IMAGENET AS A REPRESENTATIVE BASIS FOR DERIVING GENERALLY EFFECTIVE CNN ARCHITECTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate and improve the representativeness of ImageNet as a basis for deriving generally effective convolutional neural network (CNN) architectures that perform well on a diverse set of datasets and application domains. To this end, we conduct an extensive empirical study for which we train 500 CNN architectures, sampled from the broad AnyNetX design space, on ImageNet as well as 8 other image classification datasets. We observe that the performances of the architectures are highly dataset dependent. Some datasets even exhibit a negative error correlation with ImageNet across all architectures. We show how to significantly increase these correlations by *utilizing ImageNet subsets restricted to fewer classes*. We also identify the *cumulative width across layers* as well as the *total depth of the network* as the most sensitive design parameter with respect to changing datasets.

1 INTRODUCTION

Deep convolutional neural networks (CNNs) are the core building block for most modern visual recognition systems and lead to major breakthroughs in many domains of computer perception in the past several years. Therefore, the community has been searching the high dimensional space of possible network architectures for models with desirable properties. Important milestones such as DanNet (Ciresan et al., 2012), AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2015), HighwayNet (Srivastava et al., 2015), and ResNet (He et al., 2016) (a HighwayNet with open gates) can be seen as update steps in this stochastic optimization problem and stand testament that the manual architecture search works. But it is of great importance that the right metrics are used during the search for new neural network architectures. Only when we measure performance with a truly meaningful metric is it certain that a new high-scoring architecture is also fundamentally better. So far, the metric of choice in the community has often been the performance on the most well-known benchmarking dataset—ImageNet (Russakovsky et al., 2014).

More specifically, it would be desirable to construct such a metric from a solid theoretical understanding of deep CNNs. Due to the absence of a solid theoretical basis novel neural network designs are tested in an empirical fashion. Traditionally, model performance has been judged using accuracy point estimates (Krizhevsky et al., 2012; Zeiler & Fergus, 2014; Simonyan & Zisserman, 2015). This simple measure ignores important aspects such as model complexity and speed. Newer work addresses this issue by reporting a curve of the accuracy at different complexity settings of the model, highlighting how well a design deals with the accuracy versus complexity tradeoff (Xie et al., 2017; Zoph et al., 2018).

Very recent work strives to improve the quality of the empiric evaluation even further. There have been attempts to use extensive empirical studies to discover general rules on neural network design (Hestness et al., 2017; Rosenfeld et al., 2020; Kaplan et al., 2020; Tuggener et al., 2020), instead of simply showing the merits of a single neural network architecture. Another line of research aims to improve empiricism by sampling whole populations of models and comparing error distributions instead of individual scalar errors (Radosavovic et al., 2019).

We acknowledge the importance of the above-mentioned improvements in the empirical methods used to test neural networks, but identify a weak spot that runs trough the above-mentioned work: the heavy reliance on ImageNet (Russakovsky et al., 2014) (and to some extent the very similar Cifar100 (Krizhevsky et al., 2009)). In 2011, Torralba and Efros already pointed out that visual recognition datasets that were built to represent the visual world tend to become a small world in themselves

(Torralba & Efros, 2011). Objects are no longer in the dataset because they are important, they are important because they are in the dataset. *In this paper, we investigate how well ImageNet represents a diverse set of visual classification datasets—and present methods to improve said representation, such that CNN architectures optimized on ImageNet become more effective on visual classification beyond ImageNet.* Specifically, our contributions are: (a) an extensive empirical study showcasing the fitness of ImageNet as a basis for generally effective CNN architectures; (b) we show how class-wise subsampled versions of ImageNet in conjunction with the original datasets yield a 2.5-fold improvement in average error correlations with other datasets (c) we identify cumulative block depth and width as the architecture parameters most sensitive to changing datasets.

As a tool for this investigation we introduce the notion of architecture and performance relationship (APR). The performance of a CNN architecture does not exist in a vacuum, it is only defined in relation to the dataset on which it is used. This dependency is what we call APR induced by a dataset. We study the change in APRs between datasets by sampling 500 neural network architectures and training all of them on a set of datasets[1]. We then compare errors of the same architectures across datasets, revealing the changes in APR (see Figure 1). This approach allows us to study the APRs induced by different datasets on a whole population of diverse network designs rather than just a family of similar architectures such as the ResNets (He et al., 2016) or MobileNets (Howard et al., 2017).

All of our code, data and trained models will be made publicly available to ensure reproducibility and facilitate future research.

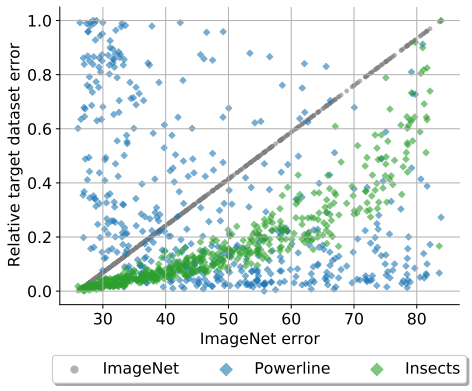

Figure 1: Is a CNN *architecture* that performs well on ImageNet automatically a good choice for a different vision dataset? This plot suggests otherwise: It displays the relative test errors of 500 randomly sampled CNN architectures on three datasets (ImageNet, Powerline, and Insects) plotted against the test error of the same architectures on ImageNet. The architectures have been trained from scratch on all three datasets. Architectures with low errors on ImageNet also perform well on Insects, on Powerline the opposite is the case.

2 Related Work

Neural network design. With the introduction of the first deep CNNs (Ciresan et al., 2012; Krizhevsky et al., 2012) the design of neural networks immediately became an active research area. In the following years many improved architectures where introduced, such as VGG (Simonyan & Zisserman, 2015), Inception (Szegedy et al., 2015), HighwayNet (Srivastava et al., 2015), ResNet (He et al., 2016) (a HighwayNet with open gates), ResNeXt (Xie et al., 2017), or MobileNet (Howard et al., 2017). These architectures are the result of manual search aimed at finding new design principles that improve performance, for example increased network depth and skip connections. More recently, reinforcement learning (Zoph et al., 2018), evolutionary algorithms (Real et al., 2019) or gradient descent (Liu et al., 2019) have been successfully used to find suitable network architectures automatically. Our work relates to manual and automatic architecture design because it adds perspective on how stable results based on one or a few datasets are.

Empirical studies. In the absence of a solid theoretical understanding, large-scale empirical studies are the best tool at our disposal to gain insight into the nature of deep neural networks. These studies can aid network design (Greff et al., 2017; Collins et al., 2017; Novak et al., 2018) or be employed to show the merits of different approaches, for example that the classic LSTM (Hochreiter & Schmidhuber, 1997) architecture can outperform more modern models (Melis et al., 2018), when it is properly regularised. More recently, empirical studies have been used to infer more general rules on the behaviour of neural networks such as a power-law describing the relationship between

[1]Since we only sample models in the complexity regime of 340 mega flops (MF) to 400MF (ResNet-152 has 11.5GF) we could complete the necessary 7500 model trainings within a moderate 85 GPU days on Tesla V100-SXM2-32GB GPUs.

Table 1: Meta data of the used datasets.

DATASET	NO. IMAGES	NO. CLASSES	IMG. SIZE	DOMAIN
CONCRETE	40K	2	227×227	MAINTENANCE
MLC2008	43K	9	312×312	BIOLOGY
IMAGENET	1.3M	1000	256×256	EVERYDAY OBJECTS
HAM10000	10K	7	296×296	MEDICAL
POWERLINE	8K	2	128×128	MAINTENANCE
INSECTS	63K	291	296×296	BIOLOGY
NATURAL	25K	6	150×150	NATURAL SCENES
CIFAR10	60K	10	32×32	EVERYDAY OBJECTS
CIFAR100	60K	100	32×32	EVERYDAY OBJECTS

generalization error and dataset size (Hestness et al., 2017) or scaling laws for neural language models (Kaplan et al., 2020).

Generalization in neural networks. Despite their vast size have deep neural networks shown in practice that they can generalize extraordinarily well to unseen data stemming from the same distribution as the training data. Why neural networks generalize so well is still an open and very active research area (Kawaguchi et al., 2017; Dinh et al., 2017; Zhang et al., 2017). This work is not concerned with the generalization of a trained network to new data, but with the generalization of the architecture design progress itself. Does an architecture designed for a certain dataset, e.g. natural photo classification using ImageNet, work just as well for medical imaging? There has been work investigating the generalization to a newly collected test set, but in this case the test set was designed to be of the same distribution as the original training data (Recht et al., 2019).

Neural network transferability It is known that the best architecture for ImageNet is not necessarily the best base architecture for other applications such as semantic segmentation (Long et al., 2015) or object detection (Chen et al., 2019). Researchers who computed a taxonomy of multiple visions tasks identified that the simmilarities between tasks did not depend on the used architecture (Zamir et al., 2019). Research that investigates the relation between model performance on ImageNet and new classification datasets in the context of transfer learning (Razavian et al., 2014; Donahue et al., 2014) suggests that there is a strong correlation which is also heavily dependent on the training regime used (Kornblith et al., 2019). Our work differs form the ones mentioned above in that we are not interested in the transfer of learned features but transfer of the architecture designs and therefore we train our networks from scratch on each dataset. Moreover do we not only test transferability on a few select architectures but on a whole network space.

Neural network design space analysis. Radosavovic et al. (Radosavovic et al., 2019) introduced network design spaces for visual recognition. They define a design space as a set of architectures defined in a parametric form with a fixed base structure and architectural hyperparameters that can be varied, similar to the search space definition in neural architecture search (Zoph et al., 2018; Real et al., 2019; Liu et al., 2019). The error distribution of a given design space can be computed by randomly sampling model instances from it and computing their training error. We use a similar methodology but instead of comparing different design spaces, we compare the results of the same design space on different datasets.

3 DATASETS

To enable cross dataset comparison of APRs we assembled a corpus of datasets. We chose datasets according to the following principles: (a) include datasets from a wide spectrum of application areas, such that generalization is tested on a diverse set of datasets; (b) only use datasets that are publicly available to anyone to ensure easy reproducibility of our work. Table 1 shows some meta-data of the chosen datasets. For more detailed dataset descriptions and example images please see Chapter A and Figure 8 in the appendix.

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETUP

We sample our architectures form the very general AnyNetX (Radosavovic et al., 2020) parametric network space. The networks in AnyNetX consist of a stem, a body, and a head. The body performs the majority of the computation, stem and head are kept fixed across all sampled models. The body consists of four stages, each stage i starts with a 1×1 convolution with stride s_i, the remainder is a sequence of d_i identical blocks. The blocks are standard residual bottleneck blocks with group convolution (Xie et al., 2017), with a total block width w_i, bottleneck ratio b_i and a group width g_i (into how many parallel convolutions the total width is grouped into). Within a stage, all the block parameters are shared. See Figure 2 for a comprehensive schematic.

The AnyNetX design space has a total of 16 degrees of freedom, having 4 stages with 4 parameters each. We obtain our model instances by performing log-uniform sampling of $d_i \leq 16$, $w_i \leq 1024$ and divisible by 8, $b_i \in 1, 2, 4$, and $g_i \in 1, 2, ..., 32$. The stride s_i is fixed with a stride of 1 for the first stage and a stride of 2 for the rest. We repeatedly draw samples until we have obtained a total of 500 architectures in our target complexity regime of 360 mega flops (MF) to 400 MF. We use a very basic training regime that consists of only flipping and cropping of the inputs in conjunction with SGD plus momentum and weight decay. The same 500 models are trained on each dataset until the loss is reasonably saturated. The exact number of epochs has been determined in preliminary experiments and depends on the dataset (see Table 2). For extensive ablation studies ensuring the empirical stability of our experiments with respect to Cifar10 performance, training duration, training variability, top-1 to top-5 error comparisons, overfitting and class distribution see chapters B.1 to B.7 in the appendix.

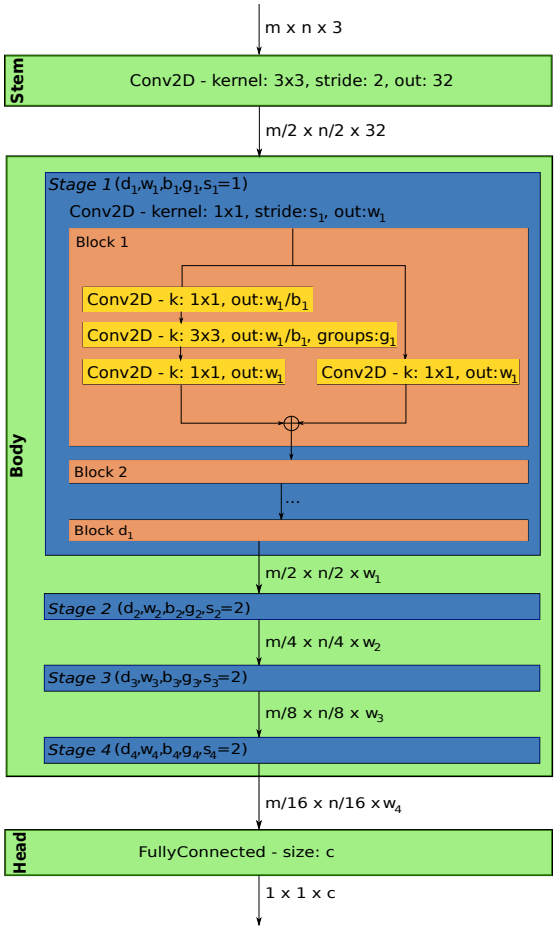

Figure 2: The structure of models in the AnyNetX design space, with a fixed stem and a head, consisting of one fully-connected layer of size c, (where c is the number of classes). Each stage i of the body is parametrised by d_i, w_i, b_i, g_i, the strides of the stages are fixed with $s_1 = 1$ and $s_i = 2$ for the remainder.

4.2 EXPERIMENTAL RESULTS

We analyze the architecture-performance relationship (APRs) in two ways. For every target dataset (datsets which are not ImageNet) we plot the test error of every sampled architecture against the test error of the same architecture (trained and tested) on ImageNet, visualizing the relationship of the target dataset's APR with the APR on ImageNet. Second, we compute Spearman's ρ rank correlation coefficient (Freedman et al., 2007). It is a nonparametric measure for the strength of the relation between two variables (here the error on the target datasets with the error of the same architecture on ImageNet). Spearman's ρ is defined on $[-1, 1]$, where 0 indicates no relationship and -1 or 1 indicates that the relationship between the two variables can be fully described using only a monotonic function.

Table 2: Dataset-specific experimental settings.

