How much Data is Augmentation Worth?

Anonymous Authors[1]

Abstract

Despite the clear performance benefits of data augmentations, little is known about why they are so effective. In this paper, we disentangle several key mechanisms through which data augmentations operate. Establishing an *exchange rate* between augmented and additional real data, we find that augmentations can provide nearly the same performance gains as additional data samples for in-domain generalization and even greater performance gains for out-of-distribution test sets. We also find that neural networks with hard-coded invariances underperform those with invariances learned via data augmentations. Our experiments suggest that these benefits to generalization arise from the additional stochasticity conferred by randomized augmentations, leading to flatter minima.

1. Introduction

Even with the proliferation of large-scale image datasets, deep neural networks for computer vision parameterize highly flexible model families and often contain orders of magnitude more parameters than the size of their training sets. As a result, large models trained on limited datasets still have the capacity for improvement (LeCun et al., 1998a). The importance of data augmentation for boosting performance leads us to wonder whether it benefits training through more complex mechanisms than simply adding more data.

In addition to adding extra samples, augmentation may regularize models by promoting invariance. Just as classifiers learn to make the same prediction across samples in each class, data augmentations (DA) encourage models to make consistent predictions across augmented views of each sample. Data augmentations thus promote invariance by imposing implicit model constraints that can assist in the underdetermined model regime. We can use our a-priori beliefs about which invariances are present in the data to design augmentations. The need to incorporate invariances in neural networks has motivated the development of architectures that are instead explicitly constrained to be equivariant to transformations (Weiler and Cesa, 2019; Finzi et al., 2020). If the downstream effects of data augmentations were attributable solely to invariance, then we could replace DA with explicit model constraints. However, data augmentations may affect training dynamics beyond imposing constraints.

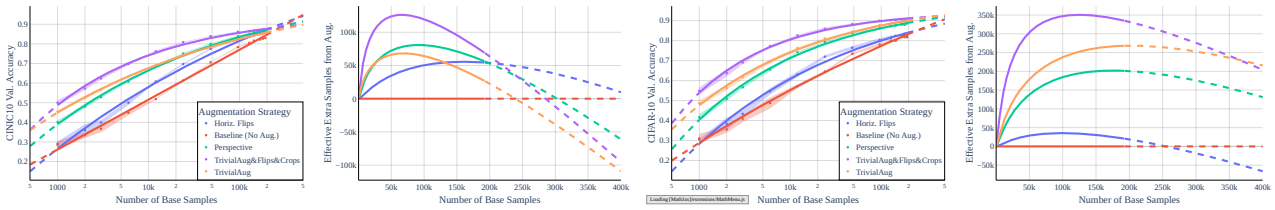
In addition to promoting invariance, augmentation serves as an extra source of stochasticity. Under DA, randomization during training comes not only from randomly selecting samples from the dataset to form batches but also from sampling transformations with which to augment data (Fort et al., 2022). Stochastic optimization is associated with benefits in non-convex problems wherein the optimizer can bias parameters towards flatter minima (Jastrzębski et al., 2017; Geiping et al., 2021; Liu et al., 2021).

In this paper, we revisit the role of data augmentation, as it may be more multi-faceted than the classical view of extra data:

- We quantify the relationship between augmented views of training samples and extra data. We find that augmentations can confer comparable benefits to independently drawn samples on in-domain test sets and even stronger benefits on slightly out-of-distribution testing.
- We observe that models which learn invariances via data augmentation consistently outperform architectures that are instead constrained with equivariance to the same transformations, suggesting that DA regularizes models beyond invariance. Moreover, the standard model of data augmentation dictates that one should choose transforms under which the distribution is invariant, and yet we show that invariances which are uncharacteristic of the data distribution still benefit performance.
- We consider a sampling strategy for augmentation that reduces randomness during training by averaging each batch over all transformed views of an image at once and find that DA exhibits flatness-seeking behavior and greater stochasticity during late stages of training.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.



(a) Evaluated on CINIC-10 (in-domain)

(b) Evaluated on CIFAR-10 (slightly out-of-domain)

Figure 1. **Power laws** $f(x) = ax^{-c} + b$ for select augmentations applied randomly and the gain in terms of effective extra samples. Fitted curves marked in solid colors, with extrapolated regions dashed. Figure 1a shows performance on CINIC-10, Figure 1b on CIFAR-10. Left for each: Number of base samples (from CINIC-10) on the logarithmic horizontal axis compared to validation accuracy. The scaling behavior of each augmentation is closely matched by these power laws. Right for each: Number of base samples compared to effective extra data, showing how the benefits of each data augmentation scale as the model is trained on more and more data. Even the minor shift from CINIC-10 to CIFAR-10 completely changes the dynamics of effective samples.

2. Augmentations as Additional Data

A central role of data augmentation is to serve as *extra data* and expand limited datasets used for training large models. In this section, we quantify this property, conducting a series of experiments to quantify the *exchange rates*, which indicate how much data an augmentation is worth – a number of additional samples which yields the same performance boost as the augmentation policy.

We conduct these experiments on the CINIC-10 dataset (Darlow et al., 2018), a drop-in replacement for CIFAR-10 (Krizhevsky, 2009) which contains numerous additional samples. This allows us to train models with augmented data on dataset sizes similar to CIFAR-10, but compare to reference models trained without augmentations on larger datasets. We further use this replacement to illustrate a behavior of augmentations that is often underappreciated: We evaluate the accuracy of the same models trained on CINIC-10 not only on a CINIC-10 validation set, but also the CIFAR-10 validation set. Both datasets are nearly indistinguishable using simple summary statistics (Darlow et al., 2018), yet there is a minor distribution shift caused by different image processing protocols during dataset curation. We argue that this is a reasonable test case for practical scenarios in which there could be even a minor uncertainty about the distribution of test data.

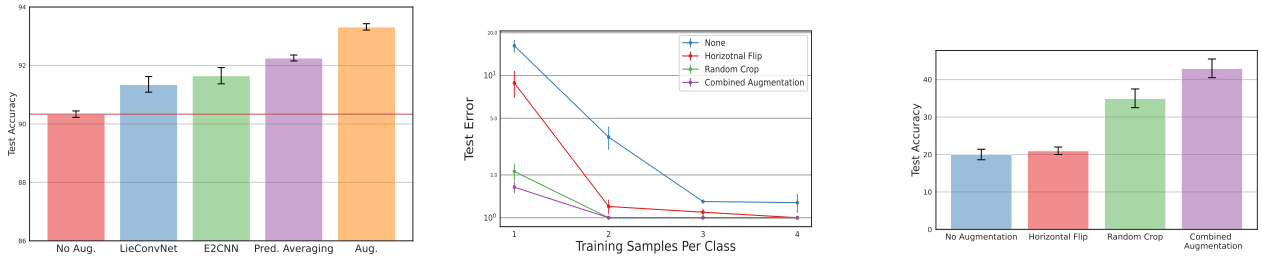
How much *extra data* is gained through data augmentations and how does this number change as the number of base samples increase, or the data distribution shifts? Figure 1 shows that the validation accuracy for this model is well modeled by power laws of the form $f(x) = ax^{-c} + b$ for both validation sets. We then evaluate the relative difference between these power laws in terms of absolute number of extra samples gained through augmentation and extrapolate to larger sample sizes.

Under this view, we see that data augmentations are transient. Investigating Figure 1a, we find that for some amount of base samples from the original dataset, the effectiveness of each augmentation peaks and the augmentation effectively generates the most extra data; then with more base samples, the benefits of all augmentations diminish up to a point where we extrapolate that they would not be beneficial any more and may in fact hurt performance. Here, we also see a reversal of the trends from our previous study on diversity. The diverse augmentations such as TrivialAug peak early. Yet, the consistent, and least diverse, augmentations of horizontal flips falls off slowest, showing that with enough real samples, diverse, but inconsistent policies make poorer augmentations.

Yet, this behavior is highly contingent on the evaluation on in-domain data. Even, the shift to evaluation on CIFAR-10 in Figure 1b reveals that while the transient behavior described in the previous paragraph is still present, the actual extra data gained is much more significant.

3. Augmentations are Worth More Than Invariance

A different angle on data augmentations is that they encourage invariance to image transformations by enforcing the assignment of identical labels across transformations of each training sample. If the success of data augmentations can be attributed solely to invariance, then we can build *exactly invariant* models which achieve comparable accuracy when trained without data augmentation. Several works propose mechanisms for constraining neural network layers to be invariant, and we will leverage these in our study. In this section, we find that the benefits of data augmentation can only partially be recovered by building invariant models. Moreover, we discover that augmentations which reflect invariances not present in the data distribution can still offer performance boosts, indicating that the benefits of data augmentation are not limited to invariances which exist in the data distribution.



(a) Data augmentations outperform invariant networks for normal training on CIFAR-10.

(b) Accuracy vs samples for each class for different augm. . Both the train and the test data are different views of the same image.

(c) The accuracy for different augm. for rotated samples from CIFAR-10 where the train and the test data are different images.

Figure 2. Figure 2a shows data augmentations vs invariant models for experiments performed on CIFAR-10 with a ResNet-18. LieConvNet and E2CNN are invariant w.r.t. horizontal flips, and the right-most bar corresponds to random horizontal flips. Section 2: Test error (log-scale) as a function of training samples when test images are rotations of training images. Figure 2c Test accuracy on rotated samples from CIFAR-10 other than those used for training.

One might think that the ability to remain only approximately invariant to transformations is a valuable feature of data augmentations, since a particular invariance might be inconsistent with the data distribution (Finzi et al., 2021). However, a notable property of training on augmented data is that the loss function does not distinguish between raw and augmented images, and so the model is not able to overcome invariances which are inconsistent with the training labels. Therefore, when models are trained with data augmentations, instead of determining whether invariances are consistent with the distribution, they learn only approximate invariances since they may not perfectly fit the augmented training data, and their invariance might not generalize to test data. This fact makes it all the more mysterious that exactly invariant networks underperform those trained with augmentation.

3.1. Invariant Neural Networks Without Augmentation

In order to probe the benefits of invariance without augmentations, we adopt the following three methods for constructing invariant networks. **Prediction averaging:** Insert augmented views of a sample into the network and average the corresponding predictions (Shanmugam et al., 2021; Gandikota et al., 2021). We use this procedure during both training and inference. Note that this method still involves passing augmented data into the model. **E2CNN:** General E(2)-Equivariant Steerable CNN (E2CNN) (Weiler and Cesa, 2019) constrains convolutional kernels to reflect a group equivariance. **LieConv:** LieConv (Finzi et al., 2020) is a convolutional layer which is equivariant to a user-specified Lie group, yielding entire models which possess this equivariance property. We use LieConv layers in a ResNet-18 model, which we will refer to as LieConvNet.

