

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

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

Phaisangittisagul, E. (2016). 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.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J., and Liang, P. (2020). Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*.

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

Simard, P., Victorri, B., LeCun, Y., and Denker, J. (1991). Tangent prop-a formalism for specifying selected invariances in an adaptive network. *Advances in neural information processing systems*, 4.

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

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

Tihonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math.*, 4:1035–1038.

Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

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

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. (2018). The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778.

Vapnik, V. and Chervonenkis, A. Y. (1974). The method of ordered risk minimization, i. *Avtomatika i Telemekhanika*, 8:21–30.

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

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

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

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

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. (2020). Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Xu, Y., Noy, A., Lin, M., Qian, Q., Li, H., and Jin, R. (2020). Wemix: How to better utilize data augmentation. *arXiv preprint arXiv:2010.01267*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR.

Supplementary Materials

A Theoretical Relations Between Data-Augmentation, Generalization, and a Model's Bias and Robustness

A.1 Data-Augmentation, Regularization and Structural Risk Minimization

Bias-Variance Decomposition. The expected error of a trained model can *always* be decomposed into two terms that can be tune by altering the considered model class and regularization, and a third term which is the inherent measurement noise, that can not be reduced. In fact, the true labels are obtained from $\boldsymbol{y} = f(\boldsymbol{x}) + \epsilon$ with f the true, unknown, data model, and ϵ some irreducible error i.e. coming from measurements or data compression. For the Mean-Squared Error (MSE) we obtain this decomposition as

$$\begin{aligned}
\mathbb{E}_{\epsilon,\boldsymbol{X},\mathbb{D}}\left[\left(\boldsymbol{y}-\hat{f}(\boldsymbol{x};\boldsymbol{D})\right)^2\right] =& \mathbb{E}_{\epsilon,\boldsymbol{X},\mathbb{D}}\left[\epsilon^2+\epsilon\left(f(\boldsymbol{x})-\hat{f}(\boldsymbol{x};\boldsymbol{D})\right)+\left(f(\boldsymbol{x})-\hat{f}(\boldsymbol{x};\boldsymbol{D})\right)^2\right] \\
=& \mathrm{Var}(\epsilon)^2+\mathbb{E}_{\boldsymbol{X}}\left[\left(f(\boldsymbol{x})-\mathbb{E}_{\mathbb{D}}\left[\hat{f}(\boldsymbol{x};\boldsymbol{D})\right]\right)^2-\mathbb{E}_{\mathbb{D}}\left[\hat{f}(\boldsymbol{x};\boldsymbol{D})\right]^2+\mathbb{E}_{\mathbb{D}}\left[\hat{f}(\boldsymbol{x};\boldsymbol{D})^2\right]\right] \\
=& \mathrm{Var}(\epsilon)^2+\mathbb{E}_{\boldsymbol{X}}\Bigg[\underbrace{\left(f(\boldsymbol{x})-\mathbb{E}_{\mathbb{D}}\left[\hat{f}(\boldsymbol{x};\boldsymbol{D})\right]\right)^2}_{\text{bias}}+\underbrace{\mathrm{Var}_{\mathbb{D}}\left(\hat{f}(\boldsymbol{x};\boldsymbol{D})\right)}_{\text{variance}}\Bigg].
\end{aligned}$$

Although we only derive here the MSE case, the same can be obtained in the multivariate case and in classification settings (Valentini and Dietterich, 2004; Hastie et al., 2009). When the model is with low complexity e.g. the model is under-parametrized, or the regularization is applied aggressively, the variance term becomes small and the bias term increases. Conversely, when the model is not regularized and is overparametrized, the variance term increases and the bias term reduces. One strategy to find the best model is offered by structural risk minimization.

