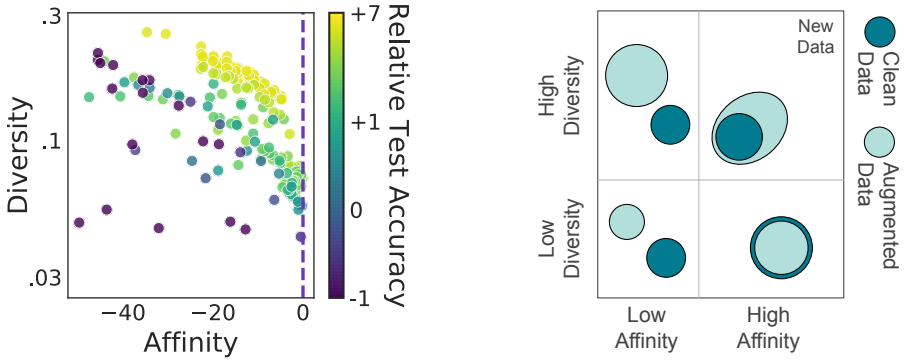
TRADEOFFS IN DATA AUGMENTATION: AN EMPIRICAL STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Though data augmentation has become a standard component of deep neural network training, the underlying mechanism behind the effectiveness of these techniques remains poorly understood. In practice, augmentation policies are often chosen using heuristics of distribution shift or augmentation diversity. Inspired by these, we conduct an empirical study to quantify how data augmentation improves model generalization. We introduce two interpretable and easy-to-compute measures: Affinity and Diversity. We find that augmentation performance is predicted not by either of these alone but by jointly optimizing the two.



(a) Affinity vs Diversity (b) Model's View of Data

Figure 1: **Affinity and Diversity parameterize the performance of a model trained with augmentation**. (a) CIFAR-10: Color shows the final test accuracy above the baseline trained without augmentation (89.7%). Each point represents a different augmentation that yields test accuracy greater than (88.7%). (b) Representation of how clean data and augmented data are related in the space of these two metrics. Higher diversity is represented by a larger bubble while distributional similarity is depicted through the overlap of bubbles. Test accuracy generally improves to the upper right in this space. Adding real new data to the training set is expected to be in the upper right corner.

1 INTRODUCTION

Models that achieve state-of-the-art in image classification often use heavy data augmentation strategies. The best techniques use various transforms applied sequentially and stochastically. Though the effectiveness of this is well-established, the mechanism through which these transformations work is not well-understood.

Since early uses of data augmentation, it has been assumed that augmentation works because it simulates realistic samples from the true data distribution: "[augmentation strategies are] reasonable since the transformed reference data is now extremely close to the original data. In this way, the amount of training data is effectively increased" (Bellegarda et al., 1992). Because of this, augmentations have often been designed with the heuristic of incurring minimal distribution shift from the training data.

This rationale does not explain why unrealistic distortions such as cutout (DeVries & Taylor, 2017), SpecAugment (Park et al., 2019), and mixup (Zhang et al., 2017) significantly improve generalization performance. Furthermore, methods do not always transfer across datasets—`Cutout`, for example, is useful on CIFAR-10 and not on ImageNet (Lopes et al., 2019). Additionally, many augmentation policies heavily modify images by stochastically applying multiple transforms to a single image. Based on this observation, some have proposed that augmentation strategies are effective because they increase the diversity of images seen by the model.

In this complex landscape, claims about diversity and distributional similarity remain unverified heuristics. Without more precise data augmentation science, finding state-of-the-art strategies requires brute force that can cost thousands of GPU hours (Cubuk et al., 2018; Zhang et al., 2019). This highlights a need to specify and measure the relationship between the original training data and the augmented dataset, as relevant to a given model's performance.