DATASET	NO. TRAINING EPOCHS	EVAL. ERROR
CONCRETE	20	TOP-1
MLC2008	20	TOP-1
IMAGENET	10	TOP-5
HAM10000	30	TOP-1
POWERLINE	20	TOP-1
INSECTS	20	TOP-5
NATURAL	20	TOP-1
CIFAR10	30	TOP-1
CIFAR100	30	TOP-5

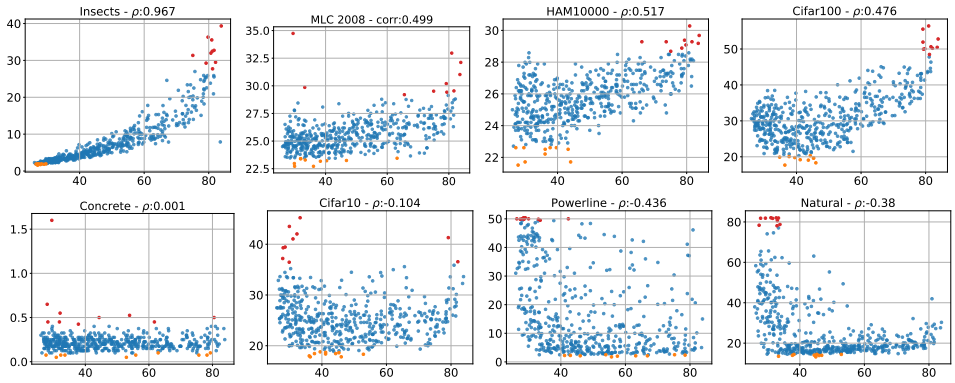

Figure 3: Test errors of all 500 sampled architectures on target datasets (y-axis) plotted against the test errors of the same architectures (trained and tested) on ImageNet (x-axis). The top 10 performances on the target datasets are plotted in orange and the worst 10 performances in red.

Figure 3 contains the described scatterplots with the corresponding correlation coefficients in the title. The datasets plotted in the top row show a strong (Insects) or medium (MLC2008, HAM10000, Cifar100) error correlation with ImageNet. This confirms that many classification tasks have an APR similar to the one induced by ImageNet, which makes ImageNet performance a decent architecture selection indicator for these datasets. The errors on Concrete are independent of the their corresponding ImageNet counterparts since the accuracies are almost saturated with errors between 0 and 0.5. This has implications for practical settings, where in such cases suitable architectures should be chosen according to computational and model complexity considerations rather than ImageNet performance, and reinforces the idea that practical problems may lie well outside of the ImageNet visual world (Stadelmann et al., 2018). The most important insight from Figure 3, however, is that some datasets have a slight (Cifar10) or even strong (Powerline, Natural) *negative error correlation* with ImageNet. Architectures which perform well on ImageNet tend perform sub-par on these datasets. A visual inspection shows that some of the very best architectures on ImageNet perform extraordinarily poor on these three datasets. We can conclude that the *APRs can vary wildly between datasets and high performing architectures on ImageNet do not necessarily work well on other datasets*.

An analysis of the correlations between all datasets (see Figure 21 in the appendix) reveals that Powerline and Natural not only have low correlation with ImageNet but also with most of the other datasets making these two truly particular datasets. Interestingly is the correlation between Powerline and Naural relatively high, which suggests that there is a common trait that makes these two datasets behave differently. MLC 2008, HAM10000 and Cifar100 have a correlation of 0.69 with each other which indicates that they induce a very similar APR. This APR seems to be fairly universal since MLC 2008, HAM10000 and Cifar100 have a moderate to high correlation with all other datasets.

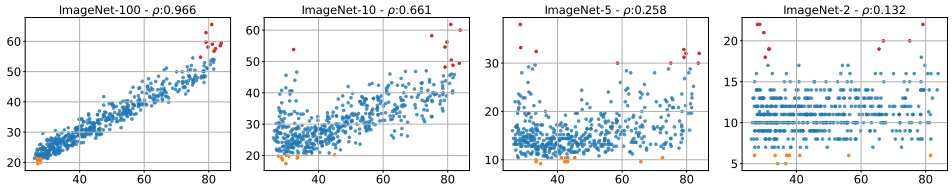

Figure 4: Error of all 500 sampled architectures on subsampled (by number of classes) versions of ImageNet (y-axis) plotted against the error of the same architectures on regular ImageNet (x-axis). The top 10 performances on the target dataset are plotted in orange and the worst 10 performances in red.

4.3 Impact of the Number of Classes

Having established that APR varies heavily between datasets, leaves us width the questions if it is possible to identify properties of the datasets themselves that influences its APR and if it is possible to control these factors to reduce the APR differences.

ImageNet has by far the largest number of classes among all the datasets. Insects, which is the dataset with the second highest class count, also shows the strongest similarity in APR to ImageNet. This suggests that the number of classes might be an important property of a dataset with respect to APR. We test this hypothesis by running an additional set of experiments on subsampled versions of ImageNet. We create new datasets by randomly choosing a varying number of classes from ImageNet and deleting the rest of the dataset. This allows us to isolate the impact of the number of classes while keeping all other aspects of the data itself identical. We create four subsampled ImageNet versions with 100, 10, 5, and 2 classes, which we call ImageNet-100, ImageNet-10, ImageNet-5, and ImageNet-2, respectively. We refer to the resulting group of datasets (including the original ImageNet) as the ImageNet-X family. The training regime for ImageNet-100 is kept identical to the one of ImageNet, for the other three datasets we switch to top-1 error and train for 40 epochs, to account for the smaller dataset size. (see section B.6 in the appendix for a control experiment that disentangles the effects of reduced dataset size and reduced number of classes)

Figure 4 shows the errors on the subsampled versions plotted against the errors on original ImageNet. APR on ImageNet-100 shows an extremely strong correlation with APR on ImageNet. This correlation significantly weakens as the class count gets smaller. ImageNet-2 is on the opposite end has errors which are practically independent from the ones on ImageNet. *This confirms our hypothesis that the number of classes is a dataset property with significant effect on the architecture to performance relationship.*

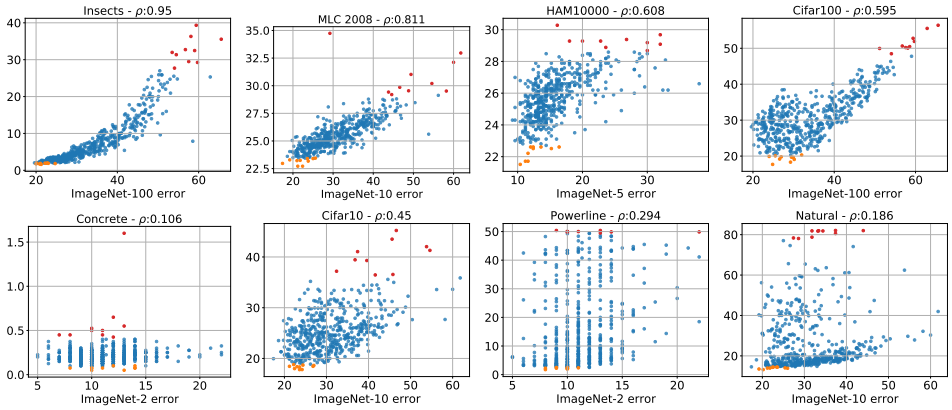

Figure 5: Test errors of all 500 sampled architectures on target datasets (y-axis) plotted against the test errors of the same architectures on the ImageNet-X (x-axis). The top 10 performances on the target dataset are orange, the worst 10 performances red.

Table 3: Comparison of error correlations between target datasets and ImageNet as well as the closest ImageNet-X member.

DATASET	ρ -IMAGENET	ρ -IMAGENET-X	DIFFERENCE
CONCRETE	0.001	0.106	0.105
MLC2008	0.476	0.811	0.335
HAM10000	0.517	0.608	0.091
POWERLINE	-0.436	0.294	0.73
INSECTS	0.967	0.95	-0.017
NATURAL	-0.38	0.186	0.566
CIFAR10	-0.104	0.45	0.554
CIFAR100	0.476	0.595	0.119
AVERAGE	0.19	0.507	0.317

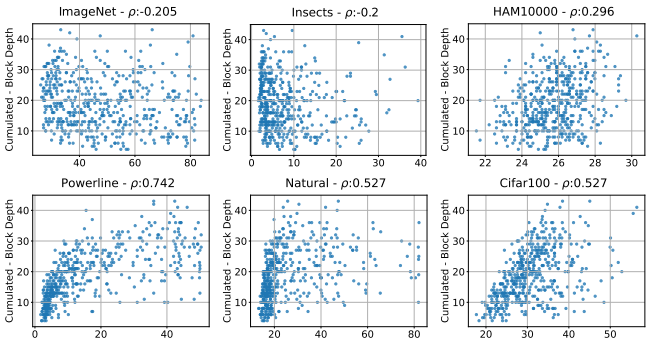

Figure 6: Errors of all 500 sampled architectures on ImageNet, Insects, HAM10000, Powerline, Natural, and Cifar100 (x-axis) plotted against the cumulative block *depths* (y-axis).

We have observed that the number of classes has a profound effect on the APR associated with ImageNet-X members. It is unlikely that simply varying the number of classes in this dataset is able to replicate the diversity of APRs present in an array of different datasets. However, it is reasonable to assume that a dataset's APR is better represented by the ImageNet-X member closest in terms of class count, instead of ImageNet. We thus recreate Figure 3 with the twist of not plotting the target dataset errors against ImageNet, but against the ImageNet-X variant closest in class count (see Figure 5). We observe gain in correlation across all datasets, in the cases of MLC2008 or Cifar10 a quite extreme one. The datasets which have a strong negative correlation with ImageNet (Powerline, Natural) have slightly (Natural) or even moderately (Powerline) positive correlation to their ImageNet-X counterparts. A visual inspection shows that the best models on Imagenet-X also yield excellent results on Powerline and Natural, which was not the case for ImageNet. Table 3 shows the error correlations of all target datasets with ImageNet as well as with their ImageNet-X counterpart. *The move from ImageNet to ImageNet-X more than doubles the average correlation (from* 0.19 *to* 0.507*), indicating that the ImageNet-X family of datasets is capable to represent a much wider variety of APRs than ImageNet alone.*

4.4 IDENTIFYING DRIVERS OF DIFFERENCE BETWEEN DATASETS

The block width and depth parameters of the top 15 architectures for ImageNet (see Figure 20 in the appendix) follow a clear structure: they consistently start with low values for both block depth and width in the first stage, then the values steadily increase across the stages for both parameters. The error relationships observed in Figure 3 are consistent with how well these patterns are replicated by the other datasets. Insects shows a very similar pattern, MLC2008 and HAM10000 have the same trends but more noise. Powerline and Natural clearly break from this structure, having a flat or decreasing structure in the block with and showing a quite clear preference for a small block depth in the final stage. Cifar10 and Cifar100 are interesting cases, they have the same behaviour as ImageNet with respect to block width but a very different one when it comes to block depth.

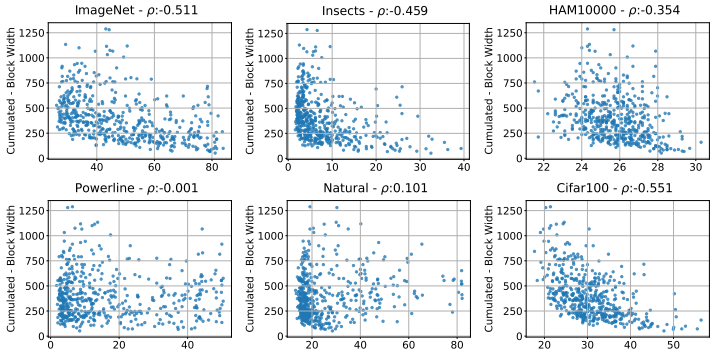

Figure 7: Errors of all 500 sampled architectures on ImageNet, Insects, HAM10000 and Cifar100 (x-axis) plotted against the cumulative block *widths* (y-axis).

Table 4: Correlation of observed error rates with the cumulative block depth and width parameters for all ImageNet-X datasets.

DATASET	C. BLOCK DEPTH	C. BLOCK WIDTH
IMAGENET	-0.205	-0.511
IMAGENET-100	-0.022	-0.558
IMAGENET-10	0.249	-0.457
IMAGENET-5	0.51	-0.338
IMAGENET-2	0.425	-0.179

We thus investigate the effect of the cumulative block depth (summation of the depth parameter for all four stages, yielding the total depth of the architecture) across the whole population of architectures by plotting the cumulative block depth against the test error for the six above-mentioned datasets. Additionally, we compute the corresponding correlation coefficients. Figure 6 shows that the best models for ImageNet have a cumulative depth of at least 10. Otherwise there is no apparent dependency between the ImageNet errors and cumulative block depth. The errors of Insects do not seem to be related to the cumulative block depth at all. HAM10000 has a slight right-leaning spread leading to a moderate correlation, but the visual inspection shows no strong pattern. The errors on Powerline, Natural, and Cifar100 on the other hand have a strong dependency with the cumulative block depth. The error increases with network depth for all three datasets. with the best models all having a cumulative depth smaller than 10.

We also plot the cumulative block widths against the errors and compute the corresponding correlation coefficients for the same six datasets (see Figure 7). We observe that the ImageNet errors are negatively correlated with the cumulative block width, and visual inspection shows that a cumulative block width of at least 250 is required to achieve a decent performance. The errors on Insects and HAM10000 replicate this pattern to a lesser extent, analogous to the top 15 architectures. Powerline and Natural have no significant error dependency with the cumulative block width, but Cifar100 has an extremely strong negative error dependency with the cumulative block width, showing that it is possible for a dataset to replicate the behaviour on ImageNet in one parameter but not the other. In the case of Cifar100 and ImageNet, low similarity in block depth and high similarity in block width yield a medium overall similarity of ARPs on Cifar100 and Imagenet. This is consistent with the overall relationship of the two datasets displayed in Figure 3.

Combining this result with the outcome of the last section, we study the interaction between the number of classes, the cumulated block depth and the cumulative block width. Table 4 contains the correlations between cumulative block depth/width and the errors on all members of ImageNet-X. With decreasing number of classes, the correlation coefficients increase for cumulative block depth and cumulative block width. Although the effect on cumulative block depth is stronger, there is a significant impact on both parameters. We therefore can conclude that *both optimal cumulative block depth and cumulative block with can drastically change based on the dataset choice and that both are simultaneously influenced by the class count.*

5 DISCUSSION AND CONCLUSIONS

ImageNet is not a perfect proxy. We have set out to explore how well other visual classification datasets are represented by ImageNet. Unsurprisingly there are differences between the APRs induced by the datasets. More surprising and worrying, however, is that for some datasets ImageNet not only is an imperfect proxy but a very bad one. The negative error correlations with Natural, Powerline and Cifar10 indicates that architecture search based on ImageNet performance is worse than random search for these datasets.

Varying the number of classes is a cheap and effective remedy. It is striking how much more accurately the ImageNet-X family is able to represent the diversity in APRs present in our dataset collection, compared to just ImageNet by itself. It has become commonplace to test new architectures in multiple complexity regimes (He et al., 2016; Howard et al., 2017), we argue for augmenting this testing regime with an additional dimension for class count. This simple and easy to implement extension would greatly extend the informative value of future studies on neural network architectures.