Structural Risk Minimization (SRM). As introduced by Vapnik and Chervonenkis (1974), SRM proposes a search strategy to obtain the best model. One first chooses a class of function that $\hat{f}$ must live in, hopefully informed from a priori knowledge e.g. polynomials of degree k, resnet50 DN architecture. Then, one finds a hierarchical construction of nested functional spaces that relate to the function complexity. For example, it could be considering polynomials of increasing degree, up to k, or considering resnet50s with weights being bounded by an increasing constant. Finally, one minimizes the empirical risk (training error) for each model and picks the one with best valid set performances. The valid set performance can be estimated empirically from a subset of the training set that has been set apart before fitting, or from other measures such as the VC-dimension of the trained model (Vapnik and Chervonenkis, 2015). For example, cross-validating the weight decay parameter of a model corresponds to SRM. We make this connection precise below.

From Data-Augmentation and Weight Decay to Nested Functional Spaces. Regularization is commonly employed during training to prevent overfitting i.e. reduce the model complexity. One might argue that regularization does not *strictly* restrict the model functional space, it simply *favors* simpler models to be used. Yet, it turns out that regularization can be cast as explicitly restricting the parameter space of the model i.e. restricting the functional space of the model.

To see this, let's consider the following loss function to be minimized $\ell(\theta, \gamma)$ that depends of the parameters θ and the dataset $\boldsymbol{D}$. We define the following two optimization problems, one with Tikhonov regularization with amplitude γ and one with a constraint on the space of θ

$$\min_{\theta \in \mathbb{R}^K} \ell(\theta, \boldsymbol{D}) + \gamma \|\theta\|_2^2, \tag{P1}$$

$$\min_{\theta \in \mathbb{R}^K : \|\theta\|_2^2 \leq \beta} \ell(\theta, \boldsymbol{D}). \tag{P2}$$

Let's also denote by $\hat{\theta}(\gamma)$ the parameter value that is a global minimum of (P1); if multiple global minimum exist, pick the one with greatest ℓ_2 norm. We now have to solve for θ, λ the following system that results from the Lagrangian, with λ the Lagrange multiplier (Bertsekas, 2014) and s^2 the slack variable of the inequality

$$\nabla_\theta \ell(\theta, \boldsymbol{D}) + 2\lambda\theta = 0 \text{ and } \|\theta\|_2^2 - \beta - s^2 = 0 \text{ and } 2\lambda s = 0,$$

setting $\lambda \leftarrow \gamma$, $\theta \leftarrow \hat{\theta}(\gamma)$, and $\beta \leftarrow \|\hat{\theta}(\gamma)\|_2^2, s \leftarrow 0$ solves the system. As a result, solving the constrained optimization problem with $\beta = \|\hat{\theta}(\gamma)\|_2^2$ and solving the Tikhonov regularized problem with γ coefficient is equivalent. Since we also have that $\|\hat{\theta}(\gamma_1)\|_2^2 \leq \|\hat{\theta}(\gamma_2)\|_2^2$, we obtain that starting from a high value of Tikhonov regularization and gradually reducing it produces nested functional spaces where the model live in. More general results can be obtained in different regularization settings in Kloft et al. (2009); Hastie et al. (2009). Moving to the case of DA, the same procedure applies. In fact, one can use the results of Phaisangittisagul (2016); LeJeune et al. (2019); Balestriero et al. (2022) to cast DAs such as dropout, and image perturbations as explicit regularizers à la Tikhonov and repeat the above procedure. In fact, DA can be seen as producing a model with lower variance and a bias towards the augmented dataset. Hence, **if the augmented dataset is not aligned with the true underlying data distribution, DA will increase the model's bias**. For results studying DA in the context of structural risk minimization, we refer the reader to Chapelle et al. (2000); Arjovsky et al. (2019); Chen et al. (2020a); Ilse et al. (2021). In short, it is clear from the above that DA is yet another form of regularization that can be applied for structural risk minimization. Yet, even if DA effectively restrict the functional space of the model, it can be at the cost of producing strong biases, as we empirically observed in section 2.