In this paper, we quantify these heuristics. Seeking to understand the mechanisms of augmentation, we focus on single transforms as a foundation. We present an empirical study of 204 different augmentations on CIFAR-10 and 225 on ImageNet, varying both broad transform families and finer transform parameters. To better understand current state of the art augmentation policies, we additionally measure 58 composite augmentations on ImageNet and three state of the art augmentations on CIFAR-10. Our contributions are:

1. We introduce Affinity and Diversity: interpretable, easy-to-compute metrics for parametrizing augmentation performance. Affinity quantifies how much an augmentation shifts the training data distribution. Diversity quantifies the complexity of the augmented data with respect to the model and learning procedure.
2. We find that performance is dependent on *both* metrics. In the Affinity-Diversity plane, the best augmentation strategies jointly optimize the two (see Fig 1).
3. We connect augmentation to other familiar forms of regularization, such as ℓ_2 and learning rate scheduling, observing common features of the dynamics: performance can be improved and training accelerated by turning off regularization at an appropriate time.
4. We find that performance is only improved when a transform increases the total number of unique training examples. The utility of these new training examples is informed by the augmentation's Affinity and Diversity.

2 Related Work

Since early uses of data augmentation in training neural networks, there has been an assumption that effective transforms for data augmentation are those that produce images from an "overlapping but different" distribution (Bengio et al., 2011; Bellegarda et al., 1992). Indeed, elastic distortions as well as distortions in the scale, position, and orientation of training images have been used on MNIST (Ciregan et al., 2012; Sato et al., 2015; Simard et al., 2003; Wan et al., 2013), while horizontal flips, random crops, and random distortions to color channels have been used on CIFAR-10 and ImageNet (Krizhevsky et al., 2012; Zagoruyko & Komodakis, 2016; Zoph et al., 2017). For object detection and image segmentation, one can also use object-centric cropping (Liu et al., 2016) or cut-and-paste new objects (Dwibedi et al., 2017; Fang et al., 2019; Ngiam et al., 2019).

In contrast, researchers have also successfully used less domain-specific transformations, such as Gaussian noise (Ford et al., 2019; Lopes et al., 2019), input dropout (Srivastava et al., 2014), erasing random patches of the training samples (DeVries & Taylor, 2017; Park et al., 2019; Zhong et al., 2017), and adversarial noise (Szegedy et al., 2013). Mixup (Zhang et al., 2017) and Sample Pairing (Inoue, 2018) are two augmentation methods that use convex combinations of training samples.

It is also possible to improve generalization by combining individual transformations. For example, reinforcement learning has been used to choose more optimal combinations of data augmentation transformations (Ratner et al., 2017; Cubuk et al., 2018). Follow-up research has lowered the computation cost of such optimization, by using population based training (Ho et al., 2019), density matching (Lim et al., 2019), adversarial policy-design that evolves throughout training (Zhang et al., 2019), or a reduced search space (Cubuk et al., 2019). Despite producing unrealistic outputs, such combinations of augmentations can be highly effective in different tasks (Berthelot et al., 2019; Tan & Le, 2019; Tan et al., 2019; Xie et al., 2019a;b).

Across these different examples, the role of distribution shift in training remains unclear. Lim et al. (2019); Hataya et al. (2019) have found augmentation policies by minimizing the distance between the distributions of augmented data and clean data. Recent work found that after training with augmented data, fine-tuning on clean training data can be beneficial (He et al., 2019), while Touvron et al. (2019) found it beneficial to fine-tune with a test-set resolution that aligns with the training-set resolution.

The true input-space distribution from which a training dataset is drawn remains elusive. To better understand the effect of distribution shift on performance, many works attempt to estimate it. Often these techniques require training secondary models, such as those based on variational methods (Goodfellow et al., 2014; Kingma & Welling, 2014; Nowozin et al., 2016; Blei et al., 2017). Others have augmented the training set by modelling the data distribution directly (Tran et al., 2017). Recent work has suggested that even unrealistic distribution modelling can be beneficial (Dai et al., 2017).

These methods try to specify the distribution separately from the model they are trying to optimize. As a result, they are insensitive to any interaction between the model and data distribution. Instead, we are interested in a measure of how much the data shifts along directions that are most relevant to the model's performance.

3 METHODS

We performed extensive experiments with various augmentations on CIFAR-10 and ImageNet. Experiments on CIFAR-10 used the WRN-28-2 model (Zagoruyko & Komodakis, 2016), trained for 78k steps with cosine learning rate decay. Results are the mean over 10 initializations and reported errors (often too small to show on figures) are the standard error on the mean. Details on the error analysis are in Sec. C.