Block depth and block are the most sensitive parameter. We have identified block depth and block width as key factors of variance between different APRs of an architecture. When designing an architecture for a practical application, where out of the box models perform poorly, these two dimensions should be explored first.

Future directions. A future similar study should shed light on how well the breadth of other domains such as object detection or speech classification are represented by their essential datasets. In doing so it could be verified if the varying the number of classes also helps covering more dataset variability in these domains.
A labeled dataset will always be a biased description of the visual world, due to having a fixed number of classes and being built with some systematic image collection process. Self-supervised learning of visual representations (Jing & Tian, 2019) could serve as remedy for this issue. Self-supervised architectures could be fed with a stream completely unrelated images, collected from an arbitrary number of sources in a randomized way. A comparison of visual features learned in this way could yield a more meaningful measure of the quality of CNN architectures.

Limitations As with any experimental analysis of a highly complex process such as training a CNN it is virtually impossible to consider every scenario. We list below three dimensions along which our experiments are limited together with measures we took to minimize the impact of these limitations.

Data scope: We criticize ImageNet for only representing a fraction of the "visual world". We are aware that our dataset collection does not span the entire "visual world" either but went to great lengths to maximise the scope of our dataset collection by purposefully choosing datasets from different domains, which are visually distinct.

Architecture scope: We sample our architectures from the large AnyNetX network space. It contains the CNN building blocks to span basic designs such as AlexNet or VGG as well as the whole ResNet, ResNeXt and RegNet families. We acknowledge that there are popular CNN components not covered, however, Radosavovic et al. (Radosavovic et al., 2020) present ablation studies showing that network designs sourced from high performing regions in the AnyNetX space also perform highly when swapping in different originally missing components such as depthwise convolutions (Chollet, 2017), swish activation functions (Ramachandran et al., 2018) or the squeeze-and-excitation (Hu et al., 2018) operations.

Training scope: When considering data augmentation and optimizer settings there are almost endless possibilities to tune the training process. We opted for a very basic setup with no bells an whistles in general. For certain such aspects of the training, which we assumed might skew the results of our study (such as training duration, dataset prepossessing etc.), we have conducted extensive ablation studies to ensure that this is not the case (see sections B.2 and B.7 in the appendix).

ACKNOWLEDGMENTS

removed for review

REFERENCES

Puneet Bansal. Intel image classification. 2018. URL https://www.kaggle.com/puneet6060/intel-image-classification.

Oscar Beijbom, Peter J. Edmunds, David I. Kline, B. Greg Mitchell, and David J. Kriegman. Automated annotation of coral reef survey images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1170–1177. IEEE Computer Society, 2012.

Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *32th Annual Conference on Neural Information Processing Systems*, pp. 6638–6648, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/228b25587479f2fc7570428e8bcbabdc-Abstract.html.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.195. URL https://doi.org/10.1109/CVPR.2017.195.

Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649. IEEE Computer Society, 2012. URL https://doi.org/10.1109/CVPR.2012.6248110.

Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. Capacity and trainability in recurrent neural networks. In *5th International Conference on Learning Representations*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BydARw9ex.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In Doina Precup and Yee Whye Teh (eds.), *34th International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017. URL http://proceedings.mlr.press/v70/dinh17b.html.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *31th International Conference on Machine Learning*, pp. 647–655. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/donahue14.html.

David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.*, 28(10):2222–2232, 2017. URL https://doi.org/10.1109/TNNLS.2016.2582924.

Oskar Liset Pryds Hansen, Jens-Christian Svenning, Kent Olsen, Steen Dupont, Beulhah H. Garner, Alexandros Iosifidis, Benjamin W. Price, and Toke T. Høye. Image data used for publication "Species-level image classification with convolutional neural network enable insect identification from habitus images ", November 2019. URL https://doi.org/10.5281/zenodo.3549369.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. IEEE Computer Society, 2016. URL https://doi.org/10.1109/CVPR.2016.90.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017. URL http://arxiv.org/abs/1712.00409.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00745. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *CoRR*, abs/1902.06162, 2019. URL http://arxiv.org/abs/1902.06162.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *CoRR*, abs/1710.05468, 2017. URL http://arxiv.org/abs/1710.05468.

Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671. Computer Vision Foundation / IEEE, 2019. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Kornblith_Do_Better_ImageNet_Models_Transfer_Better_CVPR_2019_paper.html.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (eds.), *26th Annual Conference on Neural Information Processing Systems*, pp. 1106–1114, 2012.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298965. URL https://doi.org/10.1109/CVPR.2015.7298965.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *6th International Conference on Learning Representations,*. OpenReview.net, 2018. URL https://openreview.net/forum?id=ByJHuTgA-.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *6th International Conference on Learning Representations*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HJC2SzZCW.

Ç F Özgenel and A Gönenç Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, pp. 1–8. IAARC Publications, 2018. URL https://data.mendeley.com/datasets/5y9wdsg2zt/2.

Ilija Radosavovic, Justin Johnson, Saining Xie, Wan-Yen Lo, and Piotr Dollár. On network design spaces for visual recognition. In *International Conference on Computer Vision*, pp. 1882–1890. IEEE, 2019. URL https://doi.org/10.1109/ICCV.2019.00197.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10425–10433, 2020. URL https://doi.org/10.1109/CVPR42600.2020.01044.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. In *6th International Conference on Learning Representations Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 512–519. IEEE Computer Society, 2014. URL https://doi.org/10.1109/CVPRW.2014.131.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 4780–4789. AAAI Press, 2019. URL https://doi.org/10.1609/aaai.v33i01.33014780.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *36th International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019. URL http://proceedings.mlr.press/v97/recht19a.html.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *8th International Conference on Learning Representations*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

A. S. M. Shihavuddin, Nuno Gracias, Rafael García, Arthur C. R. Gleason, and Brooke Gintert. Image-based coral reef classification and thematic mapping. *Remote. Sens.*, 5(4):1809–1841, 2013. URL https://data.mendeley.com/datasets/86y667257h/2.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1409.1556.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

Thilo Stadelmann, Mohammadreza Amirian, Ismail Arabaci, Marek Arnold, Gilbert François Duivesteijn, Ismail Elezi, Melanie Geiger, Stefan Lörwald, Benjamin Bruno Meier, Katharina Rombach, et al. Deep learning in the wild. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pp. 17–38. Springer, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. IEEE Computer Society, 2015. URL https://doi.org/10.1109/CVPR.2015.7298594.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1521–1528. IEEE Computer Society, 2011. URL https://doi.org/10.1109/CVPR.2011.5995347.

Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11): 1958–1970, 2008. URL https://doi.org/10.1109/TPAMI.2008.128.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *CoRR*, abs/1803.10417, 2018. URL https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T.

Lukas Tuggener, Mohammadreza Amirian, Fernando Benites, Pius von Däniken, Prakhar Gupta, Frank-Peter Schilling, and Thilo Stadelmann. Design patterns for resource-constrained automated deep-learning methods. *AI*, 1(4):510–538, 2020.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995. IEEE Computer Society, 2017. URL https://doi.org/10.1109/CVPR.2017.634.

Ömer Emre Yetgin, Ömer Nezih Gerek, and Ömer Nezih. Ground truth of powerline dataset (infrared-ir and visible light-vl). *Mendeley Data*, 8, 2017. URL https://data.mendeley.com/datasets/n6wrv4ry6v/8.

Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In Sarit Kraus (ed.), *International Joint Conference on Artificial Intelligence 2019*, pp. 6241–6245. ijcai.org, 2019. doi: 10.24963/ijcai.2019/871. URL https://doi.org/10.24963/ijcai.2019/871.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *13th European Conference on Computer Vision, Proceedings, Part I*, pp. 818–833. Springer, 2014. URL https://doi.org/10.1007/978-3-319-10590-1_53.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710. IEEE Computer Society, 2018. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zoph_Learning_Transferable_Architectures_CVPR_2018_paper.html.

A EXTENDED DATASET DESCRIPTION

Concrete (Özgenel & Sorguç, 2018) contains 40 thousand image snippets produced from 458 high-resolution images that have been captured from various concrete buildings on a single campus. It contains two classes, positive (which contains cracks in the concrete) and negative (with images that show intact concrete). With 20 thousand images in both classes the dataset is perfectly balanced.

MLC2008 (Shihavuddin et al., 2013) contains 43 thousand image snippets taken form the MLC dataset (Beijbom et al., 2012), which is a subset of the images collected at the Moorea Coral Reef Long Term Ecological Research site. It contains images from three reef habitats and has nine classes. The class distribution is very skewed with crustose coralline algae (CCA) being the most common by far (see Figure 16 in the Appendix).

ImageNet (Russakovsky et al., 2014) (ILSVRC 2012) is a large scale dataset containing 1.3 million photographs sourced from flickr and other search engines. It contains 1000 classes and is well balanced with almost all classes having exactly 1300 training and 50 validation samples.

HAM10000 (Tschandl et al., 2018) ("Human Against Machine with 10000 training images") is comprised of dermatoscopic images, collected from different populations and by varied modalities. It is a representative collection of all important categories of pigmented lesions that are categorized into seven classes. It is imbalanced with an extreme dominance of the melanocytic nevi (nv) class (see Figure 16 in the Appendix).

Powerline (Yetgin et al., 2017) contains images taken in different seasons as well as weather conditions from 21 different regions in Turkey. It has two classes, positive (that contain powerlines) and negative (which do not). The dataset contains 8000 images and is balanced with 4000 samples per classes. Both classes contain 2000 visible-light and 2000 infrared images.

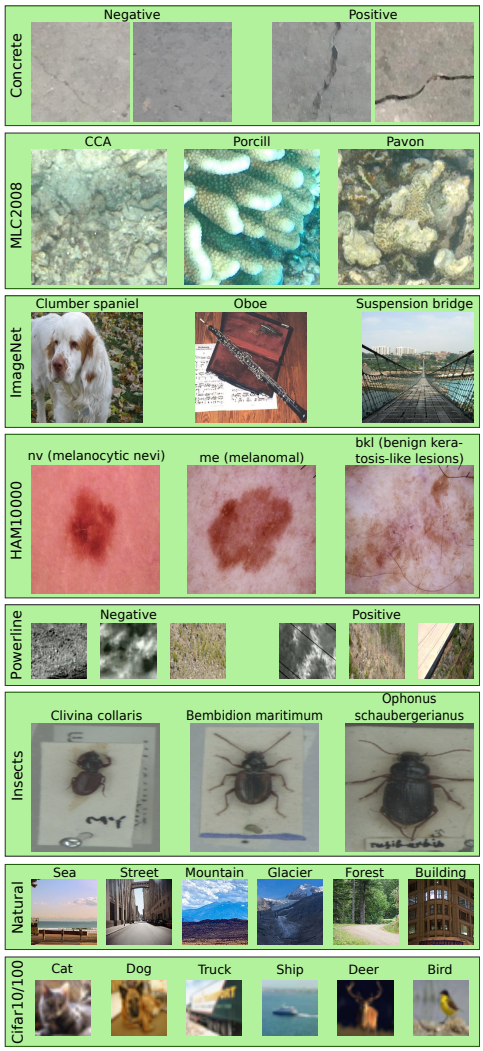

Figure 8: Example images from each dataset. Images of Cifar10/100 are magnified fourfold, the rest are shown in their original resolution (best viewed by zooming into the digital document).

Insects (Hansen et al., 2019) contains 63 thousand images of 291 insect species. The images have been taken of the collection of British carabids from the Natural History Museum London. The dataset is not completely balanced but the majority of classes have 100 to 400 examples.

Intel Image Classification (Bansal, 2018) dataset ("natural") is a natural scene classification dataset containing 25 thousand images and 6 classes. It is very well balanced with all classes having between 2.1 thousand and 2.5 thousand samples.

Cifar10 and Cifar100 (Krizhevsky et al., 2009) both consist of 60 thousand images. The images are sourced form the 80 million tiny images dataset (Torralba et al., 2008) and are therefore of similar nature (photographs of common objects) as the images found in ImageNet, bar the much smaller resolution. Cifar10 has 10 classes with 6000 images per class, Cifar100 consists of 600 images in 100 classes, making both datasets perfectly balanced.

Table 5: Top-1 error of reference network implementations (Radosavovic et al., 2020) for Cifar10.

MODEL	RESNET-56	RESNET-110	ANYNET-56	ANYNET-110
ERROR	5.91	5.23	5.68	5.59

B ADDITIONAL ABLATION STUDIES

This Chapter contains additional studies not suited for the main text. Most of these studies are designed to test for possible flaws or vulnerabilities in our experiments and therefore further strengthen the empirical robustness of our results.

B.1 STABILITY OF EMPIRICAL RESULTS ON CIFAR10

The top-1 errors of our sampled architectures on Cifar10 lie roughly between 18 and 40, which is fairly poor, not only compared to the state of the art but also compared to performance that can be achieved with fairly simple models. This calls into question if our Cifar10 results are flawed in a way that might have lead us to wrong conclusions. We address this by running additional tests on Cifar10 and evaluate their impact on our main results. We get a goalpost for what performance would be considered good with our style of neural network and training setup by running the baseline code for Cifar10 published by Radosavovic et al. (Radosavovic et al., 2020). Table 5 shows that these baseline configurations achieve much lower error rates. We aim to improve the error results on Cifar10 in two ways: First we train our architecture population with standard settings for 200 epochs instead of 30, second we replaced the standard network stem with one that is specifically built for Cifar10, featuring less stride and no pooling. Figure 9 shows scatterplots of the errors from all 500 architectures on Cifar10 against the errors on ImageNet and ImageNet-10. We can see that both new training methods manage to significantly improve the performance with a minimum top-1 error below 10 in both cases. More importantly can we observe that both new training methods have, despite lower overall error, a very similar error relationship to ImageNet. The error correlation is even slightly lower than with our original training (replicated in Figure 9 left row). We can also see that in all three cases the error relationship can be significantly strengthened by replacing ImageNet with ImageNet-10, *this shows that tuning for individual performance on a dataset does not significantly impact the error relationships between datasets which further strengthens our core claim.*

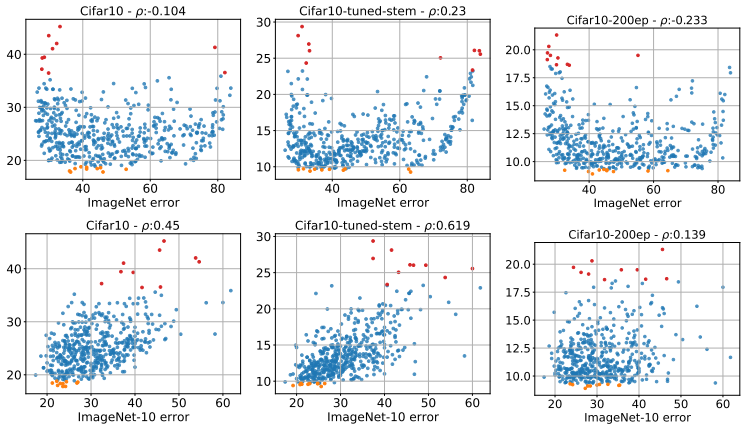

Figure 9: The Cifar10 test errors of all 500 architectures plotted against ImageNet (top row) and ImageNet-10 (bottom row), shown for our original Cifar10 training (left column), training with a Cifar10 specific stem in the architecture (middle column), and training for 200 epochs, which is roughly 6 times longer (right column). The plots show that the error correlation with ImageNet-10 is much larger in all three cases, confirming that optimizing for individual Cifar10 performance does not alter our core result.