Provable Bias Induced by Data-Augmentation. Especially relevant to our results is a recent result of Xu et al. (2020). In this work, it was theorized what 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 policy exacerbates the already present biases and makes the trained model further away from the unbiased optimum. As per our experiments from section 2, we observe that bias can take many form. For example, having classes with different statistics e.g. objects that appear at very specific scale in the image. One strategy proposed by Xu et al. (2020), assuming that the bias of a model can be measured, was to adapt the DA policy to the dataset at hand to actively correct the present bias, as done for example in McLaughlin et al. (2015); Iosifidis and Ntoutsi (2018); Jaipuria et al. (2020). In fact, when the bias is measurable, many recent studies have shown that DAs could be designed to reduce the bias present in a dataset (McLaughlin et al., 2015; Jaipuria et al., 2020).

A.2 Known Tradeoff Between Regularization, Generalization and Robustness

There exist a few previous work in that direction, especially in the field of robust and overparametrized machine learning.

In Raghunathan et al. (2020) a surprising result demonstrated that the minimum norm interpolant of the original + DA could have a larger standard error than that of the original data's minimum norm

interpolant. Even more surprising, this result was obtained using consistent DA i.e. transformations that do not alter the label information of the samples as in $p_{\boldsymbol{y}|\boldsymbol{x}} = p_{\boldsymbol{y}|\mathcal{T}_\alpha(\boldsymbol{x})}$, or $f^*(\boldsymbol{x}) = f^*(\mathcal{T}_\alpha(\boldsymbol{x}))$ as discussed in eq. (1). This possible degradation of performance occurs as long as the model remains over-parametrized, even when considering the original+DA dataset.

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 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.

Lastly, a few studies have started to study the bias the could be cause from some DAs, although those studies focused on learned DAs e.g. from Generative Adversarial Networks (Hu and Li, 2019). In that scenario, the predicament is that the GAN in itself is biased, and thus any GAN-generated DA will inherit those biases.

B Proof of theorem 1

Proof. To streamline our derivations and without loss of generality, we first impose that $\mathcal{T}_{-\alpha}$ inverts the action of $\mathcal{T}_\alpha$, that $\mathcal{T}_0$ acts as the identity mapping and that $\mathcal{T}_\alpha \circ \mathcal{T}_\beta = \mathcal{T}_{\alpha+\beta}$. In short, we have a *group* structure that allows us to define the following equivalence class $\sim$ that is positive iff two images are related by a transformation, formally defined as

$$\boldsymbol{u} \sim \boldsymbol{v} \iff \exists \alpha : T_\alpha(\boldsymbol{u}) = \boldsymbol{v}, \tag{2}$$

which is *reflexive* ($\boldsymbol{u} \sim \boldsymbol{u}$ with $\alpha = 0$), *symmetric* ($\boldsymbol{u} \sim \boldsymbol{v}$ with α implies that $\boldsymbol{v} \sim \boldsymbol{u}$ with $-\alpha$), and *transitive* ($\boldsymbol{u} \sim \boldsymbol{v}$ with α and $\boldsymbol{v} \sim \boldsymbol{w}$ with β implies that $\boldsymbol{u} \sim \boldsymbol{w}$ with $\alpha + \beta$).

Lastly, given a set of samples $\mathbb{X}$, we define the following a mapping that will decompose $\mathbb{X}$ into a set generators and corresponding DA parameters that allow to recover those subsets when applied to the subset generator. Formally, we have

$$g(\mathbb{X}) \mapsto \{(\mu_1, \mathbb{A}_1), (\mu_2, \mathbb{A}_2), \dots\} \text{ s.t. } \cup_{(\mu,\mathbb{A})\in g(\mathbb{X})} \{T_\alpha(\mu), \forall \alpha \in \mathbb{A}\} = \mathbb{X}. \tag{3}$$

Note that there is a bijection between the pairs given by $g(\mathbb{X})$ and the set of equivalent samples defined by $\mathbb{X}/\sim$. Using this, we obtain the following decomposition of the empirical error

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{X}}\left[\|f^*(\boldsymbol{x}) - f_\theta(\boldsymbol{x})\|\right] = \sum_{(\mu,\mathbb{A})\in g(\mathbb{X})} \sum_{\alpha\in\mathbb{A}} \|f^*(\mathcal{T}_\alpha(\mu)) - f_\theta(\mathcal{T}_\alpha(\mu))\| p(\mathcal{T}_\alpha(\mu)).$$