Experiments on ImageNet used the ResNet-50 model (He et al., 2016), trained for 112.6k steps with a weight decay rate of 1e-4, and a learning rate of 0.2, which is decayed by 10 at epochs 30, 60, and 80.

Images were pre-processed by dividing each pixel value by 255 and normalizing by the data set statistics. Random crop was also applied on all ImageNet models. These pre-processed data without further augmentation are "clean data" and a model trained on it is the "clean baseline". We followed the same implementation details as Cubuk et al. (2018)[1], including for most augmentation operations. Further implementation details are in Sec. A.

Unless specified otherwise, data augmentation was applied following standard practice: each time an image is drawn, the given augmentation is applied with a given probability. We call this mode *dynamic* augmentation. Due to whatever stochasticity is in the transform itself (such as randomly selecting the location for a crop) or in the policy (such as applying a flip only with 50% probability), the augmented image could be different each time. Thus, most of the tested augmentations increase the number of possible distinct images that can be shown during training.

We also performed select experiments using *static* training. In static augmentation, the augmentation policy (one or more transforms) is applied once to the entire clean training set. Static augmentation does not change the number of unique images in the dataset.

3.1 AFFINITY: A SIMPLE METRIC FOR DISTRIBUTION SHIFT

Thus far, heuristics of distribution shift have motivated design of augmentation policies. Inspired by this focus, we introduce a simple metric to quantify how augmentation shifts data *with respect to the decision boundary of the clean baseline model*.

We start by noting that a trained model is often sensitive to the distribution of the training data. That is, model performance varies greatly between new samples from the true data distribution and samples from a shifted distribution.

Importantly, the model's sensitivity to distribution shift is not purely a function of the input data distribution, since training dynamics and the model's implicit biases affect performance. Because the goal of augmentation is improving model performance, measuring shifts with respect to the distribution captured by the model is more meaningful than measuring shifts in the distribution of the input data alone.

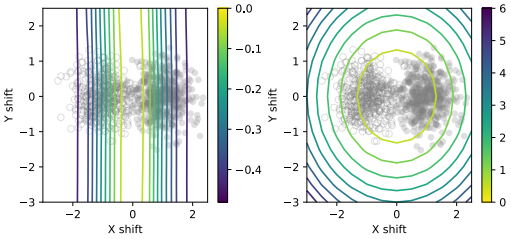
[1] Available at `bit.ly/2v2FojN`

We thus define Affinity to be the difference between the validation accuracy of a model trained on clean data and tested on clean data, and the accuracy of the same model tested on an augmented validation set. Here, the augmentation is applied to the validation dataset in one pass, as a static augmentation. More formally we define:

Definition 1. *Let D_{train} and D_{val} be training and validation datasets drawn IID from the same* clean *data distribution, and let D'_{val} be derived from D_{val} by applying a stochastic augmentation strategy, a, once to each image in D_{val}, $D'_{val} = \{(a(x), y) \ : \forall \ (x, y) \in D_{val}\}$. Further let m be a model trained on D_{train} and $\mathcal{A}(m, D)$ denote the model's accuracy when evaluated on dataset D. The Affinity, $\mathcal{T}[a; m; D_{val}]$, is given by*

$$\mathcal{T}[a; m; D_{val}] = \mathcal{A}(m, D'_{val}) - \mathcal{A}(m, D_{val}) \, . \tag{1}$$

With this definition, Affinity of zero represents no shift and a negative number suggests that the augmented data is out-of-distribution for the model.



(a) Affinity (b) D_{KL}

Figure 2: **Affinity is a model-sensitive measure of distribution shift**. Contours indicate lines of equal (a) Affinity, or (b) KL Divergence between the joint distribution of the original data and targets and the shifted data. The two axes indicate the actual shifts that define the augmentation. Affinity captures model-dependent features, such as the decision boundary.