We employ the above methods to train ResNet-18 models on CIFAR-10 which are exactly invariant to horizontal flips. Figure 2a compares their performance.E2CNN, LieConvNet, and prediction averaging models all improve performance over non-invariant models trained on non-augmented data, there remains a considerable accuracy gap between invariant neural networks and the same architecture trained instead with horizontal flip data augmentation.

3.2. Out-Of-Distribution Augmentations Still Improve Performance

Previously, we observed the performance benefits of data augmentations which promote invariances consistent with the data distribution, or approximately so, even over existent invariant neural networks. But can it still be useful to augment our data with a transformation that generates samples completely outside the support of the data distribution and which are inconsistent with any label? To answer this question, we construct a synthetic dataset in which the exact invariances are known.

We begin by randomly sampling a single base image from each CIFAR-10 class. We then construct 10 classes by rotating each of the base images, so that all samples in a class correspond to rotations of a single image. Thus, the classification task at hand is to determine which base image was rotated to form the test sample. We randomly sample rotations from each of these classes to serve as training data and another disjoint set of rotations to serve as test data. We then use horizontal flip and random crop data augmentations to generate out-of-distribution samples, since horizontally flipped or cropped image views cannot be formed merely via rotation. Note that this experiment is distinct from typical covariate shift setups where the distribution of data domains differs, but the support is far from disjoint and may even be identical.

In Figure 2, we see that these out-of-distribution augmentations are beneficial nonetheless. Notably, random crops, which can generate significantly more unique views than horizontal flips, yield massive performance boosts for

Table 1. **End-of-training stochasticity correlates strongly with flatness.** Gradient standard deviation across batches at the end of training and flatness measurements with various augmentations and strategies for sampling augmented views.

Augmentation	Fixed Views	Same Batch	Grad. Std.	Flatness
-	-	-	13.201	11.4192
Flips&Crops	✗	✗	24.7397	16.3284
	✓	✗	13.612	10.1113
	✗	✓	15.676	11.914
	✓	✓	10.263	8.2033
TrivialAug &Flips&Crops	✗	✗	31.339	20.1336
	✓	✗	22.689	16.3189
	✗	✓	25.53	16.1076
	✓	✓	14.105	10.4287
RandAug &Flips&Crops	✗	✗	29.912	18.6886
	✓	✗	16.585	11.5707
	✗	✓	22.0127	13.3998
	✓	✓	10.865	8.1097

identifying rotated images, even though we know that the cropped samples are out-of-distribution. We also see in this figure that random crops are especially useful if we instead use as our test set rotations of other samples from CIFAR-10 than those used from training. Specifically, we assign a base test image and its rotations the same label as the base image from the training set with the same CIFAR-10 label. This experiment supports the observations from Section 2 that augmentations can be particularly beneficial for OOD generalization.

4. Data Augmentation As a Source of Stochasticity During Training

Typical neural network loss functions contain individual terms for each training sample. During optimization, randomly sampling batches of data points (i.e. terms in the loss) and computing gradients on batches rather than the full loss gives rise to stochasticity. Augmentations increase the number of terms in the loss function, often to such an extent that we never sample the same term twice during training.

Since data augmentations expand and diversify the body of samples, and equivalently terms of the loss function, available for sampling, they may serve as additional sources of stochasticity during optimization. As neural networks are typically trained with first-order optimizers, this boost in stochasticity can be examined by measuring the variety of gradients across batches during training. If data augmentations do in fact increase the variety of gradients, then they could as a result cause us to find qualitatively different minima. Stochastic optimization is thought to be associated with flat minima of the loss landscape which are in turn associated with superior generalization (Jastrzębski et al., 2017; Huang et al., 2020; Liu et al., 2021). Moreover, this *flatness-seeking behavior* may be the effect not only of the augmented loss function but also how we sample it.

In this section, we put this hypothesis to the test. We measure the standard deviation of gradients during optimization for models trained with and without data augmentations, and we quantify the flatness of the minima these variants find. We describe how we measure both stochasticity and flatness in the appendix. We construct experiments that disentangle the augmented loss function from the additional stochasticity produced by randomly sampling augmented views. We consider a "same batch" strategy in which gradient updates are averaged over multiple views of a single image, resulting in lower stochasticity. We also consider "fixed views" experiments in which we pre-compute a frozen set of augmented views per element of the training set, which can then be sampled during training.

In Table 1, we see that for each augmentation policy, applying augmentations randomly results in the most stochasticity at the end of training, while including multiple random views in the same batch (Hoffer et al., 2020) results in less. Sampling augmentations from a fixed set of four views per sample (denoted "fixed views") results in even less stochasticity, and including each of the four views in every batch results in the least stochasticity (denoted "fixed vews", "same batch"). This ordering, which holds across all data augmentations we try, is consistent with the intuition that more randomness in augmentation leads to more stochasticity in training. Including each of the four views for each member of the batch entirely removes the randomization of data augmentations but still preserves the same exact possible combinations of base images available for sampling under all other training setups we consider, allowing us to train on the same augmented loss function but without the extra randomization from sampling augmented views. In this case, we have 4 views per base image and 128 base images, making a total of $4 \times 128 = 512$ samples in the batch including augmentations.

We further observe that flatness correlates strongly with late-training stochasticity. Models trained without augmentation or with non-random augmentation, where all views are seen in each batch, exhibit less stochasticity at the end of training and find sharper minima. While previous works have associated SGD with flatness-seeking behavior (Jastrzębski et al., 2017; Geiping et al., 2021), the findings here indicate that data augmentations can also contribute to this phenomenon. Simply put, training with randomized data augmentations finds flatter minima, and models trained with strong data augmentations such as RandAugment and TrivialAugment lie at especially flat minima.

5. Discussion

This work promotes an all-encompassing understanding of neural network training which incorporates the under-appreciated factor that is data augmentation, covering the effective *extra data*, *invariance* and *stochasticity* provided by augmentations.

References

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996. (p. 8)

Randall Balestriero, Leon Bottou, and Yann LeCun. The effects of regularization and data augmentation are class dependent. *arXiv preprint arXiv:2204.03632*, 2022a. (p. 8)

Randall Balestriero, Ishan Misra, and Yann LeCun. A data-augmentation is worth a thousand samples: Exact quantification from analytical augmented sample moments. *arXiv preprint arXiv:2202.08325*, 2022b. (p. 8)

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning Invariances in Neural Networks from Training Data. In *Advances in Neural Information Processing Systems*, volume 33, pages 17605–17616. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/cc8090c4d2791cdd9cd2cb3c24296190-Abstract.html. (p. 8)

Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. MultiGrain: A unified image embedding for classes and instances. *arXiv:1902.05509 [cs]*, April 2019. URL http://arxiv.org/abs/1902.05509. (p. 8)

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. (p. 8)

Shuxiao Chen, Edgar Dobriban, and Jane Lee. A Group-Theoretic Framework for Data Augmentation. In *Advances in Neural Information Processing Systems*, volume 33, pages 21321–21333. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f4573fc71c731d5c362f0d7860945b88-Abstract.html. (p. 8)

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: An extension of MNIST to handwritten letters. *arXiv:1702.05373 [cs]*, February 2017. doi: 10.48550/arXiv.1702.05373. (p. 11)

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data. *arXiv:1805.09501 [cs, stat]*, April 2019a. URL http://arxiv.org/abs/1805.09501. (p. 8, 11)

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. *arXiv:1909.13719 [cs]*, November 2019b. URL http://arxiv.org/abs/1909.13719. (p. 8, 11)

Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Re. A Kernel Theory of Modern Data Augmentation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1528–1537. PMLR, May 2019. URL https://proceedings.mlr.press/v97/dao19b.html. (p. 8)

Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv:1810.03505 [cs, stat]*, October 2018. URL http://arxiv.org/abs/1810.03505. (p. 2, 11)

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020. (p. 1, 3)

Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems*, 34, 2021. (p. 3)

Ruth Fong and Andrea Vedaldi. Occlusions for Effective Data Augmentation in Image Classification. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4158–4166, October 2019. doi: 10.1109/ICCVW.2019.00511. (p. 8)

Stanislav Fort, Andrew Brock, Razvan Pascanu, Soham De, and Samuel L. Smith. Drawing Multiple Augmentation Samples Per Image During Training Efficiently Decreases Test Error. *arXiv:2105.13343 [cs]*, February 2022. URL http://arxiv.org/abs/2105.13343. (p. 1, 8)

Kanchana Vaishnavi Gandikota, Jonas Geiping, Zorah Lähner, Adam Czapliński, and Michael Moeller. Training or architecture? how to incorporate invariance in neural networks. *arXiv preprint arXiv:2106.10044*, 2021. (p. 3)

Jonas Geiping, Micah Goldblum, Phil Pope, Michael Moeller, and Tom Goldstein. Stochastic Training is Not Necessary for Generalization. In *International Conference on Learning Representations*, September 2021. URL https://openreview.net/forum?id=ZBESeIUB5k. (p. 1, 4, 8)

Raphael Gontijo-Lopes, Sylvia Smullin, Ekin Dogus Cubuk, and Ethan Dyer. Tradeoffs in Data Augmentation: An Empirical Study. In *International Conference on Learning Representations*, September 2020a. URL https://

openreview.net/forum?id=ZcKPWuhG6wy. (p. 8)

Raphael Gontijo-Lopes, Sylvia J. Smullin, Ekin D. Cubuk, and Ethan Dyer. Affinity and Diversity: Quantifying Mechanisms of Data Augmentation. *arXiv:2002.08973 [cs, stat]*, June 2020b. URL http://arxiv.org/abs/2002.08973. (p. 8, 11)

Boris Hanin and Yi Sun. How Data Augmentation affects Optimization for Linear Regression. In *Advances in Neural Information Processing Systems*, volume 34, pages 8095–8105. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/442b548e816f05640dec68f497ca38ac-Abstract.html. (p. 8)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. (p. 10)

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of Tricks for Image Classification with Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019a. URL https://openaccess.thecvf.com/content_CVPR_2019/html/He_Bag_of_Tricks_for_Image_Classification_with_Convolutional_Neural_Networks_CVPR_2019_paper.html. (p. 10)

Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data. *arXiv:1909.09148 [cs, stat]*, November 2019b. URL http://arxiv.org/abs/1909.09148. (p. 8)

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, September 2018. URL https://openreview.net/forum?id=HJz6tiCqYm. (p. 11)

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *arXiv:1912.02781 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1912.02781. (p. 8, 11)