Given the above, we now simply split a dataset $\mathbb{X}$ into two, one that contains training samples, and one that contains everything else. Without loss of generality we assume that $p(\mathcal{T}_\alpha(\mu))$ is a constant and thus omit it. Using the fact that the model has 0 training error, we obtain that the approximation error between f^* and f_θ can not be 0 unless the DA is perfectly aligned with the level-sets of f^* as in

$$\begin{aligned}
\mathcal{L} &= \sum_{(\mu,\mathbb{A})\in g(\mathbb{X}\setminus\mathbb{X}_{\text{train}})} \sum_{\alpha\in\mathbb{A}} \|f^*(\mathcal{T}_\alpha(\mu)) - f_\theta(\mathcal{T}_\alpha(\mu))\|_2^2 + \sum_{(\mu,\mathbb{A})\in g(\mathbb{X}_{\text{train}})} \sum_{\alpha\in\mathbb{A}} \|f^*(\mathcal{T}_\alpha(\mu)) - f_\theta(\mathcal{T}_\alpha(\mu))\|_2^2 \\
&\geq \sum_{(\mu,\mathbb{A})\in g(\mathbb{X}_{\text{train}})} \sum_{\alpha\in\mathbb{A}} \|f^*(\mathcal{T}_\alpha(\mu)) - f_\theta(\mathcal{T}_\alpha(\mu))\|_2^2 \\
&\geq \sum_{(\mu,\mathbb{A})\in g(\mathbb{X}_{\text{train}})} \sum_{\alpha\in\mathbb{A}} \|f^*(\mathcal{T}_\alpha(\mu)) - \text{const}\|_2^2,
\end{aligned}$$

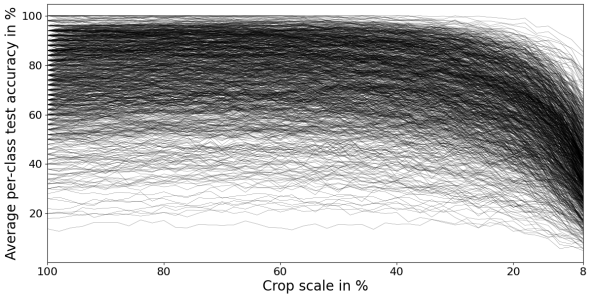

Figure 7: Per-class accuracy evolution when the proportion of the image being cropped (θ) is progressively increased from 8 to 100. While the overall trends mostly applies across classes, some extreme cases can present different sensitivity to aggressive cropping, leading to some classes being more successful as being label-preserved after application of the cropping. The weighted average of those lines (based on the proportion of samples coming from each class) gives back fig. 2).

where the last equality assumes that the model has 0 training error, and will thus predict the same constant for all DA version of the same sample. And since that last equation is always positive if the DA does not respect the true level-sets of the true function f^* we obtain the desired result. The same derivation can be carried out with any desired loss e.g. for classification tasks. □

C Additional Figures

Figure 8: Random samples from the validation set of Imagenet that correspond to fig. 4.