In Fig. 2 we illustrate Affinity with a two-class classification task on a mixture of two Gaussians. Augmentation in this example comprises shift of the means of the Gaussians of the validation data compared to those used for training. Under this shift, we calculate both Affinity and KL divergence of the shifted data with respect to the original data. Affinity changes only when the shift in the data is with respect to the model's decision boundary, whereas the KL divergence changes even when data is shifted in the direction that is irrelevant to the classification task. In this way, Affinity captures what is relevant to a model: shifts that impact predictions.

This same metric has been used as a measure of a model's robustness to image corruptions that do not change images' semantic content (Azulay & Weiss, 2018; Dodge & Karam, 2017; Ford et al., 2019; Hendrycks & Dietterich, 2018; Rosenfeld et al., 2018; Yin et al., 2019). Here we, turn this around and use it to quantify the shift of augmented data compared to clean data.

Affinity has the following advantages as a metric:

1. It is easy to measure. It requires only clean training of the model in question.
2. It is independent of any confounding interaction between the data augmentation and the training process, since augmentation is only used on the validation set and applied statically.
3. It is a measure of distance sensitive to properties of both the data distribution *and* the model.

We gain confidence in this metric by comparing it to other potential model-dependant measures of distribution shift. We consider the mean log likelihood of augmented test images(Grathwohl et al., 2019), and the Watanabe–Akaike information criterion (WAIC) (Watanabe, 2010). These other metrics have high correlation with Affinity. Details can be found in Sec. F.

3.2 Diversity: A measure of augmentation complexity

Inspired by the observation that multi-factor augmentation policies such as `FlipLR`+`Crop`+`Cutout` and `RandAugment`(Cubuk et al., 2019) greatly improve performance, we propose another axis on which to view augmentation policies, which we dub *Diversity*. This measure is intended to quantify the intuition that augmentations prevent models from over-fitting by increasing the number of samples in the training set; the importance of this is shown in Sec. 4.3.

Based on the intuition that more diverse data should be more difficult for a model to fit, we propose a model-based measure. The Diversity metric in this paper is the final training loss of a model trained with a given augmentation:

Definition 2. *Let a be an augmentation and D'_{train} be the augmented training data resulting from applying the augmentation, a, stochastically. Further, let L_{train} be the training loss for a model, m, trained on D'_{train}. We define the Diversity, $\mathcal{D}[a; m; D_{train}]$, as*

$$\mathcal{D}[a; m; D_{train}] := \mathbb{E}_{D'_{train}} \left[L_{train} \right] . \tag{2}$$

Though determining the training loss requires the same amount of work as determining final test accuracy, here we focus on this metric as a tool for understanding.

As with Affinity, this definition of Diversity has the advantage that it can capture model-dependent elements, i.e. it is informed by the class of priors implicit in choosing a model and optimization scheme as well as by the stopping criterion used in training.

Another potential diversity measure is the entropy of the transformed data, $\mathcal{D}_{\text{Ent}}$. This is inspired by the intuition that augmentations with more degrees of freedom perform better. For discrete transformations, we consider the conditional entropy of the augmented data.

$$\mathcal{D}_{\text{Ent}} := H(X'|X) = -\mathbb{E}_X \left[\Sigma_{x'} p(x'|X) \log(p(x'|X)) \right] .$$

Here $x \in X$ is a clean training image and $x' \in X'$ is an augmented image. This measure has the property that it can be evaluated without any training or reference to model architecture. However, the appropriate entropy for continuously-varying transforms is less straightforward.

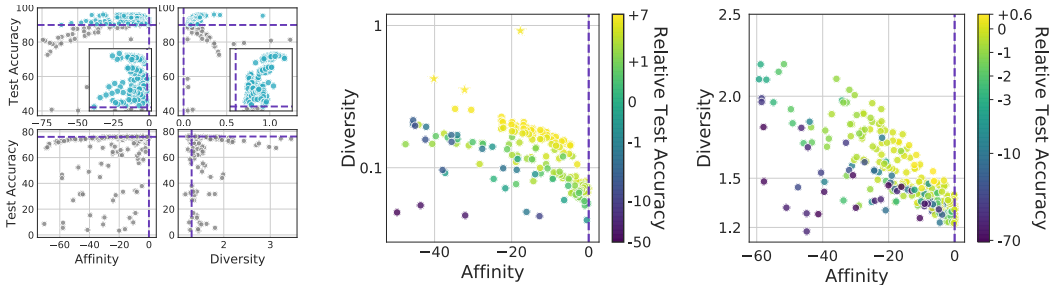
A third proxy for Diversity is the training time needed for a model to reach a given training accuracy threshold. In Sec. E, we show that these three metrics correlate well with each other.