Alex Hernández-García and Peter König. Further Advantages of Data Augmentation on Convolutional Neural Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Lecture Notes in Computer Science, pages 95–103, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01418-6. doi: 10.1007/978-3-030-01418-6_10. (p. 8)

Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment Your Batch: Improving Generalization Through Instance Repetition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8126–8135, June 2020. doi: 10.1109/CVPR42600.2020.00815. (p. 4, 8)

W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. 2020. (p. 4, 8, 10)

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017. (p. 1, 4, 8)

Sanyam Kapoor, Wesley J Maddox, Pavel Izmailov, and Andrew Gordon Wilson. On uncertainty, tempering, and data augmentation in bayesian classification. *arXiv preprint arXiv:2203.16481*, 2022. (p. 8)

Jaehyung Kim, Dongyeop Kang, Sungsoo Ahn, and Jinwoo Shin. What Makes Better Augmentation Strategies? Augment Difficult but Not too Different. In *International Conference on Learning Representations*, September 2021. URL https://openreview.net/forum?id=Ucx3DQbC9GH. (p. 8)

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. (p. 2, 11)

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a. (p. 1, 8)

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

Daniel LeJeune, Randall Balestriero, Hamid Javadi, and Richard G. Baraniuk. Implicit Rugosity Regularization via Data Augmentation. *arXiv:1905.11639 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1905.11639. (p. 8)

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6391–6401, 2018. (p. 10)

Tianyi Liu, Yan Li, Song Wei, Enlu Zhou, and Tuo Zhao. Noisy gradient descent converges to flat minima for non-convex matrix factorization. In *International Conference on Artificial Intelligence and Statistics*, pages 1891–1899. PMLR, 2021. (p. 1, 4, 8)

Antonia Marcu and Adam Prügel-Bennett. On the Effects of Data Distortion on Model Analysis and Training. *arXiv:2110.13968 [cs]*, October 2021. URL http://arxiv.org/abs/2110.13968. (p. 8)

Samuel G. Müller and Frank Hutter. TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation. *arXiv:2103.10158 [cs]*, August 2021. URL http://arxiv.org/abs/2103.10158. (p. 8, 11, 12)

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017. (p. 8)

Shashank Rajput, Zhili Feng, Zachary Charles, Po-Ling Loh, and Dimitris Papailiopoulos. Does Data Augmentation Lead to Positive Margin? In *Proceedings of the 36th International Conference on Machine Learning*, pages 5321–5330. PMLR, May 2019. URL https://proceedings.mlr.press/v97/rajput19a.html. (p. 8)

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better Aggregation in Test-Time Augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1214–1223, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Shanmugam_Better_Aggregation_in_Test-Time_Augmentation_ICCV_2021_paper.html. (p. 3)

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. (p. 8)

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, September 2014. URL http://arxiv.org/abs/1409.1556. (p. 10)

Luke Taylor and Geoff Nitschke. Improving Deep Learning with Generic Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547, November 2018. doi: 10.1109/SSCI.2018.8628742. (p. 8)

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv:2012.12877 [cs]*, January 2021. URL http://arxiv.org/abs/2012.12877. (p. 8)

Asher Trockman and J. Zico Kolter. Patches Are All You Need? *arXiv.2201.09792*, January 2022. doi: 10.48550/arXiv.2201.09792. (p. 10)

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019. (p. 1, 3)

Ross Wightman, Hugo Touvron, and Hervé Jégou. ResNet strikes back: An improved training procedure in timm. *arXiv:2110.00476 [cs]*, October 2021. URL http://arxiv.org/abs/2110.00476. (p. 8)

Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. (arXiv:2205.01491), May 2022. doi: 10.48550/arXiv.2205.01491. (p. 8)

Larry Yaeger, Richard Lyon, and Brandyn Webb. Effective training of a neural network character classifier for word recognition. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, pages 807–813, Cambridge, MA, USA, December 1996. MIT Press. (p. 8)

Sicheng Zhu, Bang An, and Furong Huang. Understanding the Generalization Benefit of Model Invariance from a Data Perspective. In *Advances in Neural Information Processing Systems*, volume 34, pages 4328–4341. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2287c6b8641dd2d21ab050eb9ff795f3-Abstract.html. (p. 8)

A. Background and Related Work

Data Augmentations in Computer Vision. Data augmentations have been a staple of deep learning, used to deform handwritten digits as early as Yaeger et al. (1996); LeCun et al. (1998a) or to improve oversampling on class-imbalanced datasets (Chawla et al., 2002). These early works hypothesize that data augmentations are necessary to prevent overfitting when training neural networks since they typically contain many more parameters than training data points (LeCun et al., 1998a).

We restrict our study to augmentations which act on a single sample and do not modify labels. Namely, we study augmentations which can be written as $(\mathcal{T}(\mathbf{x}), y)$, where $(\mathbf{x}, y)$ denotes an input-label pair, and $\mathcal{T} \sim T$ is a random transformation sampled from a distribution of such transformations. For a broad and thorough discussion on image augmentations, their categorization, and applications to computer vision, see (Shorten and Khoshgoftaar, 2019; Xu et al., 2022). We consider basic geometric (random crops, flips, perspective) and photometric (jitter, blur, contrast) transformations, and common augmentation policies, such as AutoAug (Cubuk et al., 2019a), RandAug (Cubuk et al., 2019b), AugMix (Hendrycks et al., 2020) and TrivialAug (Müller and Hutter, 2021) which combine basic augmentations.

Understanding the Role of Augmentation and Invariance. A number of works, such as Hernández-García and König (2018), propose that data augmentations (DA) induce implicit regularization. Empirical evaluations describe useful augmentations as "label preserving", namely they do not significantly change the conditional probability over labels (Taylor and Nitschke, 2018). Gontijo-Lopes et al. (2020b;a) investigate empirical notions of consistency and diversity. They measure *consistency* (referred to as affinity) evaluating models trained without augmentation on augmented validation accuracy yielded. They also measure *diversity* as the ratio of training loss of a model trained with augmentations and a model trained without them and conclude that strong data augmentations should be both consistent and diverse. A trade-off between diversity and consistency is also seen in NLP applications (Kim et al., 2021). In contrast to Gontijo-Lopes et al. (2020b), Marcu and Prügel-Bennett (2021) find that the value of data augmentations cannot be measured by how much they deform the data distribution. Other work proposes to learn invariances parameterized as augmentations from the data (Benton et al., 2020), investigates the number of samples required to learn an invariance (Balestriero et al., 2022b), uncovers the tendency of augmentations to sacrifice performance on some classes in exchange for gains on others (Balestriero et al., 2022a), or argues that data augmentations cause models to misrepresent uncertainty (Kapoor et al., 2022).

Theoretical investigations in Chen et al. (2020) formalize data augmentations as label-preserving group actions and discuss an inherent *invariance-variance* trade-off. Variance regularization also arises when modeling augmentations for kernel classifiers (Dao et al., 2019). For a binary classifier with finite VC dimension, the bound on expected risk can be reduced through additional data generated via augmentations until *inconsistency* between augmented and real data distributions overwhelms would-be gains (He et al., 2019b). The regularizing effect of data augmentations is investigated in LeJeune et al. (2019) who propose a model under which continuous augmentations increase the smoothness of neural network decision boundaries. Rajput et al. (2019) similarly find that linear classifiers trained with sufficient augmentations can approximate the maximum margin solution. Hanin and Sun (2021) relate data augmentations to stochastic optimization. A different angle towards understanding invariances through data augmentations is presented in Zhu et al. (2021), where the effect of DA in increasing the theoretical sample cover of the distribution is investigated, and augmentations can reduce the amount of data required, if they "cover" the real distribution.

Stochastic Optimization and Neural Network Training. The implicit regularization of SGD is regarded as an essential component for neural network generalization (An, 1996; Neyshabur et al., 2017). Stochastic training which randomizes gradients can drive parameters into flatter minima, associated with superior generalization (Jastrzębski et al., 2017; Huang et al., 2020; Liu et al., 2021). In fact, Geiping et al. (2021) find that neural networks trained with non-stochastic full batch gradient descent require explicit flatness-seeking regularizers in order to achieve comparable test accuracy. Data augmentations provide an additional source of stochasticity during training on top of batch sampling, which we will investigate in this work.

A window into the effect of data augmentations on stochasticity is "batch augmentation" (Hoffer et al., 2020; Fong and Vedaldi, 2019), also termed "repeated augmentations" (Berman et al., 2019; Touvron et al., 2021; Wightman et al., 2021). Batch augmentation strategies average over multiple augmentations of each sample in the batch, resulting in gradients with less randomness that if a single augmentation is chosen per sample. Although this strategy is not well understood, it is employed in several modern training recipes. Fort et al. (2022) find that batch augmentation via decreasing the number of independent samples in a batch and including multiple augmentations of each, which decreases the stochasticity from augmentation while increasing the stochasticity from batch sampling, can boost accuracy. They conclude that the additional

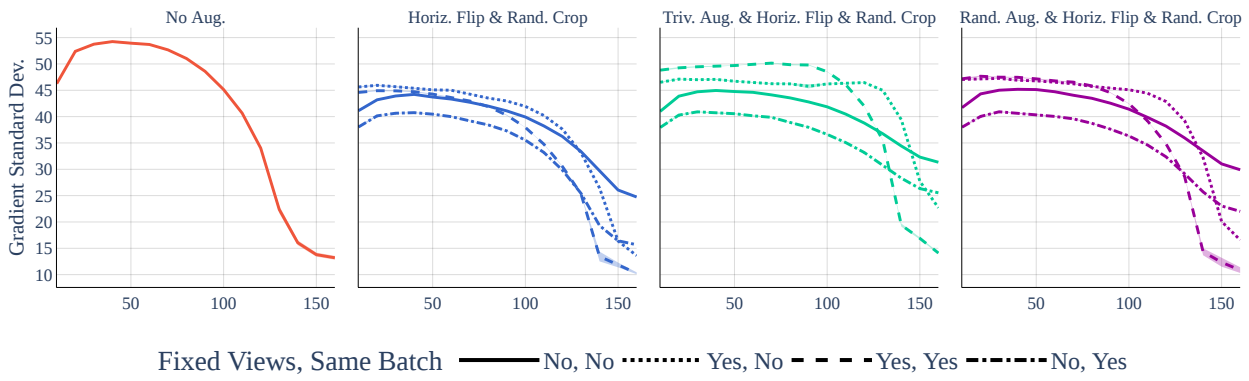


Figure 3. **Randomly applied augmentations significantly increase stochasticity late in training but decrease stochasticity early.** Standard deviation of gradient across epochs for different augmentations and different mini-batch sampling strategies. Each dot indicates the mean over 10 runs, and shaded regions represent confidence intervals of width one standard error.

stochasticity from data augmentation is harmful rather than helpful, although they do not compare to cases where training data is augmented but the additional stochasticity is eliminated entirely. We ablate away the stochasticity introduced by DA entirely, and we instead find that this stochasticity has the positive benefit of discovering flatter minima.