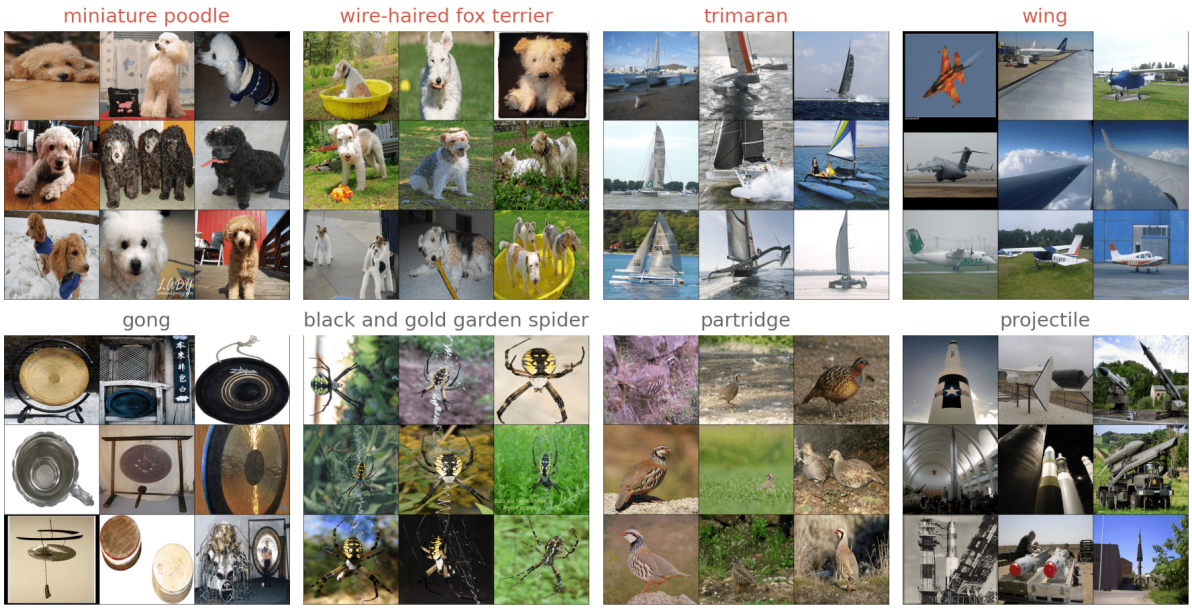

Figure 9: Random samples from the validation set of Imagenet that correspond to fig. 5.