In the remaining sections we describe how the complementary metrics of Diversity and Affinity can be used to characterize and understand augmentation performance.

4 Results

4.1 Augmentation performance is determined by both Affinity and Diversity

Despite the original inspiration to mimic realistic transformations and minimize distribution shift, many state-of-the-art augmentations yield unrealistic images. This suggests that distribution shift alone does not fully describe or predict augmentation performance.



(a) T: CIFAR-10. B: ImageNet (b) CIFAR-10 (c) ImageNet

Figure 3: **Augmentation performance is determined by both Affinity and Diversity**. (a) Test accuracy plotted against each of Affinity and Diversity for the two datasets, showing that neither metric alone predicts performance. In the CIFAR-10 plots (top), blue highlights (also in inset) are the augmentations that increase test accuracy above the clean baseline. Dashed lines indicate the clean baseline. (b) and (c) show test accuracy relative to clean baseline on the color scale in the plane of Affinity and Diversity. The three star markers in (b) are (left to right) `RandAugment`, `AutoAugment`, and `mixup`. For fixed values of Affinity, test accuracy generally increases with higher values of Diversity. For fixed values of Diversity, test accuracy generally increases with higher values of Affinity.

Figure 3(a) (left) measures Affinity across 204 different augmentations for CIFAR-10 and 223 for ImageNet respectively. We find that for the most important augmentations—those that help performance—Affinity is a poor predictor of accuracy. Furthermore, we find many successful augmentations with low Affinity. For example, `Rotate(fixed, 45deg, 50%)`, `Cutout(16)`,

and combinations of `FlipLR`, `Crop(32)`, and `Cutout(16)` all have Affinity$< -15\%$ and test accuracy$> 2\%$ above clean baseline on CIFAR-10. Augmentation details are in Sec. B.

As Affinity does not fully characterize the performance of an augmentation, we seek another metric. To assess the importance of an augmentation's complexity, we measure Diversity across the same set of augmentations. We find that Diversity is complementary in explaining how augmentations can increase test performance. As shown in Fig. 3(b) and (c), Affinity and Diversity together provide a much clearer parameterization of an augmentation policy's benefit to performance. For a fixed level of Diversity, augmentations with higher Affinity are consistently better. Similarly, for a fixed Affinity, it is generally better to have higher Diversity.

A simple case study is presented in Fig. 4. The probability of the transform `Rotate(fixed, 60deg)` is varied. The accuracy and Affinity are not monotonically related, with the peak accuracy falling at an intermediate value of Affinity. Similarly, accuracy is correlated with Diversity for low probability transformations, but does not track for higher probabilities. The optimal probability for `Rotate(fixed, 60deg)` lies at an intermediate value of Affinity and Diversity.

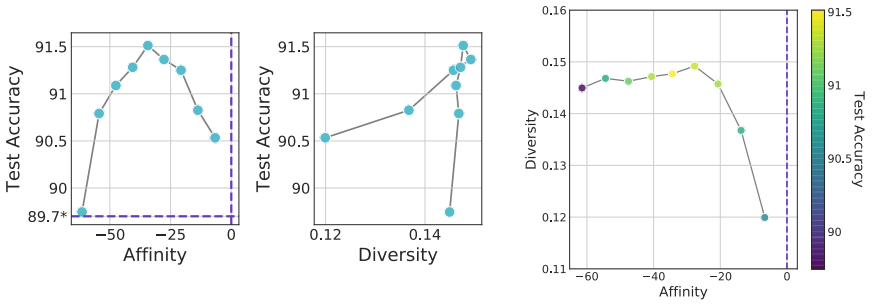


Figure 4: **Test accuracy varies differently than either Affinity or Diversity**. Here, the probability of `Rotate(fixed, 60deg)` on CIFAR-10 is varied from 10% to 90%. Left: as probability increases, Affinity decreases linearly while the accuracy changes non-monotonically. Center: accuracy and Diversity vary differently from each other as probability is changed. Right: test accuracy is maximized at intermediate values.