We fuse together the above three topics and explore the role of data augmentations play in learning invariance and in increasing stochasticity during late stages of training. In doing so, we fill in several gaps in the literature discussed in this section. Unlike other works which measure the effectiveness of data augmentations in terms of accuracy boosts, we compare the benefits of augmentations to those achieved by instead collecting more data. While other works have studied the role of data augmentations in learning invariance, we find that even invariances which have no relationship to invariances in the training data distribution are still effective. Finally, we develop an understanding of batch augmentation by showing that stochastically applied augmentations increase gradient noise during training, leading to qualitatively distinct minima.

B. Experimental Setup

For all sections if not otherwise mentioned, we run the following protocol. We train the model (in the main body a ResNet-18), with stochastic gradient descent for 60000 steps with a batch size of 128. This corresponds to 160 epochs for a dataset of size 48000. For experiments with $8\times$ or larger enlarged datasets, we set a minimum number of 40 epochs, e.g. corresponding to 120000 steps for the $8\times$ experiments, but otherwise do not modify the number of gradient steps when increasing or decreasing the dataset size. We linearly warm up the learning rate for the first 2000 steps (about 5 epochs) up to peak rate of 0.2 and then decay to zero by a half-cycle of cosine annealing. For all experiments, we include a standard weight decay of $5e-4$ and train with Nesterov momentum of 0.9. The data is shuffled randomly after every epoch and we record validation accuracy every 10000 steps. All training runs are non-deterministic based on stochasticity due to random shuffling and `cudnn` non-determinism. We run at least five trials for each experiment in the main body and three trials for each in the supplementary material. In each plot, the standard deviation is shaded. For five trials this corresponds close to a 97.5% confidence interval.

We use CIFAR-10 in its default configuration. For CINIC-10, we clean and resample the train and test sets. We first remove all CIFAR-10 train and test images from the dataset, we then further remove all exactly duplicated images and missing images, merge all remaining images and sample a new validation set of 10000 images. We provide code to replicate the creation of this cleaned dataset with the supplementary material. Overall we recover a new training set of size 193523. For CIFAR-10-C experiments in the supp. material, we report average accuracy over all transformations in CIFAR-10-C with a severity of 3. For all experiments where we consider only a subset of the existing data (e.g. each experiment with less than 193523 samples for CINIC-10), we sample a new subset of the training set for each experiment separately, to rule out confounding effects of good or bad splits of the training data, especially for smaller subset sizes.

For experiments in the main body where data augmentations are randomly sampled a finite number of times, we store all augmentations in a database (`lmdb`) that is recreated in each run. As a result, each experiment contains a fixed set of finite views of each original datapoint, but these views are randomized across experiments. Due to random shuffling, samples

from this enlarged dataset are drawn randomly and multiple views of the image are only guaranteed to occur in the same batch in the batch augmentation experiments in Section 5.

To create the table of exchange rates, we first compute the mean validation accuracy CINIC-10 for each experiment. We then train the reference models for CINIC-10 subset sizes of $1000, 2000, 3000, 6000, 12000, 24000, 48000, 96000, 128000, 144000, 168000, 180000, 192000$. To cross-reference the average validation accuracy of these reference models with our data augmentation experiments, we assert that validation accuracies are monotonically increasing as subset sizes increase and fit a linear spline f_{ref} for interpolation. We then compute the exchange ratios of Table 1 via $f_{\text{ref}}^{-1}(x)/b$, for the base dataset size b which is 48000 in Table 1 and input mean validation accuracy x for each experiment. For values outside the interval spanned by the minimal and maximal validation accuracies of the reference data, we reuse the power laws of the form $f_p(x) = ax^{-c} + b$ described in Sec. 3.3 and again compute $f_p^{-1}(x)/b$. We mark these extrapolated values by a $*$ in the table.

To estimate parameters a, b, c for $f_p(x) = ax^{-c} + b$ in the exchange rate table and Sec. 3.3 we use a non-linear least-squares algorithm, initialized from starting parameters that describe the curve for no augmentations. For this we use the Levenberg-Marquardt implementation of `MINPACK`, as wrapped in `scipy`.

Measuring stochasticity. We measure stochasticity during training as follows: We train a model on a given training set and augmentation strategy, and we freeze the model every 10 epochs to estimate the standard deviation (formally the norm of parameter-wise standard deviations) of its gradients over randomly sampled batches comprising 128 base images, the same batch size used during training. That is, we measure the square root of the average squared distance between a randomly sampled batch gradient and the mean gradient. We adopt a filter-normalized distance function (Li et al., 2018; Huang et al., 2020) to account for invariances in neural networks whereby shrinking the parameters in convolutional filters may not effect the network's output but may make the model more sensitive to parameter perturbations of a fixed size.

Measuring flatness. We adopt the flatness measurements from Huang et al. (2020) as these measurements are non-local, do not require Hessian computations which are dubious for non-smooth ReLU networks, and they are consistent with our filter-normalized gradient standard deviation measurements. Specifically, we measure the average filter-normalized distance in random directions from the trained model parameters before we reach a loss function value of 1.0, where loss is evaluated on the non-augmented dataset. Under this metric, larger values correspond to flatter minima where parameters can be perturbed further without greatly increasing loss. We use the same ResNet-18 models trained in the stochasticity experiments above with the same exact augmentation setups.

The ResNet-18 model employed in the model is a modern variant (He et al., 2015; 2019a) and contains the usual CIFAR-10 stem consisting of a single 3×3 convolutional layer without pooling, instead of the ImageNet stem (of two convolutional layers with stride and max-pooling). For experiments in the supplementary material, we further consider a ResNet-8 (i.e. three stages and a single block per stage) (He et al., 2015), a VGG-11 (Simonyan and Zisserman, 2014) with batch normalization, and a ConvMixer architecture (Trockman and Kolter, 2022) of depth 8 with hidden dimension 128 and spatial kernel size of 7.

We implement and run all experiments in `PyTorch` and make use of `torchvision` implementations for a range of data augmentations investigated in this work. We provide code to replicate all experiments with the supplementary material.

B.1. Hyperparameters for Augmentations

For each augmentation we broadly follow established defaults. For completion we record these, and additional details here.

Horiz. Flips: A data point is flipped horizontally with probability 0.5.

Det. Horiz. Flips: Deterministic horizontal flips. For $1\times$, this corresponds to flipping every data point. $2\times$ corresponds to both flips being contained in the dataset.

Vert. Flips: A data point is flipped vertically with probability 0.5.

Det. Vert. Flips: Deterministic vertical flips. For $1\times$, this corresponds to flipping every data point. $2\times$ corresponds to both flips being contained in the dataset.

Random Crops: The image is padded by with zero-padding by 4 pixels in each direction and then a image of size 32×32 is cropped (This is classical random cropping for CIFAR-10).

Flips&Crops: Both random crops and horizontal flips are employed, as described above.

Perspectives: Performs a random perspective transform with probability 0.5 with bilinear resampling.

Jitter: Color jitter, randomly transforming contrast, hue and brightness of the image. For each distortion, sample a new scale uniformly from $[0.5, 1.5]$.

Blur: Blurs the image with a Gaussian blur with $\sigma = 3$.

AutoAug: Employ the augmentation policy of Cubuk et al. (2019a), with the CIFAR-10 policy.

AugMix: The augmentation policy of Hendrycks et al. (2020).

RandAug: The augmentation policy of Cubuk et al. (2019b), again with the CIFAR-10 policy.

TrivialAug: The augmentation policy of Müller and Hutter (2021) in its "wide" configuration.

AutoAug&Flips&Crops: The AutoAug policy followed by random horizontal flips and random crops as described above.

AugMix&Flips&Crops: The Augmix policy followed by random horizontal flips and random crops as described above.

RandAug&Flips&Crops: The RandAug policy followed by random horizontal flips and random crops as described above.

TrivialAug&Flips&Crops: The TrivialAug policy followed by random horizontal flips and random crops as described above.

B.2. Data Licensing

We investigate, MNIST (LeCun et al., 1998b), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), EMNIST (Cohen et al., 2017), CIFAR10-C (Hendrycks and Dietterich, 2018) and CINIC-10 (Darlow et al., 2018) and refer to these publications for additional details. We remove duplicates and missing data from CINIC-10 as described above.

B.3. Computational Setup and Costs

We use an academic cluster with `NVIDIA RTXA4000` cards and `NVIDIA GTX2080ti` cards. Each job is scheduled on a single GPU and the default setting of 60000 gradient steps takes roughly an hour to train and evaluate. Including all preliminary experiments we estimate a total usage of about 400 GPU days for this project. To replicate all experiments in the main body without repeated trials, we estimate a requirement of about 15 GPU days.

C. Broader Impact

We foresee no direct negative societal consequences from this work. We do think that data augmentations are a beneficial tool, especially in applications with only limited data, or where data curation is expensive. We argue that knowing how to exchange a smaller (but verified and curated) dataset for a larger dataset that is not augmented, but also due to its size less curated, is helpful to the community.

D. Additional Results

Under our experimental setup, we can contrast the accuracy of augmented datasets in Figure 4 with reference models trained on larger unaugmented datasets from the true data distribution (black horizontal bars). We see that, for example, TrivialAug&Flips&Crops can generate enlarged datasets that match the performance of reference models trained on the entire, unaugmented 192000 sample dataset. We formalize this notion in Table 2. This table contains scaling factors that quantify how much the dataset size effectively increases when training with augmentations. We compute this quantity by matching each augmented model with the reference model that achieves the same val. accuracy and computing the ratio of reference dataset size to base size. We include additional information about augmentations and hyperparameters for each experiment in the appendix.