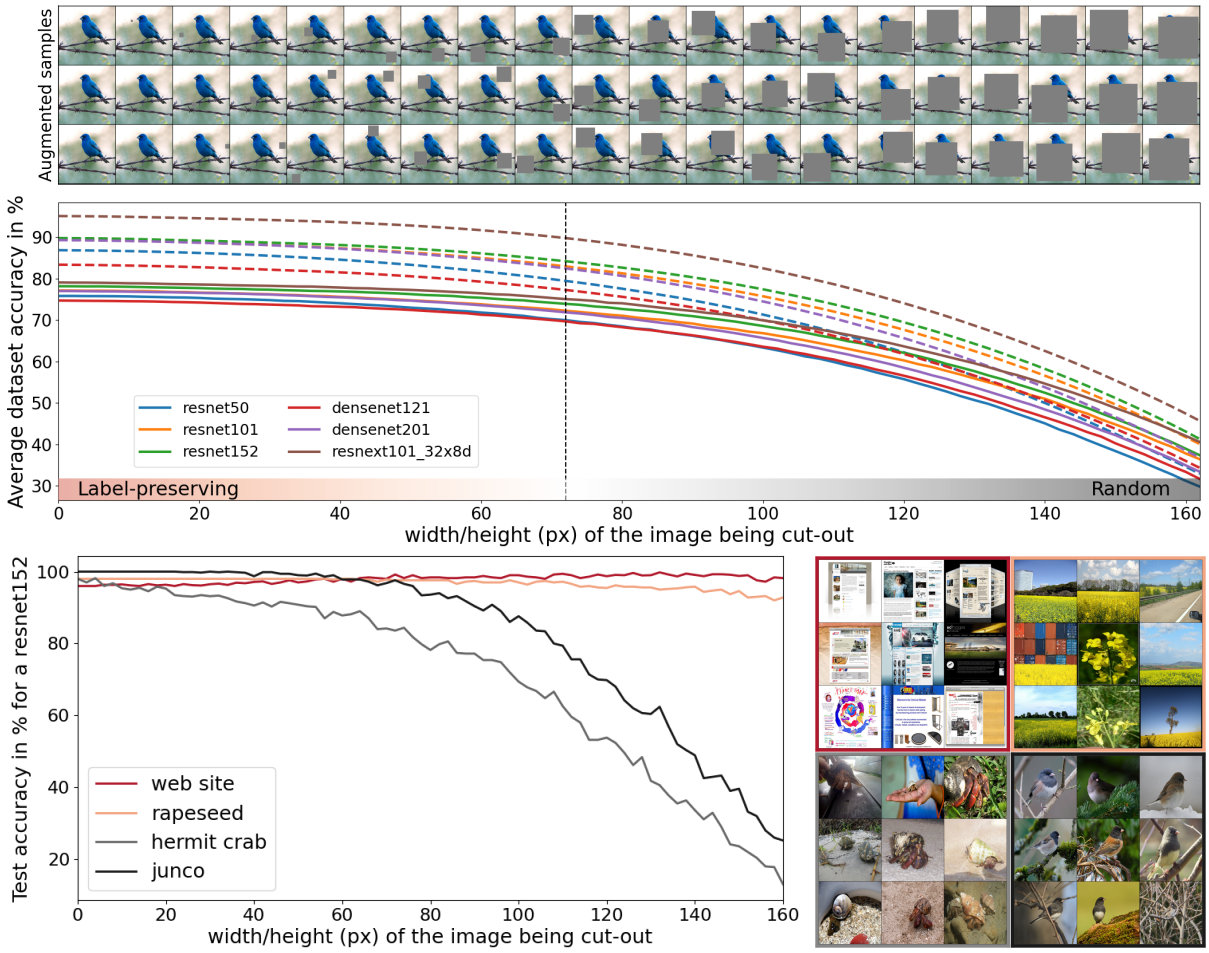

Figure 10: Reprise of fig. 2 but using color jittering DA. The same conclusions are reached, different classes have different label-preserving regimes under this DA.