B.2 VERIFYING TRAINING DURATION

Since we have a limited amount of computational resources and needed to train a vast number of networks we opted to train the networks up to the number of epochs where they started to saturate significantly in our pre-studies. As we have seen in section B.1 can the network performance still improve quite a bit if it is trained for much longer. Even though the improved performances on Cifar10 did not yield any results contradicting the findings of our study, we still deemed it necessary to closer inspect what happened in the later stages of training and thus performed a sanity check for Cifar10 as well as the other two datasets that show a negative error correlation with ImageNet—Powerline and Natural. Figure 10 shows the Cifar10 test error curves of 20 randomly selected architectures over 200 epochs. On the left side we see the same curves zoomed in to epochs 30 to 200. We see that the error decreases steadily for all architectures, the ranking among architectures barely changes past epoch 30. The relative performance between architectures and not absolute error rates are relevant for our evaluations, we can therefore conclude that the errors at epoch 30 are an accurate enough description of an architecture's power.

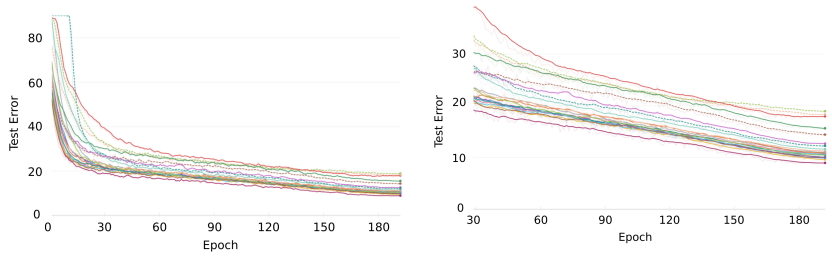

Figure 10: Cifar10 test error curves of 20 randomly sampled architectures trained over 200 epochs (left). The same error curves but cut to epochs 30 to 200.

For Powerline and Natural, we select the five best and five worst architectures respectively and continue training them for a total of five times the regular duration. Figure 11 shows the resulting error curves. Both datasets exhibit minimal changes in the errors of the top models. On Natural we observe clear improvements on the bottom five models but similar to Cifar10 there are very little changes in terms of relative performance. Powerline exhibits one clear cross-over but for the remainder of the bottom five models the ranking also stays intact. *Overall we can conclude that longer training does not have a significant effect on the APR of our datasets.*

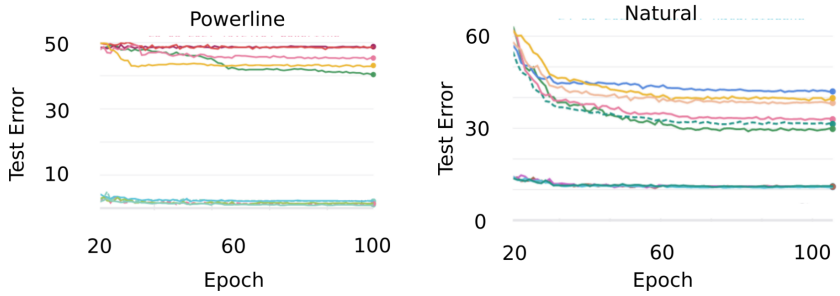

Figure 11: Test error curves of the five best and five worst models on Powerline and Natural, respectively, when training is continued to epoch 100

B.3 IMPACT OF TRAINING VARIABILITY

The random initialization of the model weights has an effect on the performance of a CNN. In an empirical study it would therefore be preferable to train each model multiple times to minimize this variability. We opted to increase the size of our population as high as our computational resources allow, this way we get a large number of measurements to control random effects as well as an error estimate of a large set of architectures. However, we still wanted to determine how much of the total

variability is caused by training noise and how much is due to changing the architectures. We estimate this by selecting two of the sampled CNN designs, number 147 performing slightly above average with an error of $e_{147} = 11.9$ and number 122 performing slightly below average with $e_{122} = 14.5$. The quantiles of the error distribution from all 500 architectures are $q_{0.25} = 11.53$, $q_{0.5} = 13.02$ and $q_{0.75} = 15.46$ with an overall mean of $\mu = 13.9$. We then train the architectures 147 and 122 each 250 times. Figure 12 shows the error distributions of both selected architectures as well as the overall distribution obtained from training each of the 500 architectures once. There is of course some variability within both architectures but both individual architectures produce very narrow densities and show essentially no overlap. *We can therefore conclude that the effect of choosing an architecture is much greater than the variability caused by random training effects.*

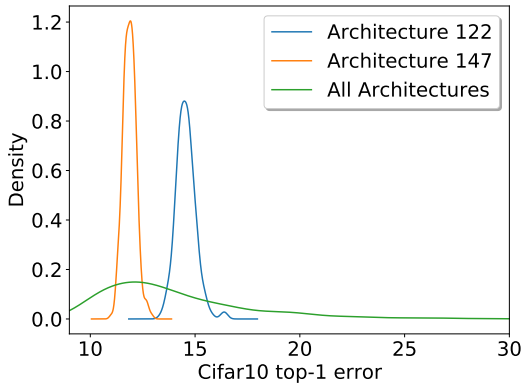

Figure 12: Error distributions on Cifar10 of two architectures (122, 147) both trained from scratch 250 times as well as the Cifar10 error distribution of all 500 architectures. The plot shows that the variability caused by changing architecture is much larger than the one caused by random training effects.

B.4 Relationship of Top-1 with Top-5 Error on ImageNet, Insects and Cifar100

We opted to use top-5 error since it is the most widely reported metric for ImageNet and the top-5 numbers are therefore easy to interpret on that dataset. Many of our datasets have a significantly lower number of classes such that top-5 error makes little sense and we opted to use top-1 for those. This raises the question if comparing top-1 with top-5 errors introduces unwanted perturbations into our analysis. We therefore compare the top-1 and top-5 errors for the three datasets on which we use top-1 error (see Figure 13). We see that the two metrics have an almost linear relationship for the ImageNet and Cifar100 datasets. More importantly are the top-1 to top-5 error graphs monotonically ascending for all three datasets, such that the ordering of architectures does not change when swapping between the two metrics. *Since we are interested in the relative performances of our sampled architectures changing between top-1 and top-5 error does not impact our analysis.*

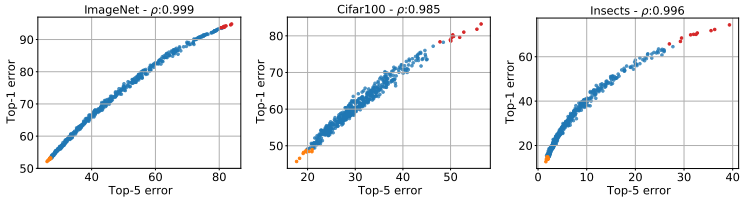

Figure 13: Top-1 error plotted against top-5 error of all 500 architectures on ImageNet, Cifar100, and Insects. The plots reveal that on all three datasets the errors have a very close relationship: it is not perfectly linear but is monotonically ascending.

B.5 OVERFITTING OF HIGH-CAPACITY ARCHITECTURES

The best architectures on Powerline, Natural and Cifar100 have a very small cumulated depth, so it is only natural to ask if the deeper architectures perform poorly due to overfitting. We address this concern by plotting the *training errors* of Powerline, Natural, and Cifar100 against the cumulative block depths (see Figure 14). The training errors are strongly correlated with the cumulative block depth, just like the test errors. *Plots of the cumulated block depth show almost the same structure for training and test errors. We can therefore exclude overfitting as a reason why the shallower networks perform better on Powerline, Natural, and Cifar100.*

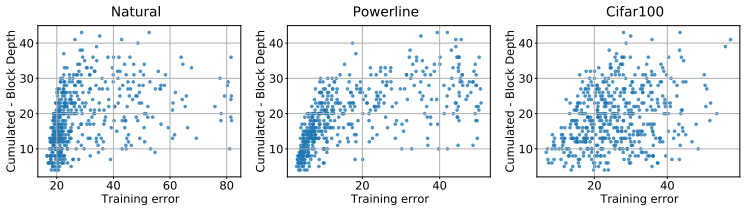

Figure 14: Training errors of the sampled architectures (x-axis) plotted against the cumulated block *depth* for the 3 datasets that have the lowest test errors on shallow architectures. We observe that for all three datasets shallow architectures also have the lowest training errors. Therefore overfitting is not the cause of this behaviour.

B.6 DISENTANGLING THE EFFECTS OF CLASS COUNT AND DATASET SIZE

A core contribution of our paper is that we show how sub-sampled versions of ImageNet matching the number of classes of the target dataset tend to represent the APR of said target dataset far better. A side effect of downsampling ImageNet to a specific number of classes is that the total number of images present in the dataset also shrinks. This raises the question if the increase in error correlation is actually due to the reduced dataset size rather than to the matching class count. We disentangle these effects by introducing another downsampled version of ImageNet, Imagenet-1000-10. It retains all 1000 classes but only 10 examples per class resulting in a datastet with the same number of classes as ImageNet but with the total number of images of ImageNet-10. We train our population of architectures on ImageNet-1000-10 and show the error relationship of Cifar10, Natural, and Powerline with ImageNet-1000-10 (as well as with ImageNet and ImageNet-10 as a reminder) in Figure 15. The plots show that there are some correlation gains by using ImageNet-1000-10 over ImageNet, but the effect is far lower compared to ImageNet-10. *This shows that downsampling size has a minor positive effect but the majority of the gain in APR similarity achieved trough class downsampling actually stems from the reduced the class number.*

B.7 IMPACT OF CLASS DISTRIBUTION

MLC2008 and HAM1000 have a strong class imbalance. They both have one class which makes up a large amount of the dataset. In order to study the impact of an imbalanced class distribution, we created two new more balanced datasets out of the existing data the following way: we reduced the number of samples in the overrepresented class such that it has the same amount of samples as the second most common class. We call these datasets MLC2008-balanced and HAM10000-balanced. Their new class distributions can be seen in Figure 16. We train our architecture population on MLC2008-balanced and HAM10000-balanced leaving the training configuration otherwise unaltered. Figure 17 shows the errors on the balanced datasets versus the errors on the unbalanced counterparts.

For both HAM10000 and and MLC2008, there is a strong correlation between the errors on the balanced and unbalanced datasets. *We can therefore conclude that class imbalance is not a determining factor for the APRs of HAM10000 or MLC2008.*

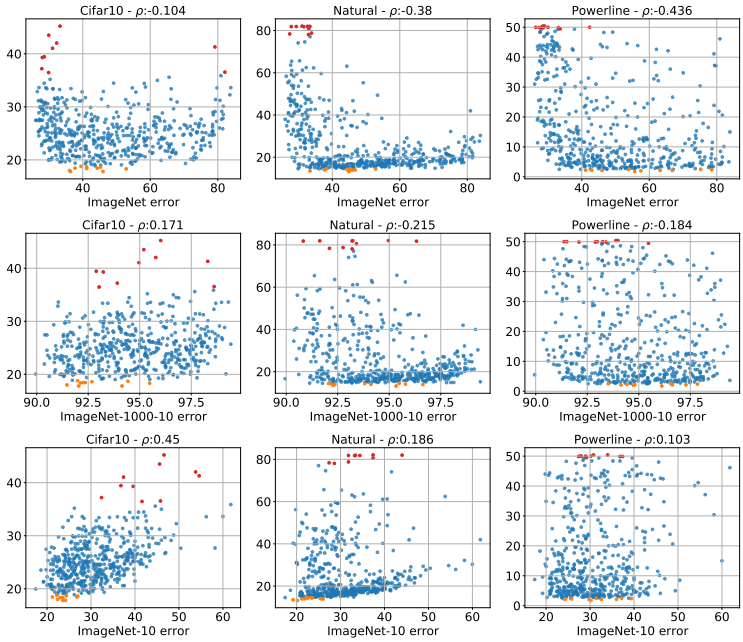

Figure 15: The errors of all 500 architectures on Cifar10, Natural, and Powerline plotted against the errors on ImageNet (top row), ImageNet-1000-10 (middle row) and ImageNet-10 (bottom row). We observe that class-wise downsampling has the largest positive effect on error correlation.

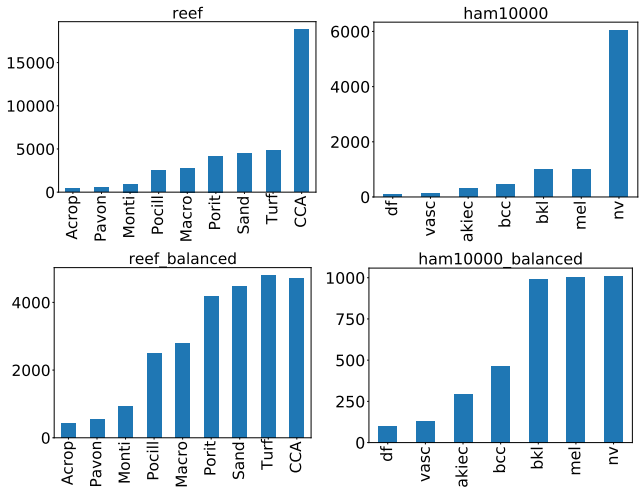

Figure 16: Class distributions of MLC2008, HAM10000, and their balanced versions.

C Impact of Pretraining

The main objective of our study is to identify how well different CNN designs perform on varying datasets and if the best architectures are consistent across the datasets. For this reason we train all of our networks from scratch on each dataset. However, we cannot ignore that pretraining on ImageNet is a huge factor in practice and we therefore study its impact on our evaluations. To this end have we train all of our sampled architectures again on each dataset but this time we initialize their weights with ImageNet pretraining (we omit Concrete, which has very low errors even without pretraining). Figure 18 shows the errors of each dataset without (blue) and with (green) pretraining plotted against the ImageNet errors. The data shows a distinct trend: the performance improvement due to pretraining

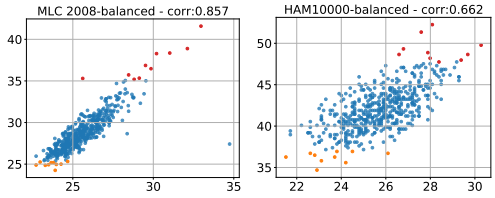

Figure 17: Errors of all 500 sampled architectures on MLC2008-balanced and HAM1000-balanced (y-axis) plotted against the errors of their unbalanced counterparts (x-axis). The top 10 performances on the target dataset are plotted in orange, the worst 10 performances in red. We observe a clear positive correlation for both datasets, hence we conclude that the dataset imbalance has a limited impact on the APRs.