To situate the tested augmentations—mostly single transforms—within the context of the state-of-the-art, we tested three high-performance augmentations from literature: `mixup` (Zhang et al., 2017), `AutoAugment` (Cubuk et al., 2018), and `RandAugment` (Cubuk et al., 2019). These are highlighted with a star marker in Fig. 3(b).

More than either of the metrics alone, Affinity and Diversity together provide a useful parameterization of an augmentation's performance. We now turn to investigating the utility of this tool for explaining other observed phenomena of data augmentations.

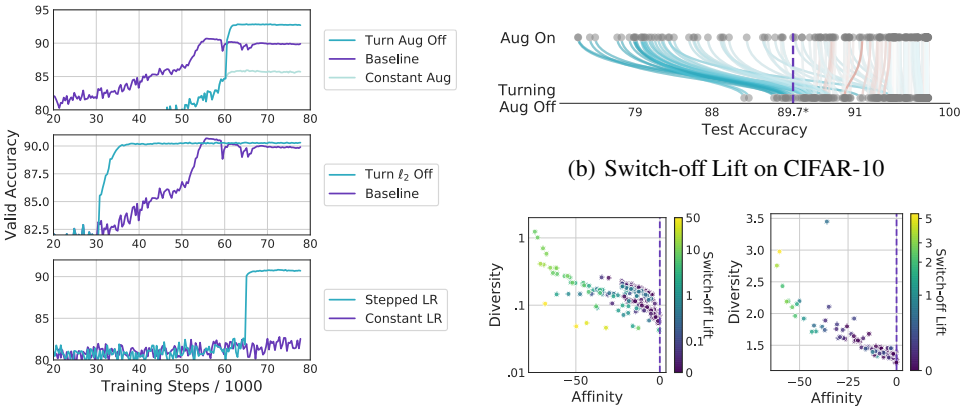
4.2 Turning augmentations off may adjust Affinity, Diversity, and performance

The term "regularizer" is ill-defined in the literature, often referring to any technique used to reduce generalization error without necessarily reducing training error (Goodfellow et al., 2016). With this definition, it is widely acknowledged that commonly-used augmentations act as regularizers (Hernández-García & König, 2018; Zhang et al., 2016; Dao et al., 2019). Though this is a broad definition, we notice another commonality across seemingly different kinds of regularizers: various regularization techniques yield boosts in performance (or at least no degradation) if the regularization is *turned off* at the right time during training. For instance:

1. Decaying a large learning rate on an appropriate schedule can be better than maintaining a large learning rate throughout training (Zagoruyko & Komodakis, 2016).
2. Turning off ℓ_2 at the right time does not hurt performance (Golatkar et al., 2019).
3. Relaxing model constraints mid-training can boost performance (d'Ascoli et al., 2019).
4. Fine-tuning on clean data can improve final accuracy (He et al., 2019).

To further study augmentation as a regularizer, we compare the constant augmentation case (with the same augmentation throughout) to the case where the augmentation is turned off partway through

training and training is completed with clean data. For each transform, we test over a range of switch-off points and select the one that yields the best final validation or test accuracy on CIFAR-10 and ImageNet respectively. The Switch-off Lift is the resulting increase in final test accuracy, compared to training with augmented data the entire time.



(a) Slingshot effect on CIFAR-10

(b) Switch-off Lift on CIFAR-10

(c) CIFAR-10 (left) ImageNet (right)

Figure 5: **(a) Switching off regularizers yields a performance boost**: Three examples of how turning off a regularizer increases the validation accuracy. This slingshot effect can speed up training and improve the best validation accuracy. Top: training with no augmentation (clean baseline), compared to constant augmentation, and augmentation that is turned off at 55k steps. Here, the augmentation is `Rotate(fixed, 20deg,100%)`. Middle: Baseline with constant ℓ_2. This is compared to turning off ℓ_2 regularization part way through training. Bottom: Constant learning rate of 0.1 compared to training where the learning rate is decayed in one step by a factor of 10. **(b) Bad augmentations can become helpful if switched off**: Colored lines connect the test accuracy with augmentation applied throughout training (top) to the test accuracy with switching mid-training. Color indicates the amount of Switch-off Lift; blue is positive and orange is negative. **(c) Switch-off Lift varies with Affinity and Diversity**. Where Switch-off Lift is negative, it is mapped to 0 on the color scale.