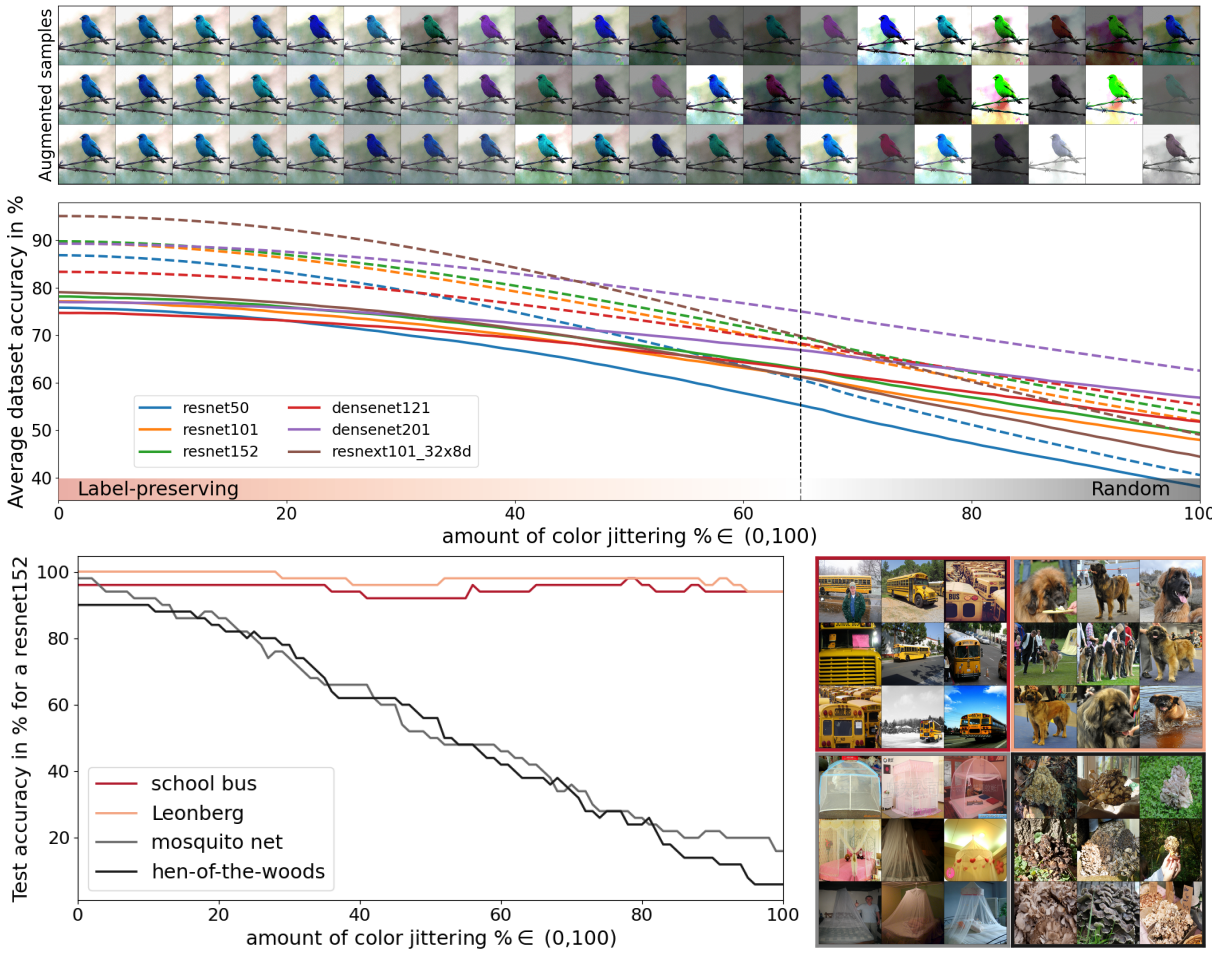

Figure 11: Reprise of fig. 2 but using color jittering DA. The same conclusions are reached, different classes have different label-preserving regimes under this DA.