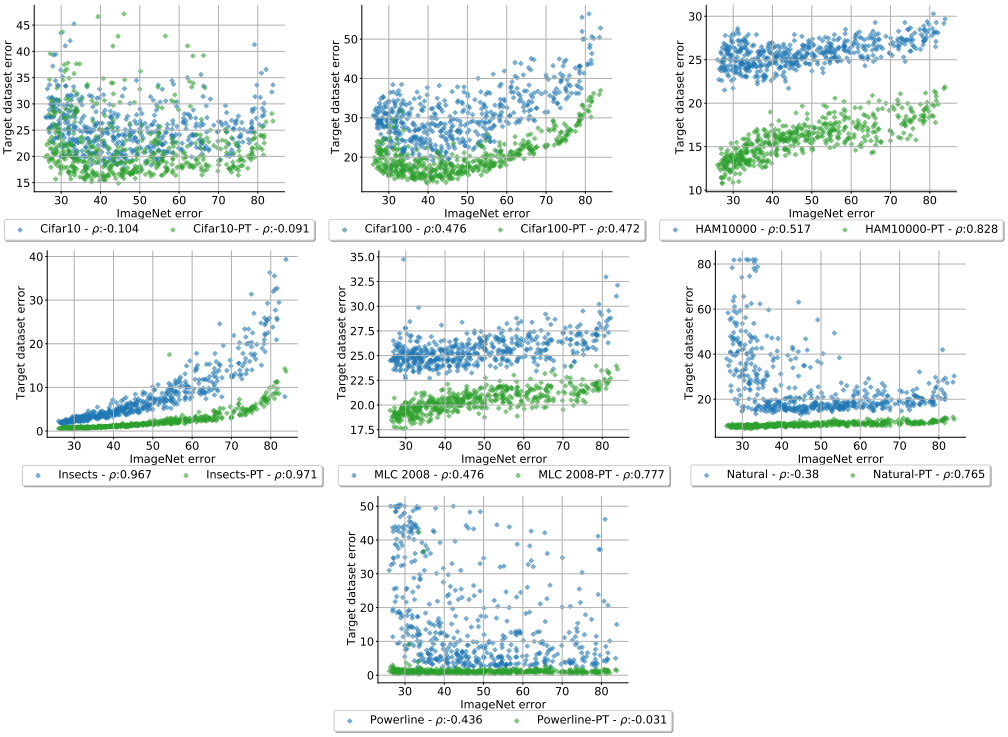

Figure 18: Errors form all 500 architectures trained from scratch (blue) as well as the same architectures pretrained on ImageNet (green), plotted against the respective ImageNet errors. We observe that the error correlation with ImageNet increases relative to the performance gain due to pretraining.

dictates how much stronger the correlation of the pretrained errors with the ImageNet errors is. For Cifar10 and Cifar100 where the performance gain with pretraining is low to moderate the error correlations do not drastically change. On the other end of the spectrum are Natural and Powerline, where pretraining leads to drastically lower errors. This in turn leads to much higher error correlation with ImageNet(the Powerline correlation can not grow significantly above 0 because the overall errors are so small across all architectures). *This shows us that our findings still apply when pretraining is used, but their effects can be masked when pretraining is the most important factor contributing to the overall final performance.*

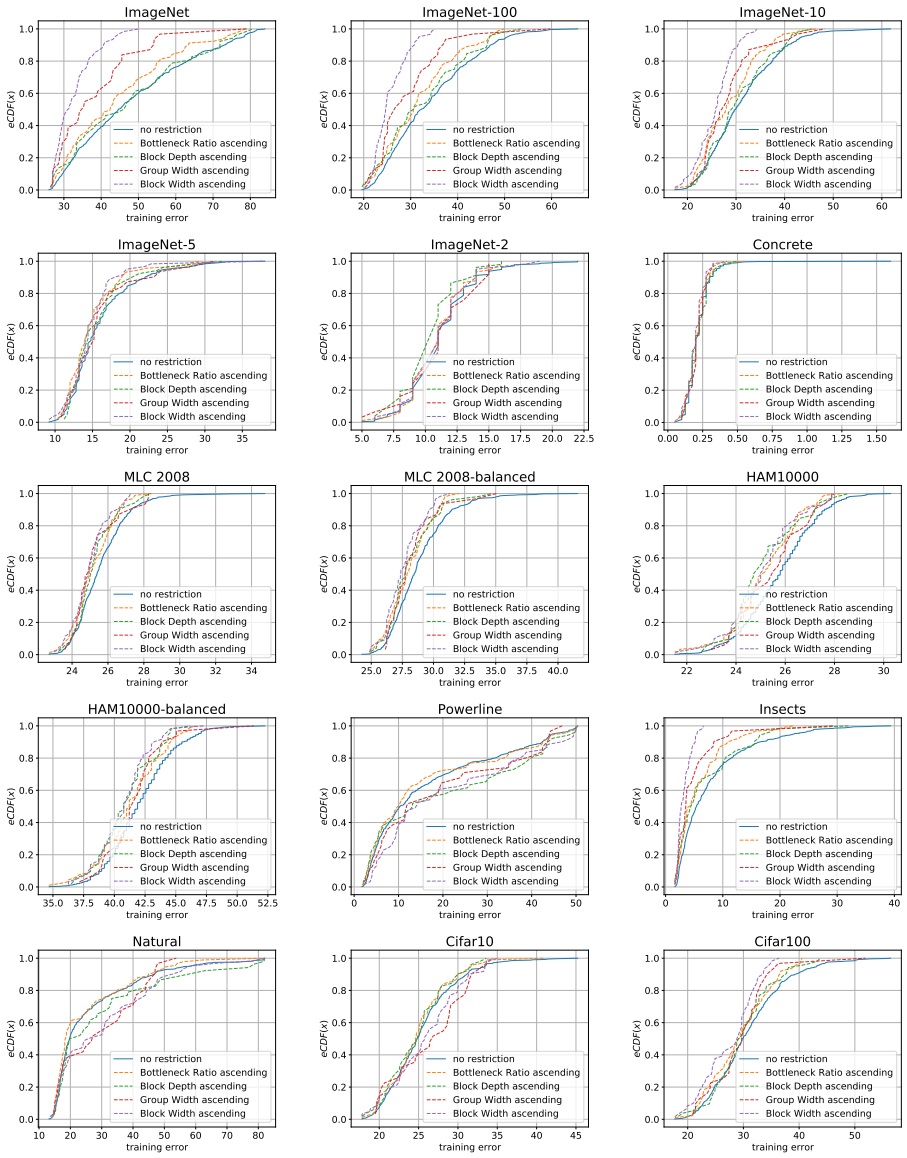

Figure 19: The $eCDFs$ of the restricted sub-populations in the context of the $eCDF$ of the whole architecture population, for all datasets.

D Restricting Parameters to be Monotonically Increasing Along Stages

We have observed that architectures with increasing parameter values across successive stages have tend to perform better on ImageNet. We thus study the impact of restricting the architectures to having strictly increasing parameter values across the stages on the other datasets. We create four subsets of our architecture population by filtering it for ascending bottleneck ratio, block depth, group width and block width each individually.

Figure 19 shows the $eCDF$ of the whole pouplation as well as the $eCDFs$ of the restricted sub-populations for every dataset. For most datasets the restrictions lead to a population with a better error distribution. The effect is most impactful for the models with ascending block width on ImageNet and Insects. Again, Powerline, Natural and to some extent Cifar10 are the odd ones where most restrictions lead to a worse overall performance.

E STRUCTURE OF TOP PERFORMING ARCHITECTURES

Figure 20 shows the configuration of the top performing architecture in blue, as well as the mean and standard deviation of the top 15 configurations for every dataset. We observe that the top 15 architectures have very high variance in both bottleneck ratio and group width.

Block width on the other hand shows a clear pattern: almost all high-performing architectures start with a very small block width that increases across the stages. Only Powerline and Natural do not show this pattern. In block depth, we observe a similar pattern with a bit more noise. For block depth, Powerline, Natural, Cifar10 and Cifar100, no such trend of increased parameter values towards the later stages is observed.

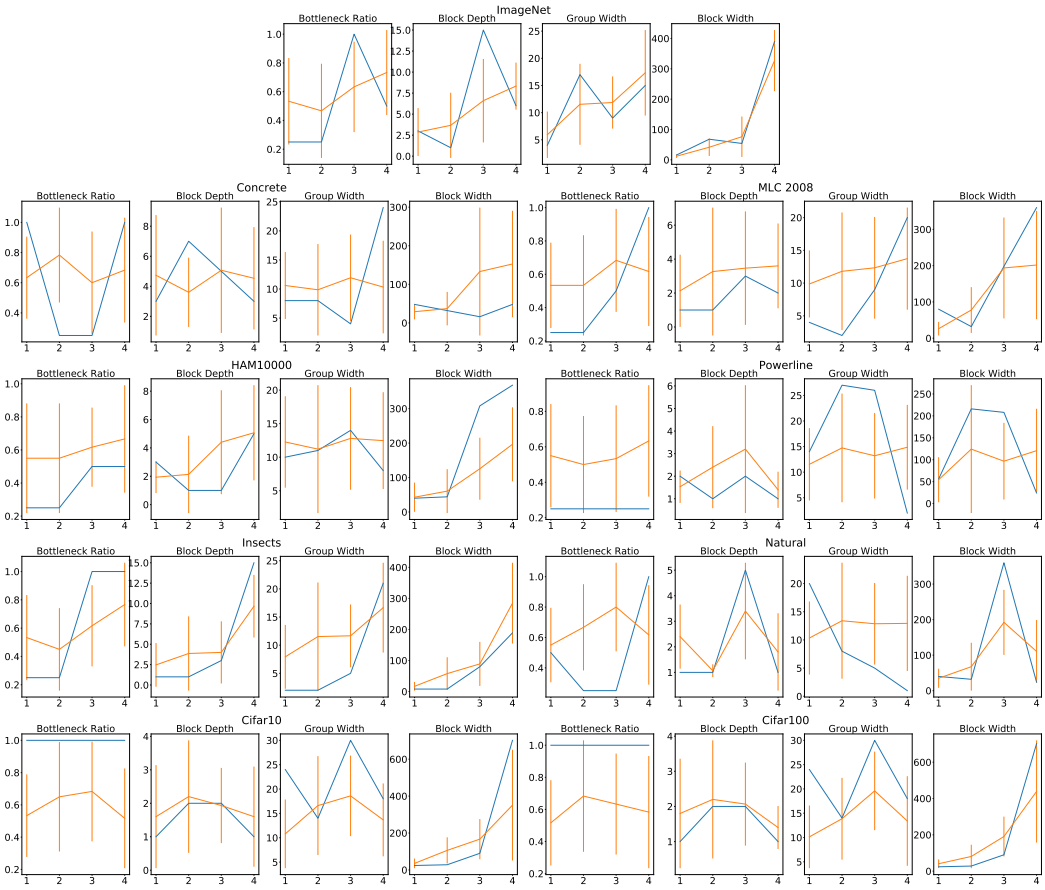

Figure 20: Configurations of the top-performing architectures, with the four stages depicted on the x-axis and the parameter values on the y-axis. The best architectures are shown in blue, the mean of the top 15 architectures is depicted in orange with with a vertical indication of one standard deviation.

F ERROR SCATTERPLOT MATRIX OF ALL DATASETS

Figure 21 shows a matrix of the error scatterplots between the datasets (concrete has been omitted). It reveals that Powerline and Natural not only have low correlation with ImageNet but also with most of the other datasets making these two truly particular datasets. Interestingly is the correlation between Powerline and Naural relatively high, which suggests that there is a common trait that makes these two datasets behave differently. The only negative correlations of Powerline and Natural happen with ImageNet and Insects, the two datasets with a high class count, further reinforcing our point that class count matters. MLC 2008, HAM10000 and Cifar100 have a correlation of 0.69 with each other which indicates that they induce a very similar APR. This APR seems to be fairly standard since MLC 2008, HAM10000 and Cifar100 have a moderate to high correlation with all other datasets.

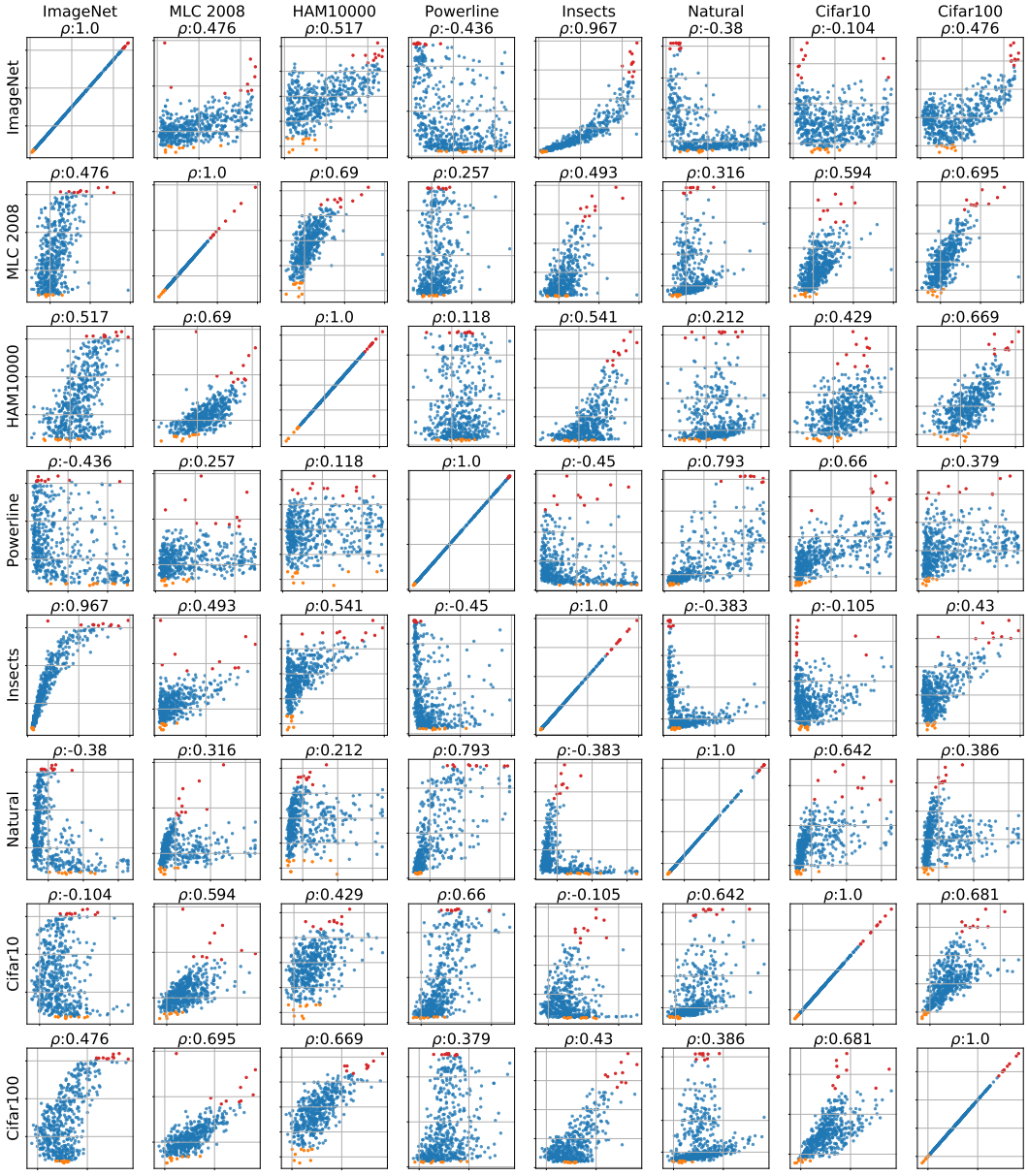

Figure 21: Matrix of error scatterplots of all datasets except Concrete (The first row replicates plots shown in Figure 3).