For some poor-performing augmentations, this gain can actually bring the test accuracy above the baseline, as shown in Fig. 5(b). We additionally observe (Fig. 5(a)) that this test accuracy improvement can happen quite rapidly for both augmentations and for the other regularizers tested. This suggests an opportunity to accelerate training without hurting performance by appropriately switching off regularization. We call this a *slingshot* effect.

Interestingly, we find the best time for turning off an augmentation is not always close to the end of training, contrary to what is shown in He et al. (2019). For example, without switching, `FlipUD(100%)` decreases test accuracy by almost 50% compared to clean baseline. When the augmentation is used for only the first third of training, final test accuracy is above the baseline.

He et al. (2019) hypothesized that the gain from turning augmentation off is due to recovery from a distribution shift. Indeed, for many detrimental transformations, the test accuracy gained by turning off the augmentation merely recovers the clean baseline performance. However, in Fig. 5(c), we see that for a given value of Affinity, Switch-off Lift can vary. This result suggests that the Switch-off Lift is derived from more than simply correction of a distribution shift.

A few of the tested augmentations, such as `FlipLR(100%)`, are fully deterministic. Thus, each time an image is drawn in training, it is augmented the same way. When such an augmentation is turned off partway through training, the model then sees images—the clean ones—that are now new. Indeed, when `FlipLR(100%)` is switched off at the right time, its final test accuracy exceeds that of `FlipLR(50%)` without switching. In this way, switching augmentation off may adjust for not only low Affinity but also low Diversity.

4.3 Increased effective training set size is crucial for data augmentation

Most augmentations we tested and those used in practice have inherent stochasticity and thus may alter a given training image differently each time the image is drawn. In the typical *dynamic* training mode, these augmentations increase the number of unique inputs seen across training epochs.

To further study how augmentations act as regularizers, we seek to discriminate this increase in effective dataset size from other effects. We train models with *static* augmentation, as described in Sec. 3. This altered training set is used without further modification during training so that the number of unique training inputs is the same between the augmented and the clean training settings.

For almost all tested augmentations, using static augmentation yields lower test accuracy than the clean baseline. Where static augmentation shows a gain (versions of `crop`), the difference is less than the standard error on the mean. As in the dynamic case, poorer performance in the static case is for transforms that have lower Affinity and lower Diversity.

Static augmentations also always perform worse than their non-deterministic, dynamic counterparts, as shown in Fig. 6. This may be because the Diversity of a static augmentation is always less than the dynamic case (see also Sec. E). The decrease in Diversity in the static case suggests a connection between Diversity and the number of training examples.

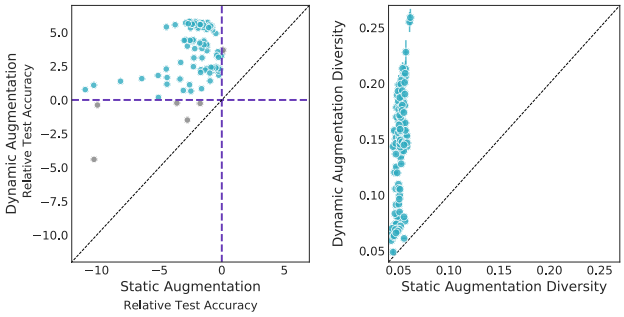


Figure 6: **Static augmentations decrease diversity and performance**. CIFAR-10, static augmentation performance is less than the clean baseline, $(0, 0)$, and less than the dynamic augmentation case. Augmentations with no stochasticity are excluded because they are trivially equal on the two axes (left). Diversity in the static case is less than in the dynamic case.(right) Diagonal line indicates where static and dynamic cases would be equal.