We show these exchange rates again for both CINIC-10 and CIFAR-10 val. data. At a glance, this table reveals consistency (as measured e.g. in Gontijo-Lopes et al. (2020b)) in the $1\times$ row, then shows diversity of each augmentation in the trend in each row, and examines robustness by cross-reference to the CIFAR-10 evaluations. This reference to out-of-domain data is especially relevant to practical use, and we find, for example for AugMix(Hendrycks et al., 2020) that behavior

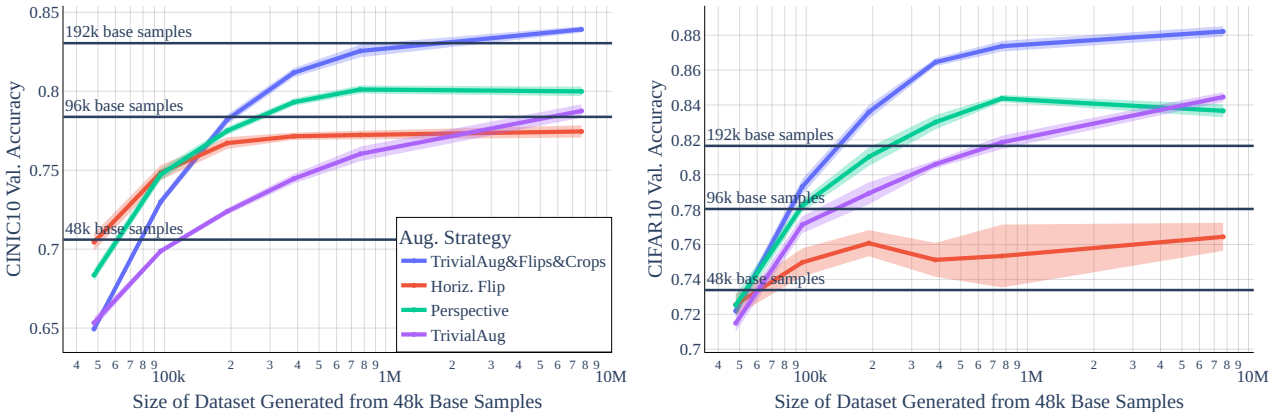


Figure 4. Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. ResNet-18 models are trained on fixed datasets generated via augmentation from 48000 base samples from the CINIC-10 train set and evaluated on the CINIC-10 val. set (**left**) and the CIFAR-10 val. set (**right**), std. error over 5 runs shaded. The accuracy of reference models trained without augmentations is marked with horizontal lines. Augmentations such as horizontal flips that are consistent with the data always improve on the baseline but saturate quickly. Diverse augmentation policies such as TrivialAug (Müller and Hutter, 2021) are inconsistent but ultimately stronger when training on larger generated datasets. These trends persist when evaluating on CIFAR-10, but augmentations are much more beneficial even under only small distribution shifts.

on in-domain data is not indicative of robust performance. AugMix, which was designed for robustness, does not improve in performance on in-domain data as more views are sampled. However, for out-of-domain data AugMix views do improve performance, and combined with flips and crops actually generate the largest amount of effective data. We also include "negative" augmentations, such as vertical flips and blur that are especially inconsistent. In the case of vertical flips, an augmentation strategy which is neither diverse nor consistent, applying the augmentation enough still yields performance benefits and tangible extra data, indicating that diversity and inconsistency alone paint an incomplete picture of the benefits of augmentation. We further probe the idea that "out-of-domain" augmentations can improve performance later on.

We include additional material for section 3 in a series of figures and tables. Table 3 is an extended version of table 1 in the main body, including repetitions up to 32× and ablating the number of steps. Behavior is consistent over additional repetitions, so we chose not to include these additional rows in the main body. Table 4 and Table 5 are then variants of this table where validation accuracy is evaluated on CIFAR-10 and CIFAR-10-C, respectively. CIFAR-10-C is a significant distribution shift that cannot be mitigated by additional CINIC-10 data, only training on, e.g. blurred samples, provides robustness to this distortion. We further find that training with horizontal flips in our experimental setup is quickly disadvantageous.

E. Additional Datasets and Models

We further verify that the findings discussed in the main body are not limited to the choice of dataset and model therein. We repeat Fig.1 and Table 1 for a range of models. We include a tiny ResNet-8 in Figure 7 and Table 6, a VGG-11 in Figure 8 and Table 7 and a ConvMixer (as representative of modern ConvNet/Transformer variations) in Figure 9 and Table 8.

We then further repeat these experiments with models trained on the MNIST training set in Figure 10 and Table 9, CIFAR-100 in Figure 11 and Table 10, as well as EMNIST in Figure 12 and Table 11.

We further include repeated experiments for Sec. 5 on CIFAR-100 in Figure 13 and Table 12.

Table 2. **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples. Each entry is the factor by which the dataset size is effectively multiplied when replacing base samples with the augmented views, e.g. training with 8 copies of each original datapoint generated through random crops reaches the same validation accuracy as training on a $2.54\times$ larger dataset of $2.54 \times 48000 = 121920$ non-augmented samples. We measure the exchange rate w.r.t. accuracy on the in-domain CINIC-10 val. set and also the slightly out-of-domain CIFAR-10 val. set. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws. These exchange rates are a direct measure of the *extra data* provided by each augmentation, and the trends in each row characterize the impact of augmentation consistency, diversity and robustness.

	CINIC-10 (in-domain)					**CIFAR-10** (minor domain shift)				
Augmentation	1x	2x	4x	8x	rand	1x	2x	4x	8x	rand
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.99	1.55	1.79	1.84	1.88	0.95	1.34	1.58	1.37	1.62
Det. Horiz. Flips	0.96	1.89	-	-	-	0.95	1.46	-	-	-
Vert. Flips	0.68	0.94	1.09	1.23	1.24	0.47	0.56	0.62	0.64	0.68
Det. Vert. Flips	0.05	1.30	-	-	-	0.02*	0.71	-	-	-
Random Crops	0.98	1.90	2.36	2.54	2.60	0.94	1.82	1.93	1.91	1.91
Flips&Crops	0.99	1.93	2.94	3.72	3.82	0.96	1.78	2.15	2.58	2.03
Perspectives	0.88	1.53	1.89	2.29	2.52	0.95	2.06	3.29	4.02*	4.38*
Jitter	0.91	0.91	0.90	0.87	0.93	0.96	0.99	1.18	0.95	1.07
Blur	0.75	0.76	0.75	0.70	0.76	1.46	1.42	1.39	1.23	1.41
AutoAug	0.76	0.95	1.01	1.18	1.64	0.99	1.57	2.06	2.37	4.00*
AugMix	0.87	0.96	0.99	0.99	1.14	1.78	2.35	2.60	3.11	3.32
RandAug	0.89	1.50	1.90	2.19	2.52	1.11	2.19	3.45	4.00*	4.26*
TrivialAug	0.71	0.96	1.23	1.50	2.16	0.89	1.78	2.29	3.12	4.77*
AutoAug&Flips&Crops	0.75	1.43	2.22	3.21	4.00*	0.96	2.53	4.46*	6.30*	7.14*
AugMix&Flips&Crops	0.86	1.60	2.48	3.04	3.72	1.81	4.45*	6.88*	8.55*	9.09*
RandAug&Flips&Crops	0.84	1.71	2.65	3.78	4.00*	0.96	2.32	4.00*	5.10*	5.14*
TrivialAug&Flips&Crops	0.70	1.31	1.98	2.86	4.00*	0.93	2.43	4.30*	6.10*	7.74*

Table 3. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples. We measure the exchange rate w.r.t. accuracy on the in-domain **CINIC-10 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws. For a select augmentations we also include experiments with 240000 steps, i.e. 640 passes through the data to verify the utility of our chosen schedule of 60000 steps.

	CINIC-10 (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.99	1.55	1.79	1.84	1.85	1.79	1.88	-
Det. Horiz. Flips	0.96	1.89	-	-	-	-	-	-
Vert. Flips	0.68	0.94	1.09	1.23	1.17	1.14	1.25	1.20
Det. Vert. Flips	0.05	1.30	-	-	-	-	-	-
Random Crops	0.98	1.90	2.36	2.54	2.61	2.74	2.59	2.74
Flips&Crops	0.99	1.93	2.94	3.72	4.00*	4.00*	3.79	-
Perspectives	0.88	1.53	1.89	2.29	2.54	2.68	2.50	-
Jitter	0.91	0.92	0.90	0.87	0.82	0.81	0.93	0.88
Blur	0.76	0.76	0.75	0.70	0.66	0.62	0.76	0.69
AutoAug	0.78	0.95	1.02	1.20	1.39	1.52	1.63	1.77
AugMix	0.87	0.95	0.98	1.00	1.02	1.00	1.14	1.13
RandAug	0.88	1.49	1.91	2.20	2.51	2.67	2.49	-
TrivialAug	0.72	0.96	1.23	1.50	1.70	1.87	2.12	-
AutoAug&Flips&Crops	0.75	1.43	2.22	3.21	4.00*	4.00*	4.00*	4.08*
Augmix&Flips&Crops	0.86	1.62	2.50	3.09	3.82	3.84	3.74	3.74
RandAug&Flips&Crops	0.84	1.71	2.65	3.78	4.00*	4.00*	4.00*	-
TrivialAug&Flips&Crops	0.70	1.31	1.98	2.86	3.71	4.00*	4.00*	-

Table 4. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples. We measure the exchange rate w.r.t. accuracy on the **CIFAR-10 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws.

	CIFAR-10 (slightly out-of-domain)						
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.95	1.34	1.58	1.37	1.42	1.35	1.66
Det. Horiz. Flips	0.95	1.46	-	-	-	-	-
Vert. Flips	0.47	0.56	0.62	0.64	0.64	0.66	0.68
Det. Vert. Flips	0.02*	0.71	-	-	-	-	-
Random Crops	0.94	1.82	1.93	1.91	1.75	1.92	1.91
Flips&Crops	0.96	1.78	2.15	2.58	2.26	3.05	1.94
Perspectives	0.95	2.06	3.29	4.02*	4.73*	4.96*	4.34*
Jitter	0.97	1.04	1.09	0.95	0.89	0.86	1.08
Blur	1.52	1.44	1.35	1.20	1.03	0.97	1.40
AutoAug	0.99	1.60	2.00	2.39	3.14	3.46	4.00*
AugMix	1.77	2.38	2.61	3.10	3.18	3.20	3.31
RandAug	1.15	2.29	3.42	4.00*	4.56*	5.19*	4.02*
TrivialAug	0.89	1.81	2.30	3.17	4.00*	4.02*	4.78*
AutoAug&Flips&Crops	0.96	2.53	4.46*	6.30*	6.93*	7.27*	7.18*
AugMix&Flips&Crops	1.74	4.41*	6.86*	8.66*	8.92*	9.45*	9.01*
RandAug&Flips&Crops	0.96	2.32	4.00*	5.10*	6.24*	6.80*	5.12*
TrivialAug&Flips&Crops	0.93	2.43	4.30*	6.10*	6.84*	7.34*	7.60*

Table 5. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples. We measure the exchange rate w.r.t. accuracy on the **CIFAR-10-C val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws. Note that especially values > 10 are an extensive extrapolation far outside the measured range. Values marked with ✓ are outside the range of the estimated power law, meaning that (at least according to the behavior predicted by it), no amount of additional real data with be sufficient to match the accuracy achieved with this augmentation - there is no exchange rate.