G COMPLETE PARAMETER BY STAGE BY DATASET SCATTERPLOTS

Figures 22 to 34 show parameter versus error scatterplots, split across the dimensions dataset, stage and parameter. A visual inspection of these plots reveals that block depth and block width show the most tangible patterns.

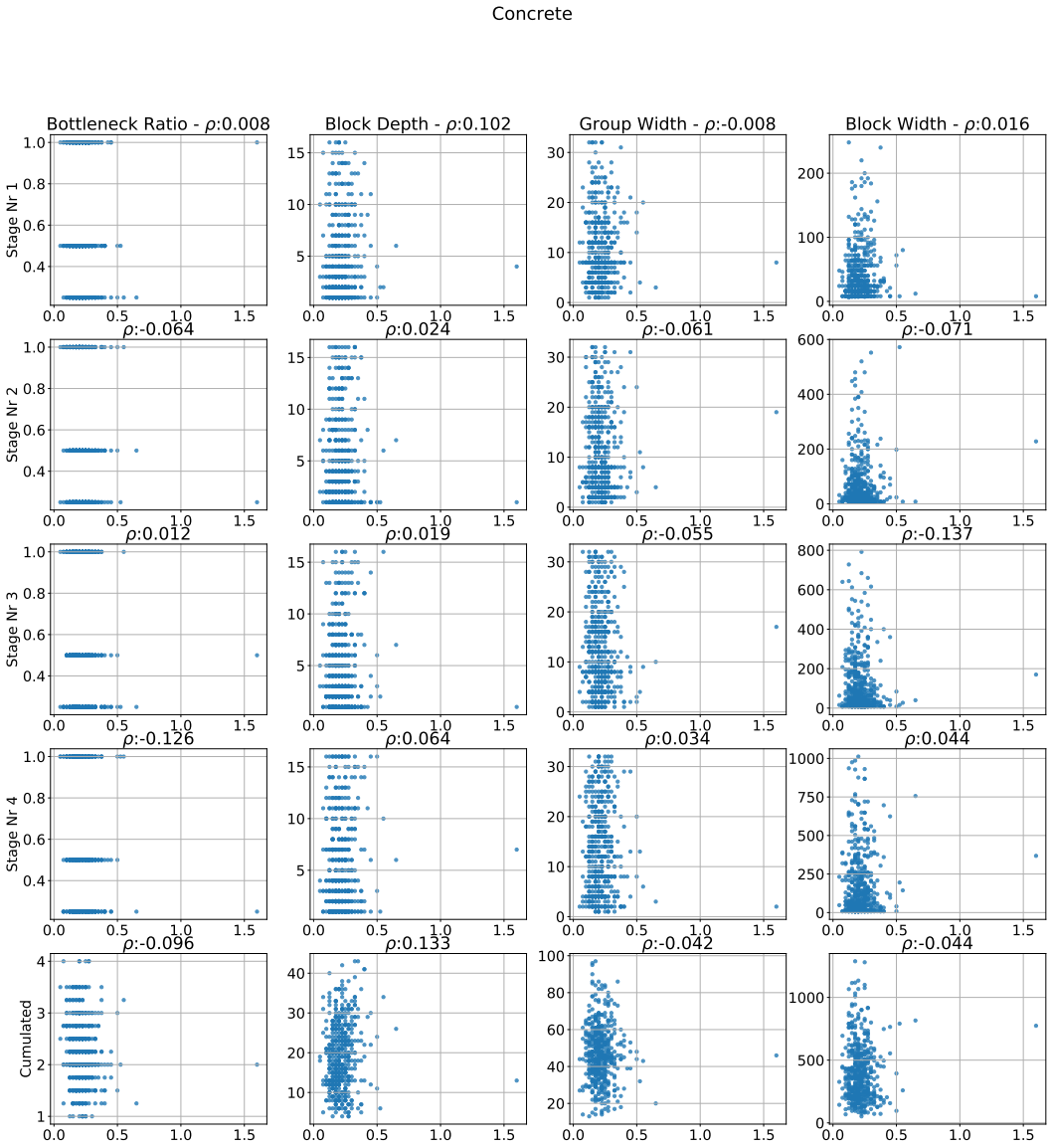

Figure 22: Individual parameter by stage versus error scatterplots for the Concrete dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

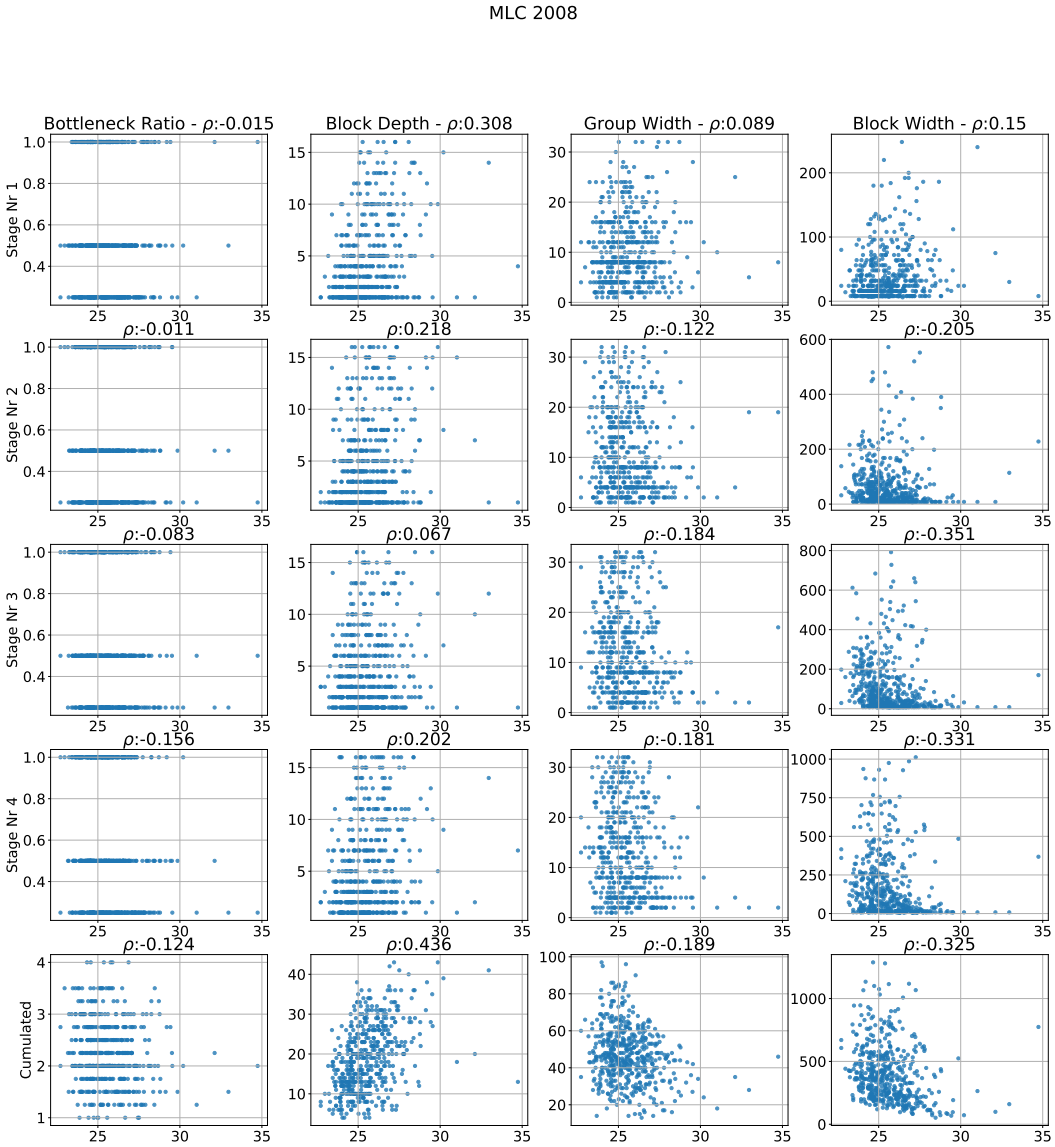

Figure 23: Individual parameter by stage versus error scatterplots for the MLC2008 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

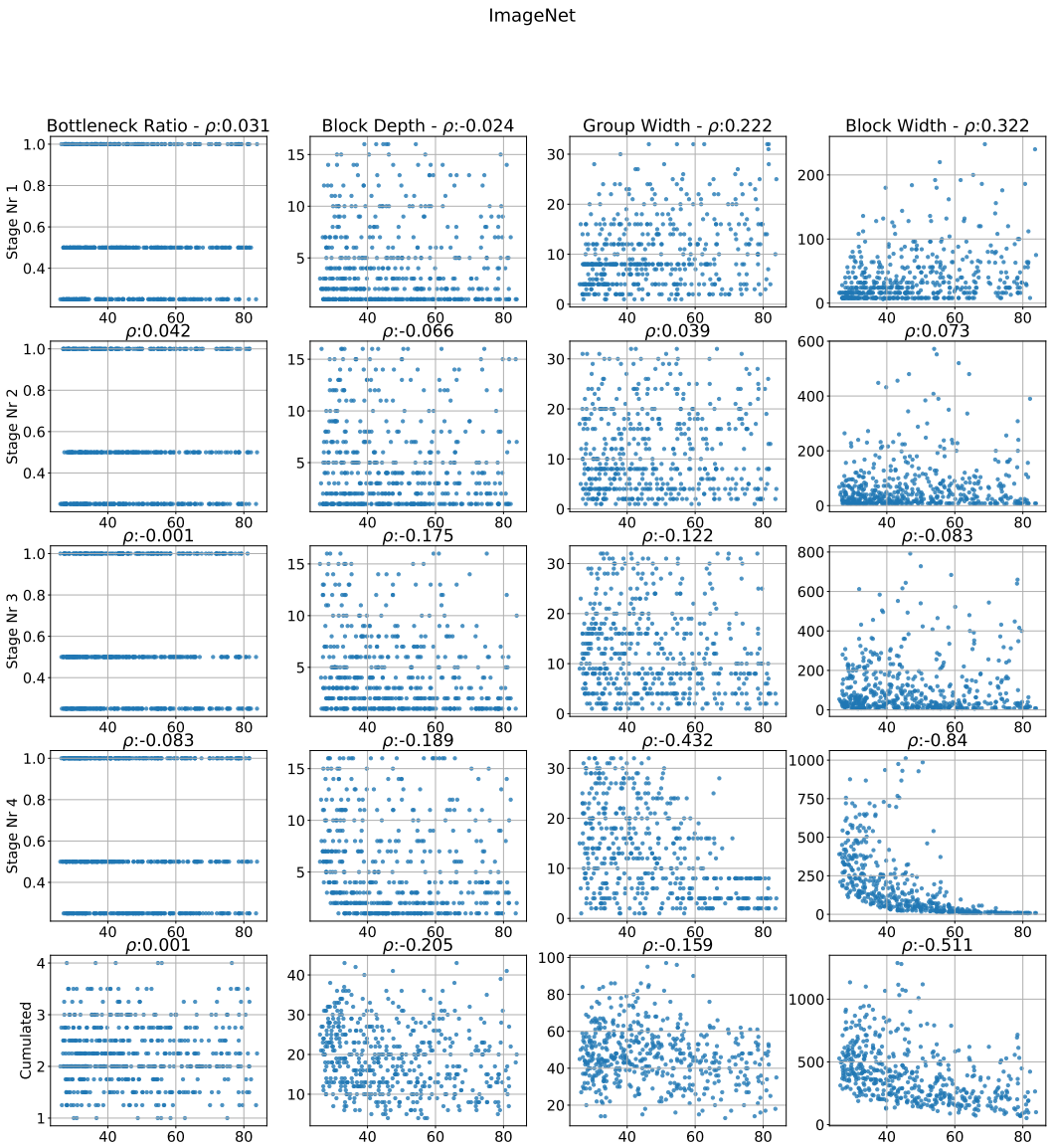

Figure 24: Individual parameter by stage versus error scatterplots for the ImageNet dataset. The x-axis gives the error, while the parameter values are given on the ordinate. The x-axis gives the error, while the parameter values are given on the ordinate.