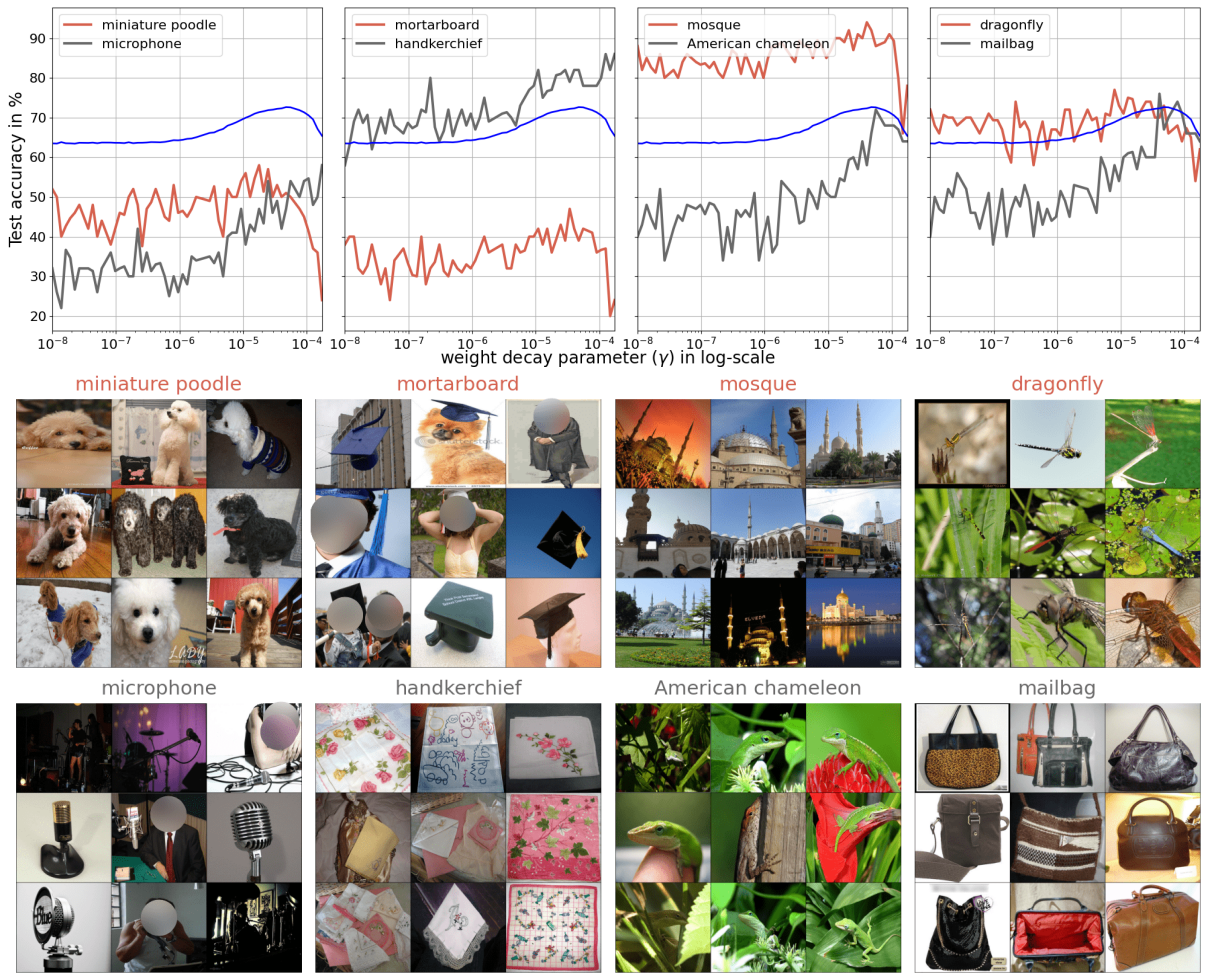

Figure 12: *All results in this figure employ official model implementations from PyTorch that we have trained on Imagenet with varying weight decay parameter.* Reprise of fig. 5 but now with a different architecture (DenseNet121) instead of the ResNet50.

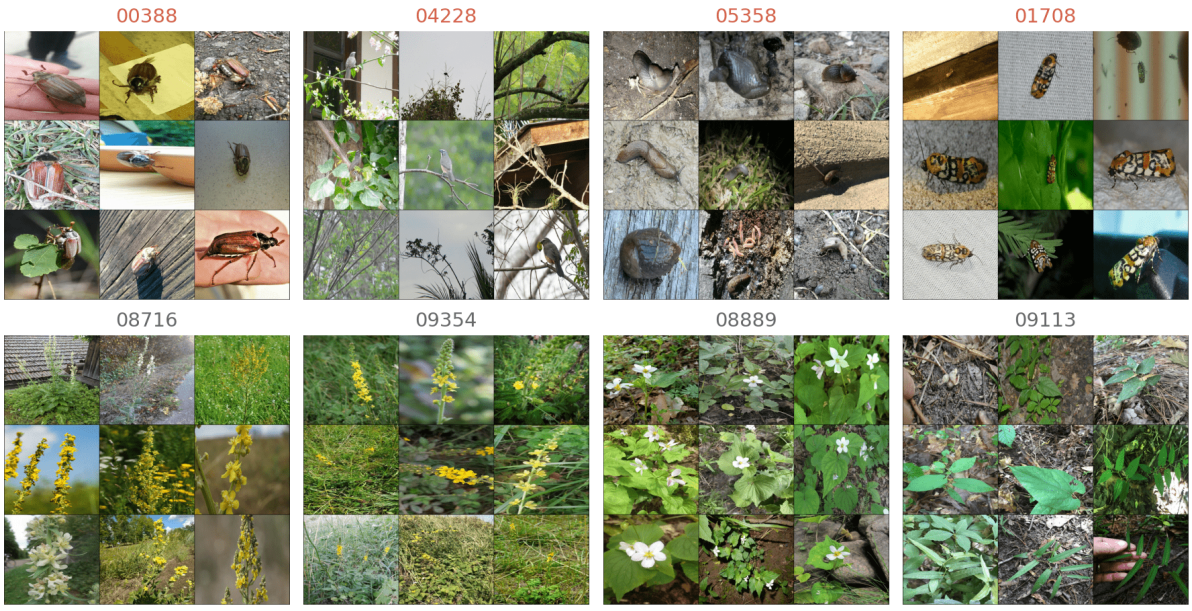

Figure 13: Random samples from the mini training set of INaturalist that correspond to fig. 6