	CIFAR-10-C (out-of-domain)						
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.84	0.67	0.66	0.55	0.53	0.52	0.63
Det. Horiz. Flips	0.93	0.55	-	-	-	-	-
Vert. Flips	0.15	0.11	0.11	0.11	0.11	0.12	0.11
Det. Vert. Flips	0.02*	0.12	-	-	-	-	-
Random Crops	0.82	0.86	0.70	0.66	0.60	0.69	0.67
Flips&Crops	0.86	0.66	0.63	0.78	0.64	0.71	0.25
Perspectives	34.16*	✓	✓	✓	✓	✓	✓
Jitter	16.81*	190.76*	✓	16.99*	7.26*	3.08	163.08*
Blur	✓	✓	✓	✓	✓	✓	✓
AutoAug	✓	✓	✓	✓	✓	✓	✓
AugMix	✓	✓	✓	✓	✓	✓	✓
RandAug	✓	✓	✓	✓	✓	✓	✓
TrivialAug	✓	✓	✓	✓	✓	✓	✓
AutoAug&Flips&Crops	✓	✓	✓	✓	✓	✓	✓
AugMix&Flips&Crops	✓	✓	✓	✓	✓	✓	✓
RandAug&Flips&Crops	91.64*	✓	✓	✓	✓	✓	✓
TrivialAug&Flips&Crops	✓	✓	✓	✓	✓	✓	✓

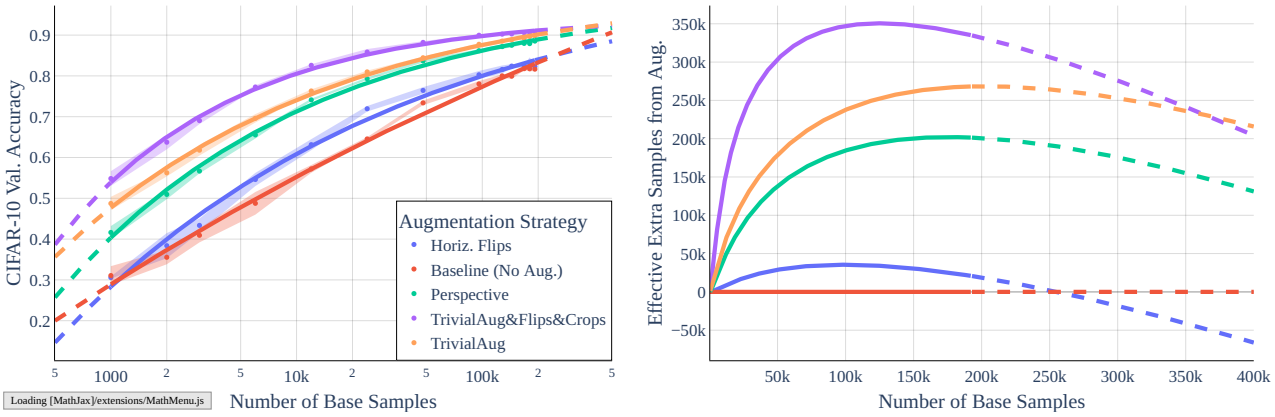


Figure 5. Variant of the **Power laws** $f(x) = ax^{-c} + b$ for select augmentations applied randomly and the gain in terms of effective extra samples, for validation accuracy measured on **CIFAR-10**. Fitted curves marked in solid colors, with extrapolated regions dashed. **Left:** Number of base samples (from CINIC-10) on the logarithmic horizontal axis compared to validation accuracy. **Right**: Number of base samples compared to effective extra data.

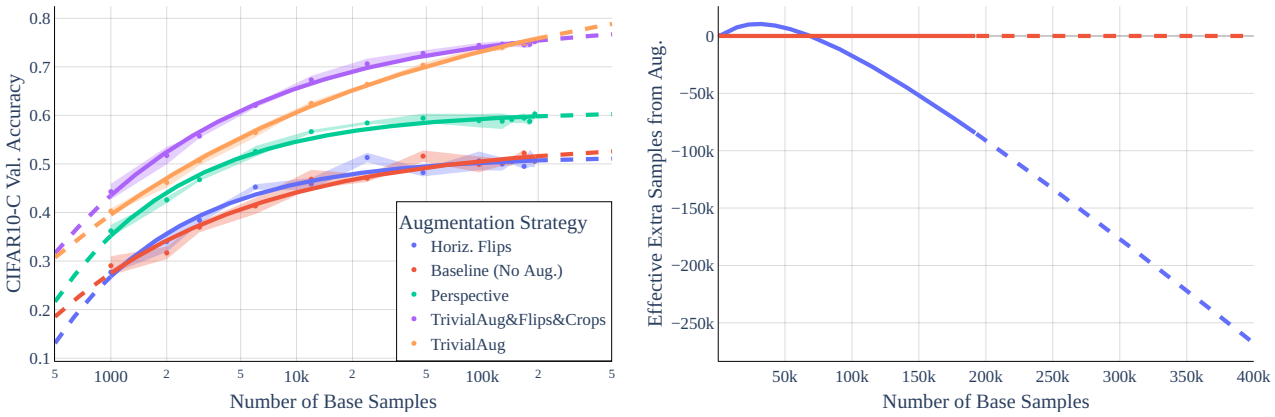


Figure 6. Variant of the **Power laws** $f(x) = ax^{-c} + b$ for select augmentations applied randomly and the gain in terms of effective extra samples, for validation accuracy measured on **CIFAR-10-C**. Fitted curves marked in solid colors, with extrapolated regions dashed. **Left:** Number of base samples (from CINIC-10) on the logarithmic horizontal axis compared to validation accuracy. **Right**: Number of base samples compared to effective extra data. Only augmentation with finite exchange rate are included in the right plot.

Table 6. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples for **ResNet-8** models. We measure the exchange rate w.r.t. accuracy on the **CINIC-10 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws.

	CINIC-10 (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	1.02	1.50	1.74	1.78	1.69	1.67	1.91	1.69
Det. Horiz. Flips	1.05	2.03	-	-	-	-	-	-
Vert. Flips	0.68	0.92	1.10	1.03	1.02	1.00	1.22	1.09
Det. Vert. Flips	0.09	1.31	-	-	-	-	-	-
Random Crops	0.98	1.82	2.40	2.44	2.51	2.62	-	2.62
Flips&Crops	0.96	1.89	2.69	2.99	3.33	4.00*	1.07	3.83
Perspectives	0.90	1.57	1.95	2.13	2.29	2.18	2.39	2.35
Jitter	1.00	1.15	1.18	1.05	1.06	1.04	1.24	1.21
Blur	0.77	0.90	0.98	0.96	0.96	0.93	1.05	0.97
AutoAug	0.93	1.32	1.64	1.64	1.69	1.68	1.91	1.80
AugMix	1.03	1.31	1.52	1.54	1.54	1.48	1.75	1.67
RandAug	0.97	1.66	2.06	2.26	2.42	2.45	2.67	2.68
TrivialAug	0.82	1.39	1.84	1.96	2.07	1.99	2.19	2.24
AutoAug&Flips&Crops	0.85	1.62	2.15	2.65	2.89	3.14	2.62	3.00
AugMix&Flips&Crops	0.92	1.75	2.44	2.79	3.08	3.45	2.82	3.22
RandAug&Flips&Crops	0.93	1.78	2.47	2.84	3.28	4.00*	2.84	3.92
TrivialAug&Flips&Crops	0.75	1.62	2.03	2.51	2.75	2.91	2.52	2.93

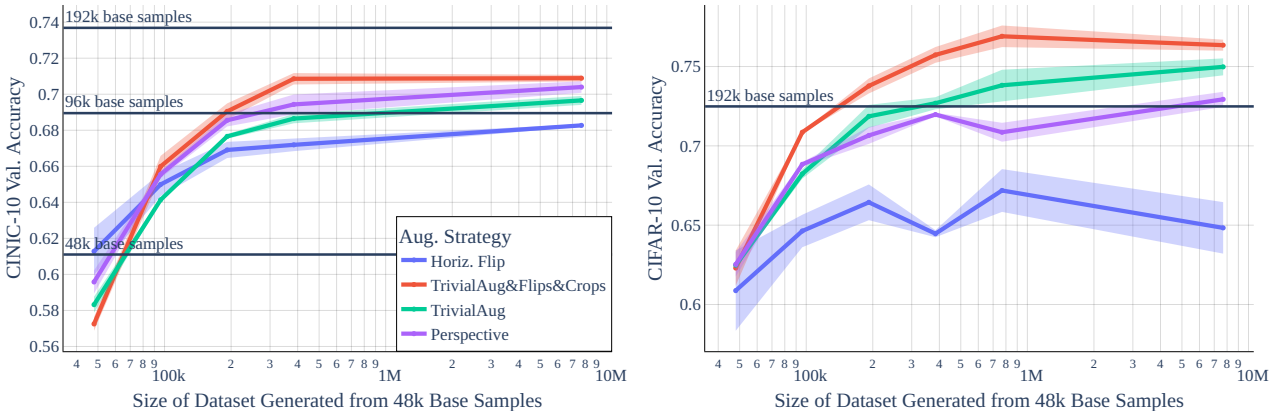


Figure 7. Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **ResNet-8** models are trained on fixed datasets generated via augmentation from 48000 base samples from the CINIC-10 train set and evaluated on the CINIC-10 val. set (**left**) and the CIFAR-10 val. set (**right**), std. error over 3 runs shaded. The accuracy of reference models trained without augmentations is marked with horizontal lines.

Table 7. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples for **VGG-11** models. We measure the exchange rate w.r.t. accuracy on the **CINIC-10 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws.