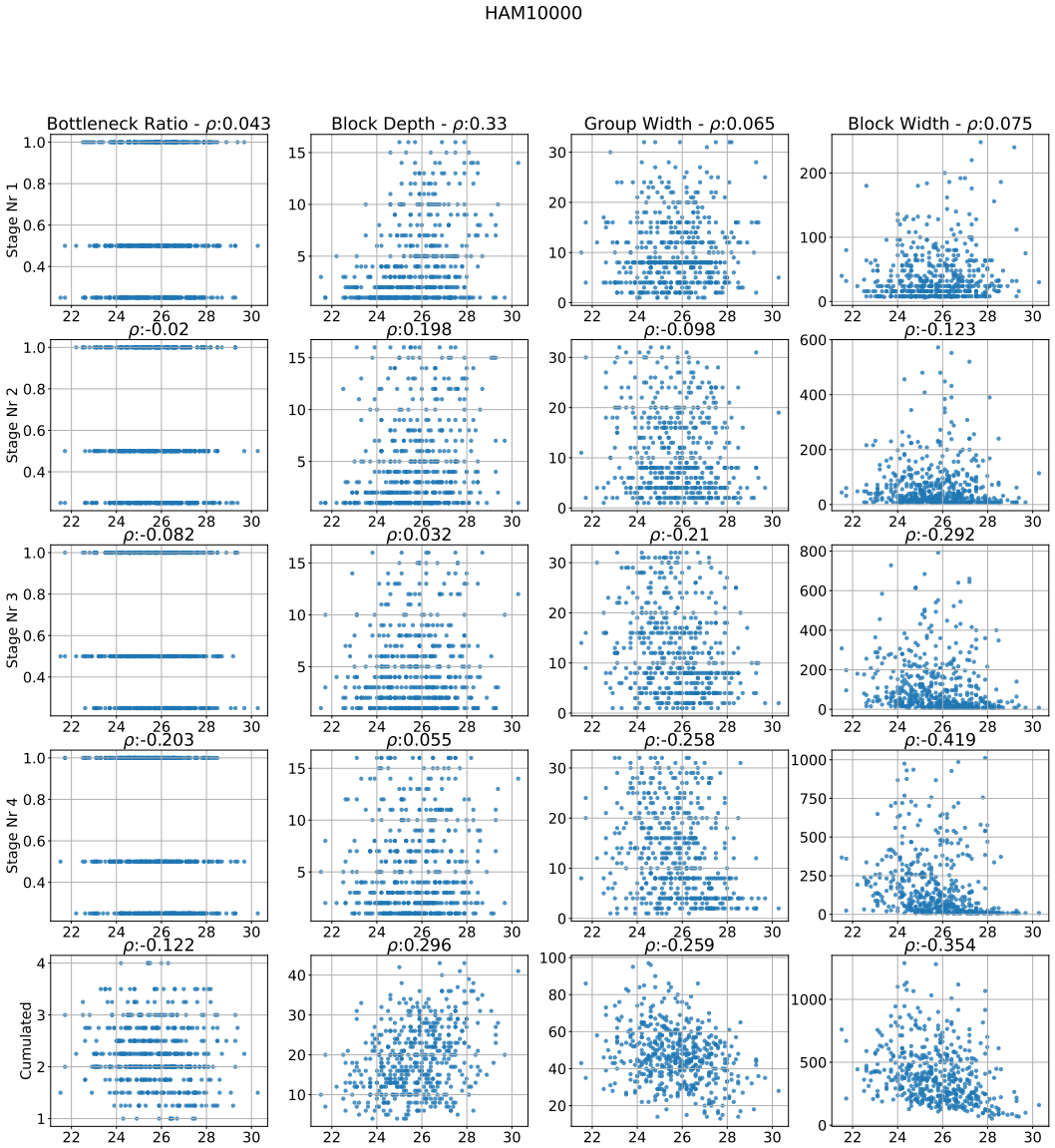

Figure 25: Individual parameter by stage versus error scatterplots for the HAM10000 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

Figure 26: Individual parameter by stage versus error scatterplots for the Powerline dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

Figure 27: Individual parameter by stage versus error scatterplots for the Insects dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

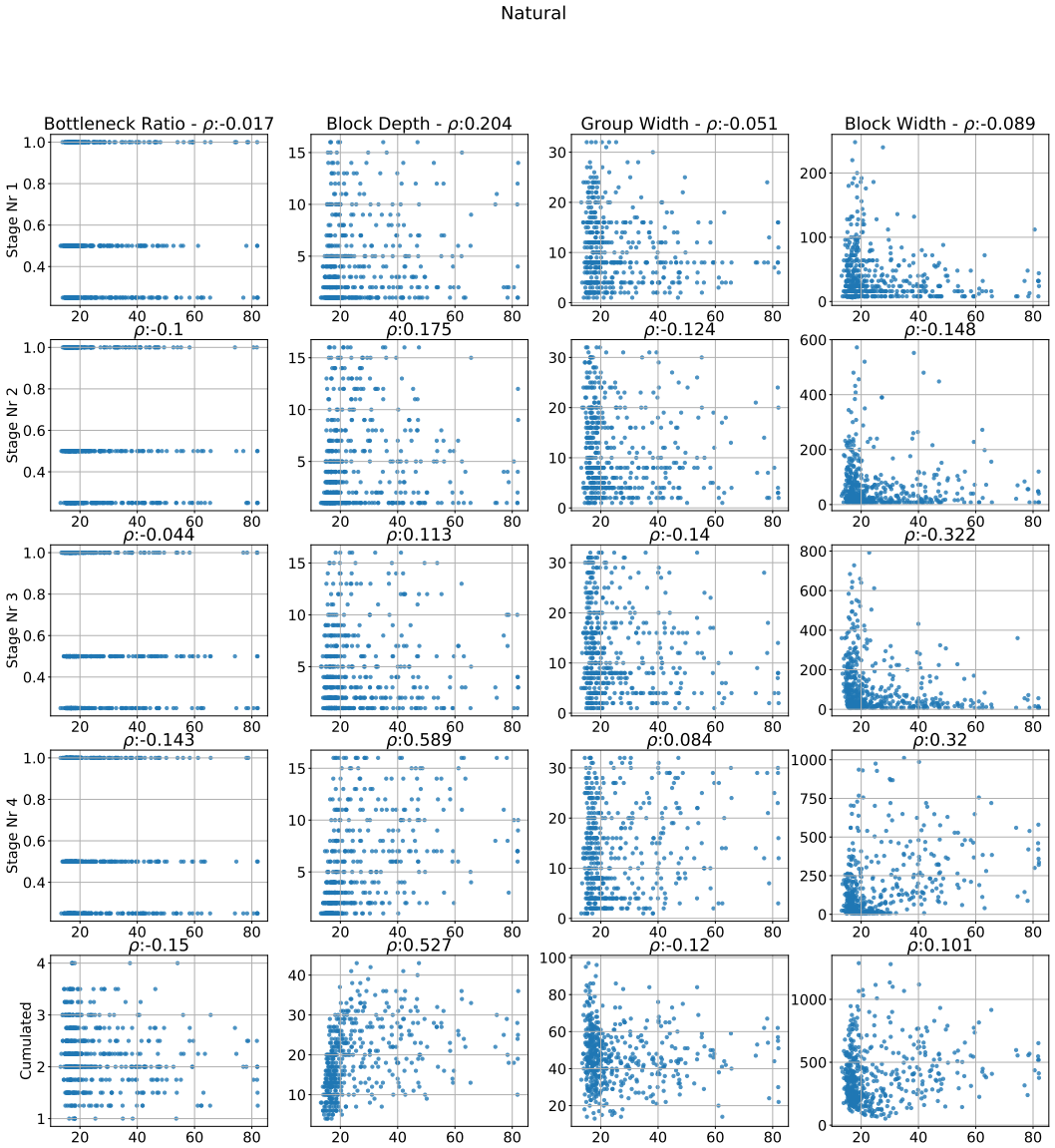

Figure 28: Individual parameter by stage versus error scatterplots for the Natural dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

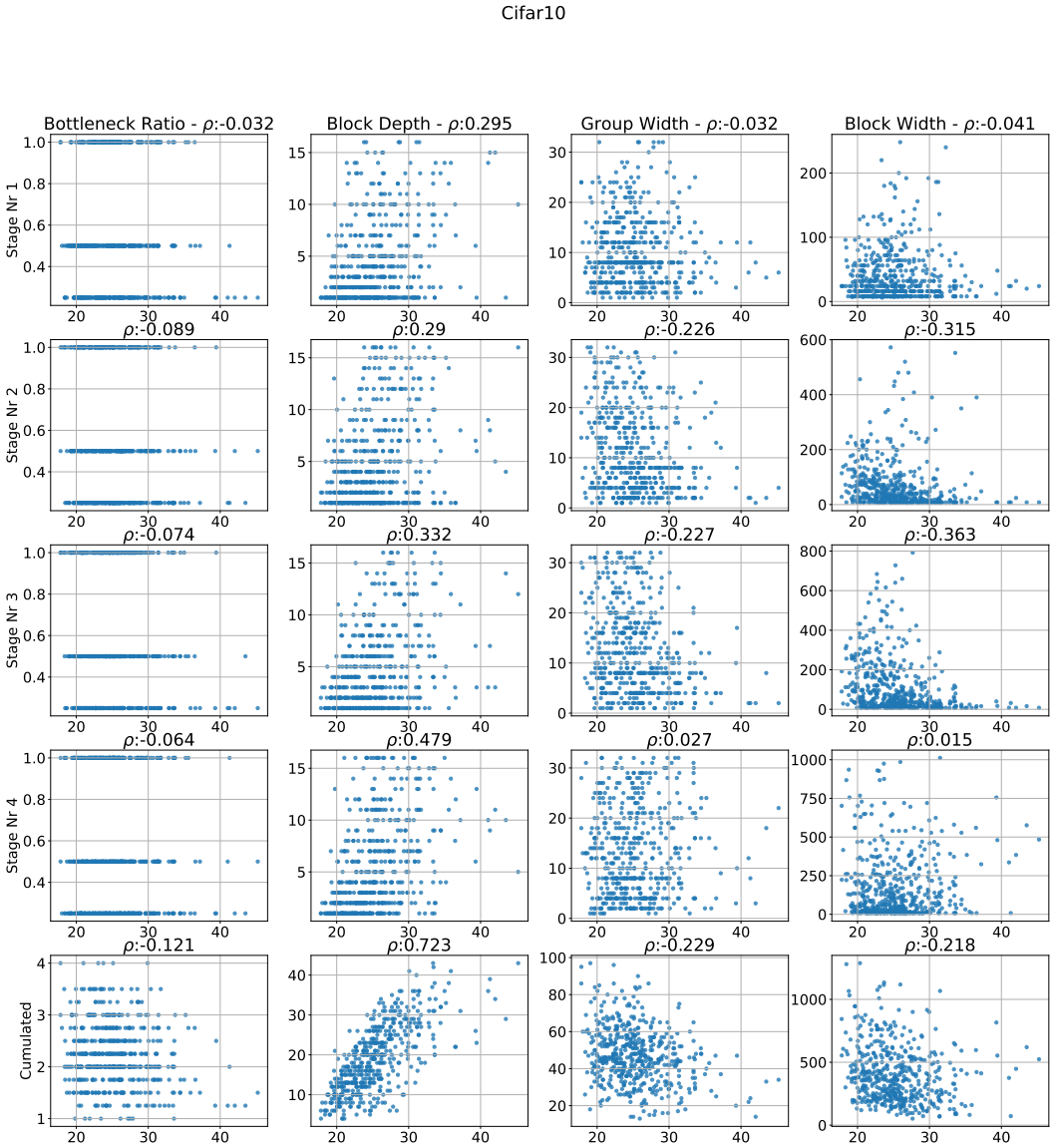

Figure 29: Individual parameter by stage versus error scatterplots for the Cifar10 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

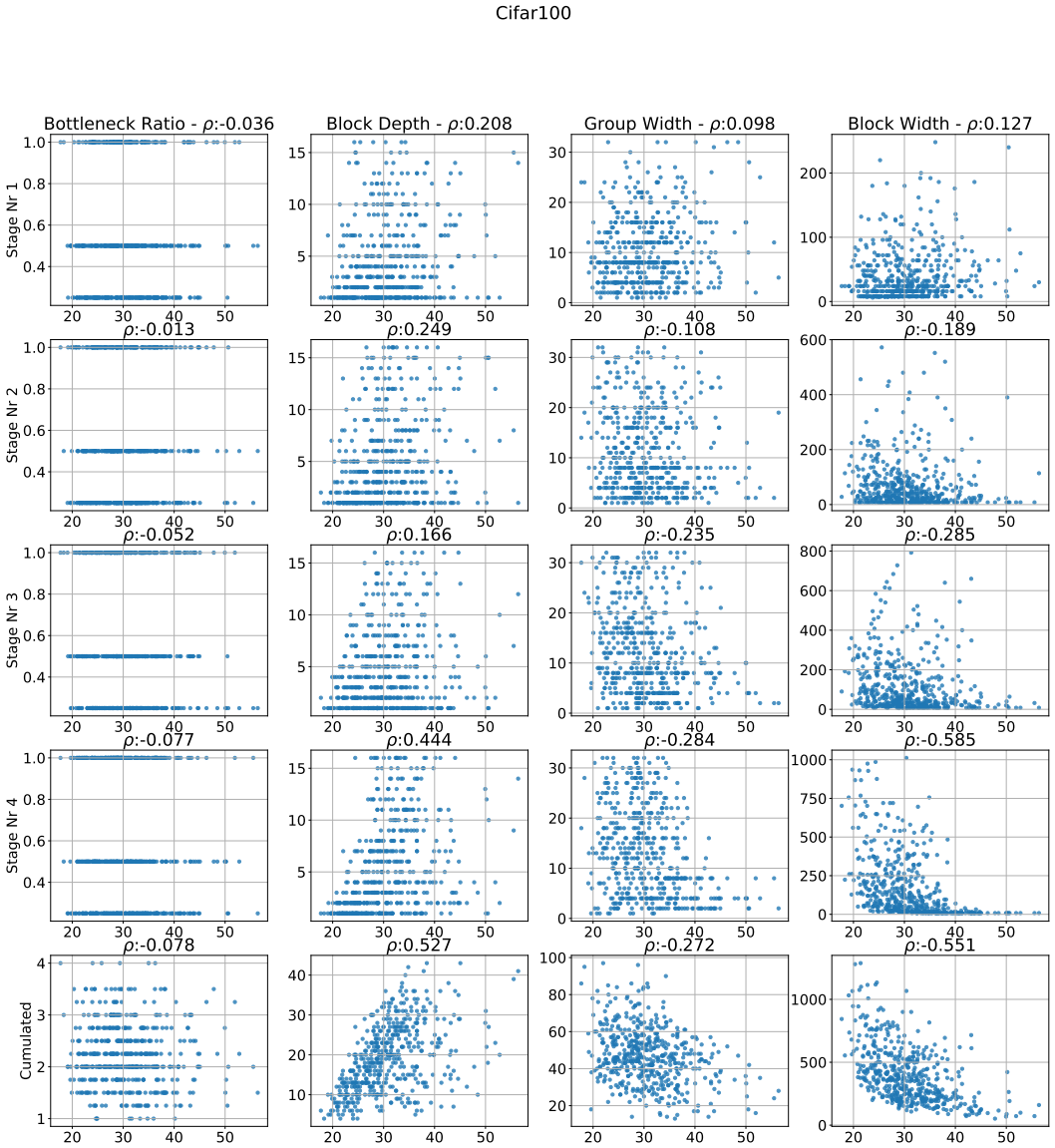

Figure 30: Individual parameter by stage versus error scatterplots for the Cifar100 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