Together, these results point to the following conclusion: *Increased effective training set size is crucial to the performance benefit of data augmentation. An augmentation's Affinity and Diversity inform how useful the additional training examples are.*

5 Discussion

In this work, we conducted an empirical study in an attempt to characterize the essential factors leading to successful augmentation policies. This builds a foundation for using these metrics to quantify and design more complex and powerful combinations of augmentations. We further hope that Affinity and Diversity may serve as useful metrics of distribution shift more generally.

Though earlier work has often explicitly focused on just one of these metrics, chosen priors have implicitly ensured reasonable values for both. One way to achieve Diversity is to use combinations of many single augmentations, as in AutoAugment (Cubuk et al., 2018). Because transforms and hyperparameters in Cubuk et al. (2018) were chosen by optimizing performance on proxy tasks, the optimal policies include high and low Affinity transforms. Fast AutoAugment (Lim et al., 2019), CTAugment (Berthelot et al., 2019; Sohn et al., 2020), and differentiable RandAugment (Cubuk et al., 2019) all aim to increase Affinity by what Lim et al. (2019) called "density-matching". However these methods use the search space of AutoAugment and thus inherit its Diversity.

On the other hand, Adversarial AutoAugment (Zhang et al., 2019) focused on increasing Diversity by optimizing policies to increase the training loss. While this method did not explicitly aim to increase Affinity, it also used transforms and hyperparameters from the AutoAugment search space which implicitly have high Affinity. Without such a prior, the goal of maximizing training loss with no other constraints would lead to data augmentation policies that erase all the information from the images.

6 Conclusion

We attempted to quantify common intuition that more in-distribution and more diverse augmentation policies perform well. To this end, we introduced two easy-to-compute metrics, Affinity and Diversity, intended to measure to what extent a given augmentation is in-distribution and how complex the augmentation is to learn. Because they are model-dependent, these metrics capture the data shifts that affect model performance. We hope our findings provide a foundation for continued scientific study of data augmentation and more general distribution shifts.

REFERENCES

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

J. R. Bellegarda, P. V. de Souza, A. J. Nadas, D. Nahamoo, M. A. Picheny, and L. R. Bahl. Robust speaker adaptation using a piecewise linear acoustic mapping. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992.

Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 164–172, 2011.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Feb 2017. ISSN 1537-274X. doi: 10.1080/01621459.2017.1285773. URL http://dx.doi.org/10.1080/01621459.2017.1285773.

Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649. IEEE, 2012.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pp. 6510–6520, 2017.

Tri Dao, Albert Gu, Alexander J Ratner, Virginia Smith, Christopher De Sa, and Christopher Ré. A kernel theory of modern data augmentation. *Proceedings of machine learning research*, 97:1528, 2019.

Stéphane d'Ascoli, Levent Sagun, Joan Bruna, and Giulio Biroli. Finding the needle in the haystack with convolutions: on the benefits of architectural bias, 2019.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pp. 1–7. IEEE, 2017.

Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1301–1310, 2017.

Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 682–691, 2019.

Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one, 2019.

Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

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, 2019.

Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

Alex Hernández-García and Peter König. Further advantages of data augmentation on convolutional neural networks. In *International Conference on Artificial Neural Networks*, pp. 95–103. Springer, 2018.

Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. *arXiv preprint arXiv:1905.05393*, 2019.

Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2014. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, et al. Starnet: Targeted computation for object detection in point clouds. *arXiv preprint arXiv:1908.11069*, 2019.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization, 2016.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, pp. 3239–3249, 2017.

Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015.

Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of International Conference on Document Analysis and Recognition*, 2003.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. *arXiv preprint arXiv:1911.09070*, 2019.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy, 2019.

Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems*, pp. 2794–2803, 2017.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pp. 1058–1066, 2013.

Sumio Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *J. Mach. Learn. Res.*, 11:3571–3594, December 2010. ISSN 1532-4435.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V Le. Adversarial examples improve image recognition. *arXiv preprint arXiv:1911.09665*, 2019a.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019b.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

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

Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment, 2019.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.