	CINIC-10 (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	1.00	1.54	1.77	1.78	1.77	1.78	1.89	1.86
Det. Horiz. Flips	0.98	1.97	-	-	-	-	-	-
Vert. Flips	0.54	0.83	0.93	0.95	0.93	0.92	1.00	0.97
Det. Vert. Flips	0.02	1.08	-	-	-	-	-	-
Random Crops	0.98	1.70	2.11	2.12	2.22	2.11	-	2.24
Flips&Crops	0.98	1.86	2.56	2.99	3.08	3.11	1.01	3.15
Perspectives	0.79	1.20	1.54	1.56	1.51	1.46	1.76	1.51
Jitter	0.89	0.86	0.86	0.84	0.79	0.81	0.86	0.82
Blur	0.60	0.59	0.58	0.54	0.56	0.51	0.62	0.55
AutoAug	0.72	0.91	1.05	1.10	1.26	1.24	1.59	1.51
AugMix	0.91	0.97	0.97	0.96	0.95	0.97	1.02	1.00
RandAug	0.83	1.37	1.79	1.92	2.00	1.98	2.14	2.19
TrivialAug	0.64	0.94	1.14	1.37	1.55	1.65	1.94	2.00
AutoAug&Flips&Crops	0.67	1.28	2.09	2.68	3.13	3.21	4.00*	4.00*
AugMix&Flips&Crops	0.83	1.55	2.31	2.79	3.06	2.81	3.22	3.12
RandAug&Flips&Crops	0.79	1.57	2.27	3.09	3.32	3.33	3.97	3.92
TrivialAug&Flips&Crops	0.54	1.13	1.83	2.46	2.93	3.35	4.00*	4.00*

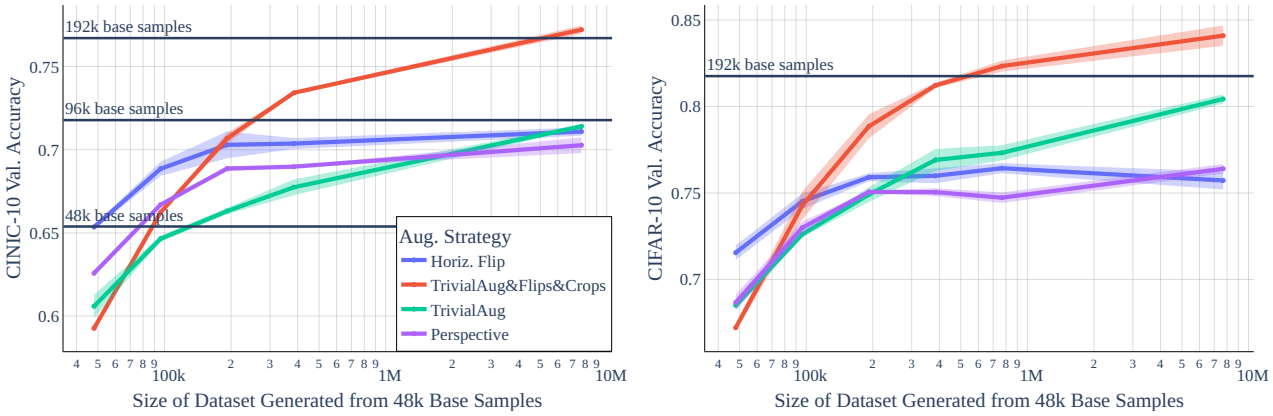


Figure 8. Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **VGG-11** models are trained on fixed datasets generated via augmentation from 48000 base samples from the CINIC-10 train set and evaluated on the CINIC-10 val. set (**left**) and the CIFAR-10 val. set (**right**), std. error over 3 runs shaded. The accuracy of reference models trained without augmentations is marked with horizontal lines.

Table 8. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the CINIC-10 training set, compared to reference models trained without augmentations on up to 192000 samples for **ConvMixer** models. We measure the exchange rate w.r.t. accuracy on the **CINIC-10 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws.

	CINIC-10 (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.95	1.37	1.39	1.59	1.50	1.69	1.41	1.66
Det. Horiz. Flips	1.04	1.50	-	-	-	-	-	-
Vert. Flips	0.37	0.45	0.45	0.48	0.55	0.50	0.48	0.49
Det. Vert. Flips	0.04	0.43	-	-	-	-	-	-
Random Crops	0.78	0.93	0.97	1.18	1.36	1.30	1.00	1.26
Flips&Crops	0.87	1.05	1.40	1.71	1.59	1.63	1.00	1.72
Perspectives	0.71	0.97	1.26	1.85	2.24	2.03	2.04	2.33
Jitter	0.78	0.91	0.90	1.04	1.11	1.34	0.87	1.26
Blur	0.34	0.46	0.51	0.49	0.56	0.61	0.54	0.65
AutoAug	0.59	0.89	1.42	1.77	1.96	2.30	1.57	2.07
AugMix	0.77	0.91	0.89	1.31	1.60	1.77	1.40	1.70
RandAug	0.80	1.28	1.71	1.76	2.05	2.59	2.21	2.28
TrivialAug	0.48	1.11	1.72	2.31	2.84	2.91	2.28	2.66
AutoAug&Flips&Crops	0.49	1.04	1.85	2.19	2.67	2.80	2.49	2.91
AugMix&Flips&Crops	0.65	1.22	1.58	1.91	2.10	2.12	1.92	2.20
RandAug&Flips&Crops	0.57	1.19	1.82	2.40	2.65	2.69	2.89	2.89
TrivialAug&Flips&Crops	0.41	1.07	1.83	2.21	2.92	3.55	2.67	3.08

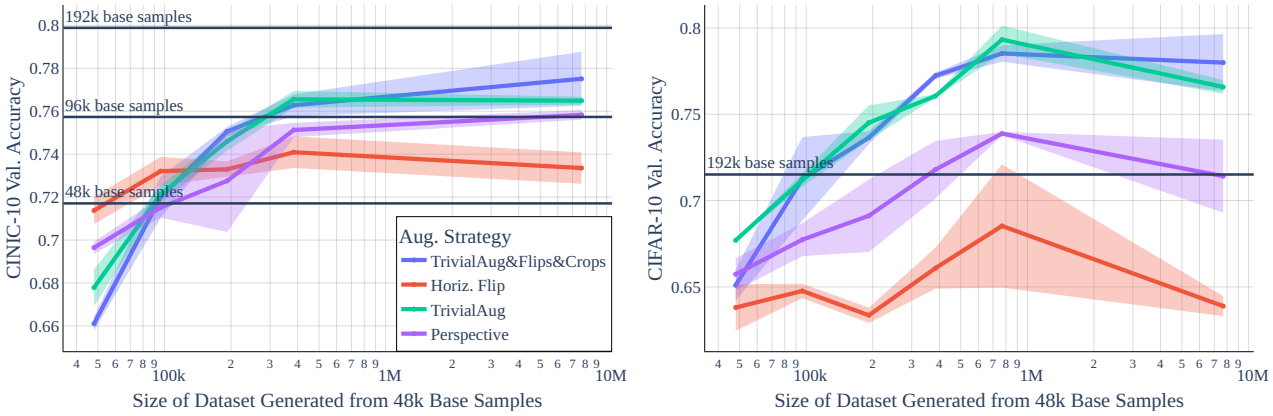


Figure 9. Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **ConvMixer** models are trained on fixed datasets generated via augmentation from 48000 base samples from the CINIC-10 train set and evaluated on the CINIC-10 val. set (**left**) and the CIFAR-10 val. set (**right**), std. error over 3 runs shaded. The accuracy of reference models trained without augmentations is marked with horizontal lines.

Table 9. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the **MNIST** training set, compared to reference models trained without augmentations on up to 60000 samples for **ResNet-18** models. We measure the exchange rate w.r.t. accuracy on the **MNIST val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws. Values marked with ✓ are outside the range of the estimated power law, meaning that (at least according to the behavior predicted by it), no amount of additional real data with be sufficient to match the accuracy achieved with this augmentation - there is no exchange rate.

	MNIST (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.24	0.24	0.34	0.27	0.26	0.29	0.40	0.25
Det. Horiz. Flips	0.02*	-	-	-	-	-	-	-
Vert. Flips	0.20	0.22	0.22	0.22	0.22	0.24	0.23	0.22
Det. Vert. Flips	0.02*	-	-	-	-	-	-	-
Random Crops	1.11	1.16	✓	1.17	✓	✓	-	37.32*
Flips&Crops	0.17	0.23	0.21	0.22	0.24	0.25	0.88	0.23
Perspectives	1.16	1.14	7.84*	1.17	1.17	✓	1.15	37.32*
Jitter	1.08	0.81	1.17	1.08	1.11	1.15	0.81	1.10
Blur	1.08	1.10	1.12	0.77	1.15	7.84*	1.17	1.14
AutoAug	0.72	1.16	1.08	1.16	4.38*	7.84*	7.84*	✓
AugMix	1.11	1.14	1.16	1.16	1.17	1.17	1.12	1.15
RandAug	1.10	1.15	4.38*	✓	✓	✓	1.16	✓
TrivialAug	0.54	4.38*	1.12	1.12	4.38*	✓	1.16	✓
AutoAug&Flips&Crops	0.14	0.17	0.20	0.24	0.26	0.24	0.24	0.26
AugMix&Flips&Crops	0.15	0.21	0.21	0.24	0.23	0.24	0.24	0.24
RandAug&Flips&Crops	0.17	0.21	0.21	0.23	0.24	-	0.23	0.23
TrivialAug&Flips&Crops	0.12	0.15	0.16	0.21	0.22	0.23	0.21	0.24

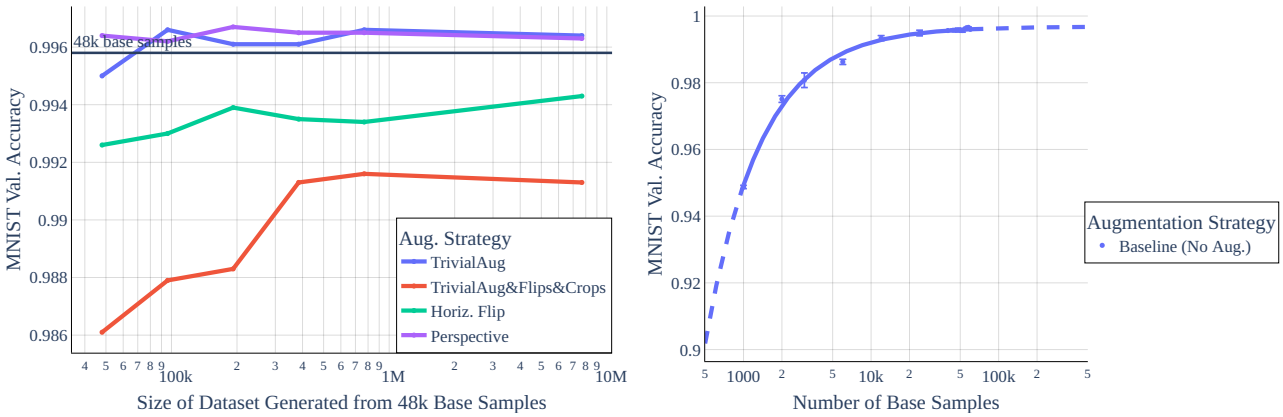


Figure 10. **Left:** Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **ResNet-18** models are trained on fixed datasets generated via augmentation from 48000 base samples from the MNIST train set and evaluated on the MNIST val. set. **Right:** Extrapolated scaling behavior of reference models for MNIST.