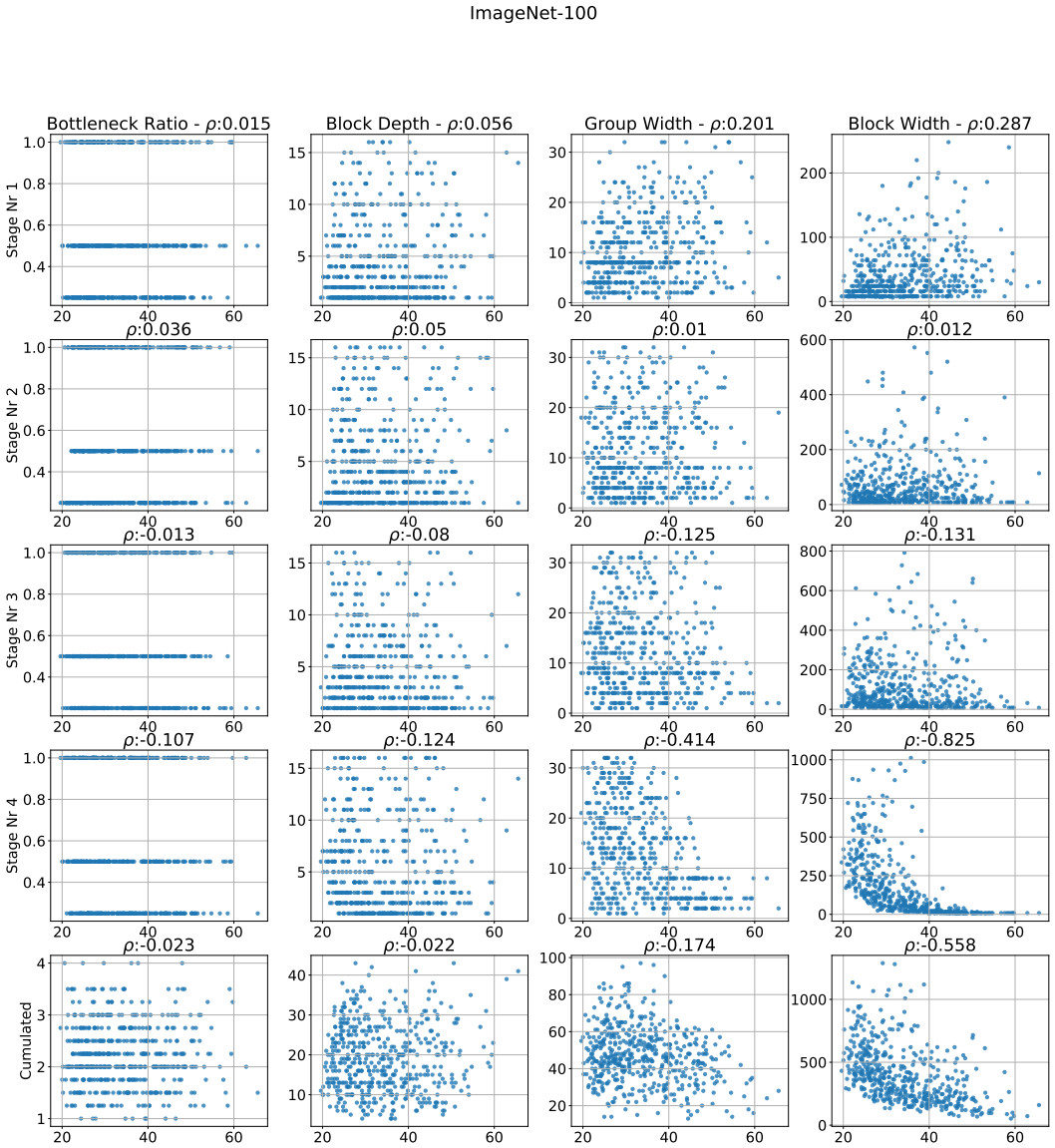

Figure 31: Individual parameter by stage versus error scatterplots for the ImageNet-100 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

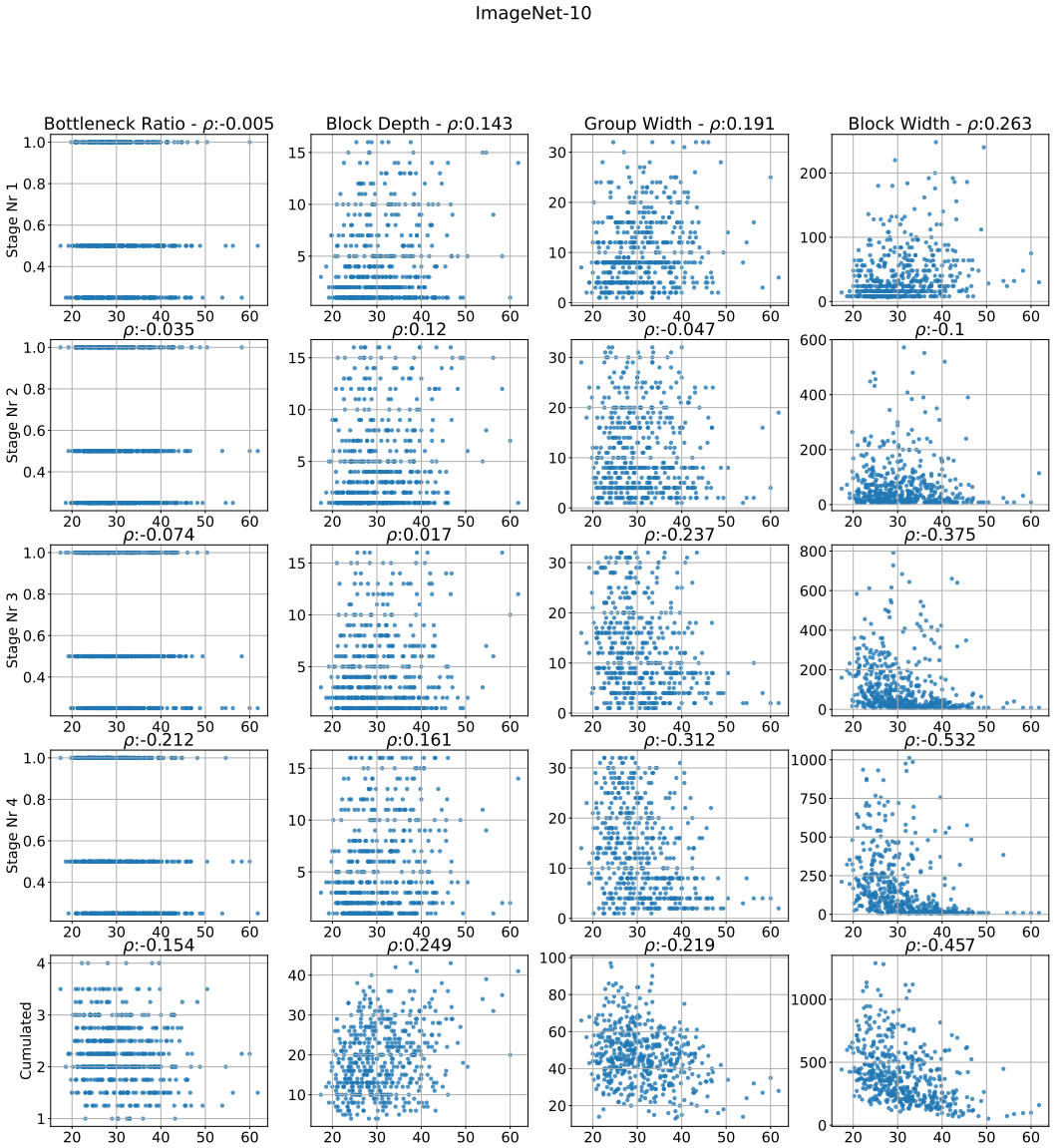

Figure 32: Individual parameter by stage versus error scatterplots for the ImageNet-10 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

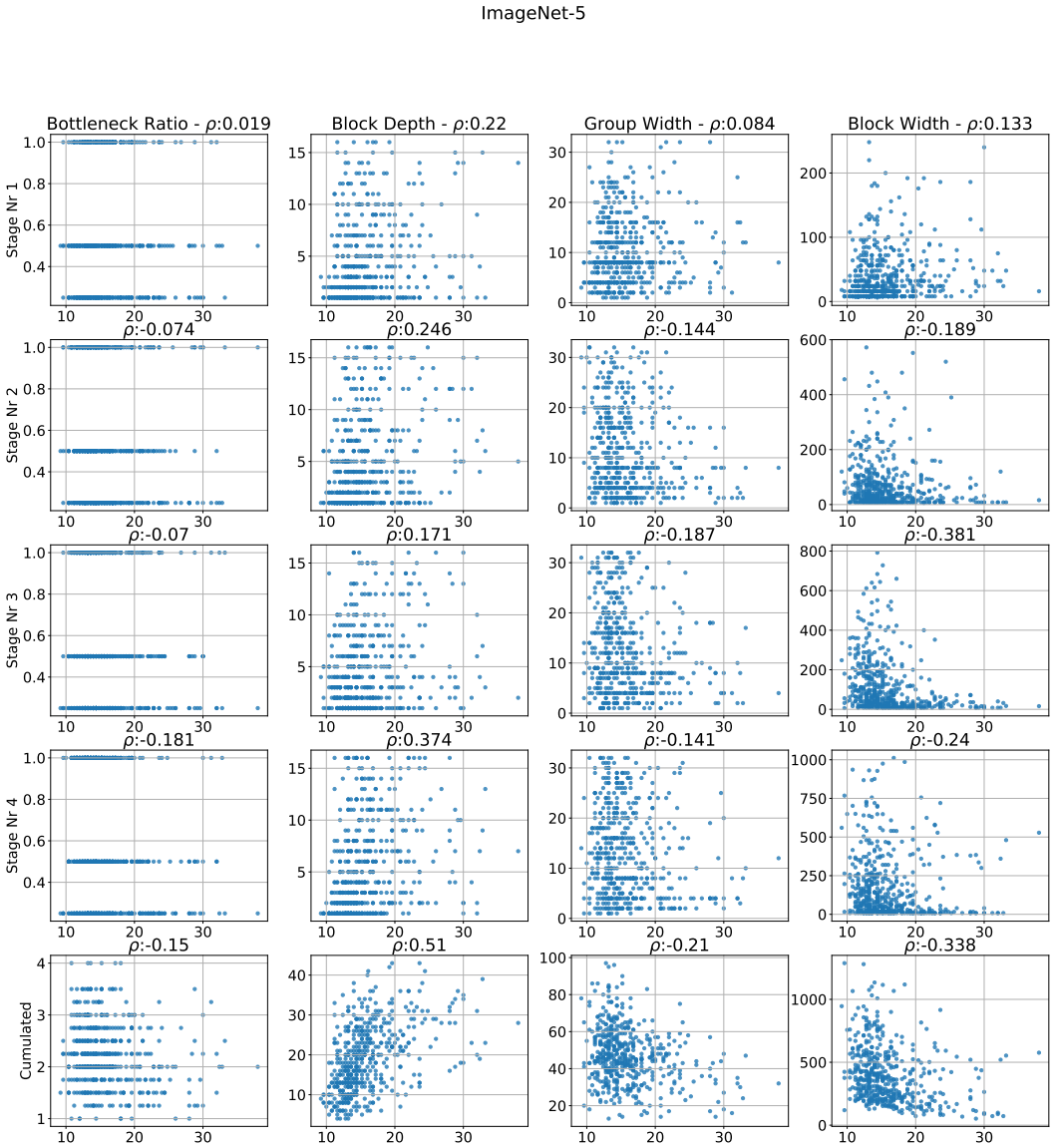

Figure 33: Individual parameter by stage versus error scatterplots for the ImageNet-5 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.

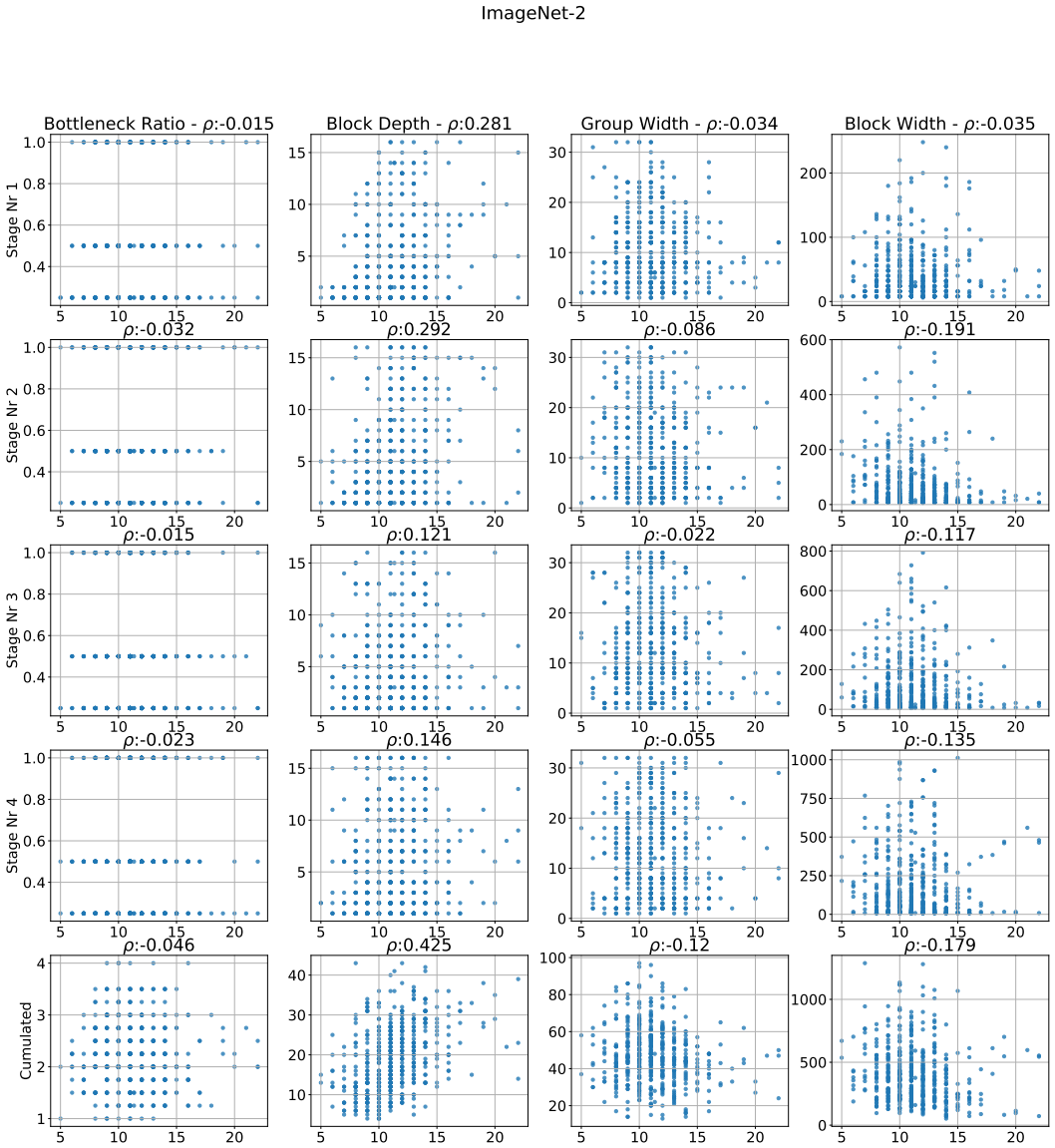

Figure 34: Individual parameter by stage versus error scatterplots for the ImageNet-2 dataset. The x-axis gives the error, while the parameter values are given on the ordinate.