Table 10. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the **CIFAR-100** training set, compared to reference models trained without augmentations on up to 50000 samples for **ResNet-18** models. We measure the exchange rate w.r.t. accuracy on the **CIFAR-100 val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws.

	CIFAR-100 (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.95	1.39*	1.57*	1.59*	1.56*	1.52*	1.61*	1.55*
Det. Horiz. Flips	0.89	1.64*	-	-	-	-	-	-
Vert. Flips	0.62	0.94	1.13*	1.14*	1.11*	1.02	1.18*	1.14*
Det. Vert. Flips	0.18	1.19*	-	-	-	-	-	-
Random Crops	0.90	1.58*	1.88*	2.00*	1.99*	1.99*	-	2.04*
Flips&Crops	0.87	1.60*	2.08*	2.30*	2.35*	2.28*	0.99	2.35*
Perspectives	0.78	1.33*	1.66*	1.90*	1.97*	1.94*	1.87*	1.96*
Jitter	0.88	0.90	0.94	0.88	0.82	0.80	0.90	0.82
Blur	0.77	0.75	0.74	0.70	0.67	0.67	0.73	0.69
AutoAug	0.71	0.92	0.99	1.01	1.10*	1.14*	1.40*	1.37*
AugMix	0.86	0.97	1.00	1.02	1.03	1.02	1.15*	1.14*
RandAug	0.79	1.30*	1.58*	1.80*	1.86*	1.88*	1.81*	1.87*
TrivialAug	0.59	0.91	1.12*	1.21*	1.32*	1.48*	1.78*	1.86*
AutoAug&Flips&Crops	0.60	1.22*	1.73*	2.04*	2.20*	2.23*	2.29*	2.37*
AugMix&Flips&Crops	0.75	1.47*	1.94*	2.23*	2.26*	2.30*	2.15*	2.20*
RandAug&Flips&Crops	0.67	1.39*	1.94*	2.23*	2.26*	2.31*	2.24*	2.24*
TrivialAug&Flips&Crops	0.51	1.03	1.58*	1.92*	2.10*	2.22*	2.37*	2.55*

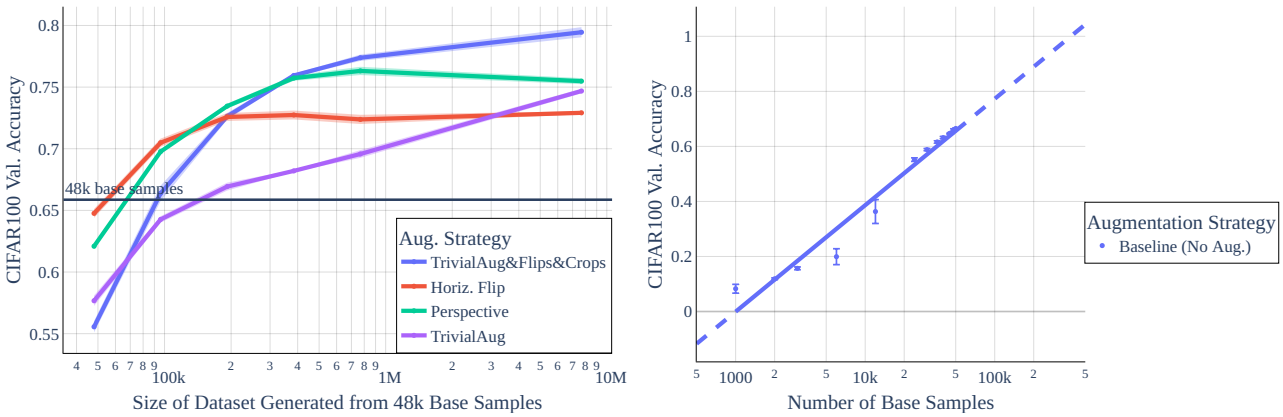


Figure 11. **Left:** Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **ResNet18** models are trained on fixed datasets generated via augmentation from 48000 base samples from the CIFAR-100 train set and evaluated on the CIFAR-100 val. set. **Right:** Extrapolated scaling behavior of reference models for CIFAR-100.

Table 11. Extended table of **Exchange rates** for augmentations applied to 48000 base samples from the **EMNIST** training set, compared to reference models trained without augmentations on up to 124800 samples for **ResNet-18** models. We measure the exchange rate w.r.t. accuracy on the **EMNIST val. set**. Values marked with $*$ fall outside the range of reference datasets and are extrapolated using power laws. Values marked with ✓ are outside the range of the estimated power law, meaning that (at least according to the behavior predicted by it), no amount of additional real data with be sufficient to match the accuracy achieved with this augmentation - there is no exchange rate.

	EMNIST (in-domain)							
Augmentation	1x	2x	4x	8x	16x	32x	rand (160)	rand (640)
-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Horiz. Flips	0.12	0.15	0.15	0.16	0.15	0.15	0.16	0.16
Det. Horiz. Flips	0.02*	-	-	-	-	-	-	-
Vert. Flips	0.12	0.16	0.17	0.18	0.16	0.17	0.17	0.17
Det. Vert. Flips	0.02*	-	-	-	-	-	-	-
Random Crops	0.61	0.93	1.90	2.10	5.90*	✓	-	✓
Flips&Crops	0.10	0.12	0.15	0.18	0.19	0.23	1.12	0.20
Perspectives	0.95	2.04	1.99	2.11	2.06	✓	✓	2.18
Jitter	0.72	1.45	0.90	0.91	1.36	0.98	1.03	1.24
Blur	0.82	0.81	0.75	0.88	0.94	0.93	0.92	1.16
AutoAug	1.24	1.26	0.99	0.92	2.12	2.07	13.79*	2.03
AugMix	1.60	0.90	1.04	2.10	2.04	2.02	1.56	1.99
RandAug	1.24	2.14	2.05	2.07	✓	✓	✓	✓
TrivialAug	0.86	0.79	0.96	1.91	1.57	1.84	✓	✓
AutoAug&Flips&Crops	0.10	0.12	0.12	0.14	0.21	0.21	0.30	0.25
AugMix&Flips&Crops	0.10	0.12	0.12	0.17	0.17	0.19	0.21	0.16
RandAug&Flips&Crops	0.10	0.13	0.15	0.18	0.20	0.20	0.24	0.21
TrivialAug&Flips&Crops	0.06	0.10	0.11	0.15	0.20	0.22	0.28	0.23

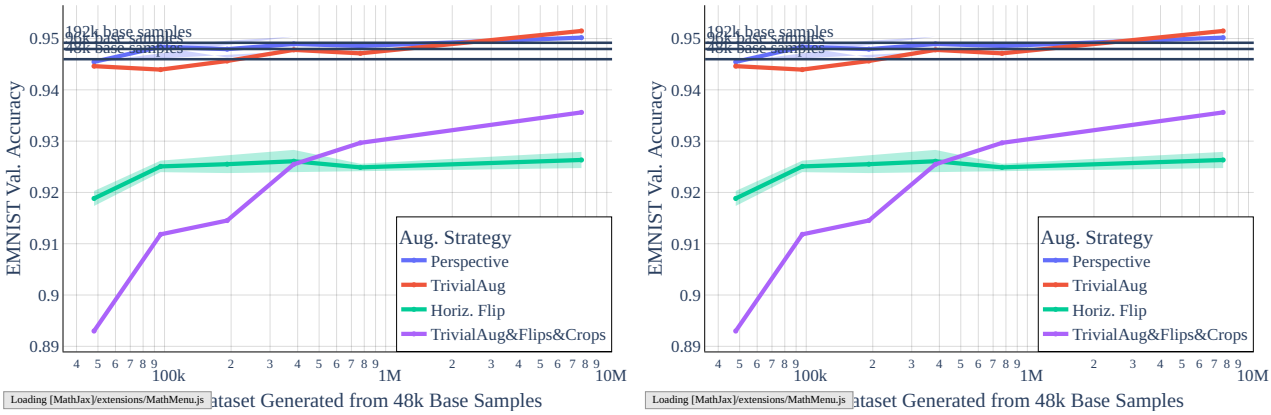


Figure 12. **Left:** Validation accuracy versus dataset size as **larger datasets are generated from a fixed number of base samples** and selected data augmentations. **ResNet-18** models are trained on fixed datasets generated via augmentation from 48000 base samples from the EMNIST train set and evaluated on the EMNIST val. set. **Right:** Extrapolated scaling behavior of reference models for EMNIST.

Table 12. Gradient standard deviation across batches at the end of training and flatness measurements for Mobilenet V2 models trained on CIFAR-100 with various augmentations and strategies for sampling augmented views. Averaged over 3 runs

Augmentation	**Fixed Views**	**Same Batch**	**Grad. Std.**	**Flatness**
No Augmentation	-	-	46.39	7.78
Horiz. Flip & Rand. Crop	No	No	46.37	7.79
	Yes	No	42.62	3.41

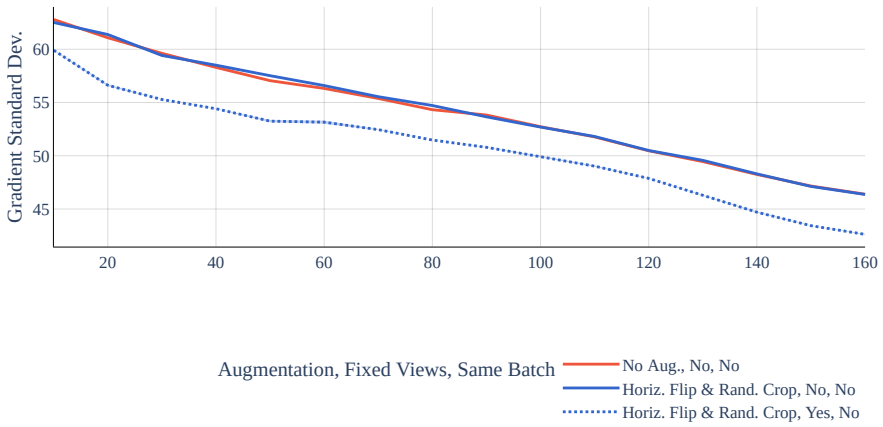


Figure 13. Standard deviation of gradient across epochs for different augmentations and different mini-batch sampling strategies. Each dot indicates the mean over 3 runs, and shaded regions represent confidence intervals of width one standard error.