Disentangling the Effects of Data Augmentation

Reza Moravej *
Department of Industrial Engineering
University of Toronto
Toronto, ON, CA
mreza.moravej@mail.utoronto.ca

Joseph Hung *
Department of Physics
University of Toronto
Toronto, ON, CA
joseph.hung@mail.utoronto.ca

Shawn Carere *
Department of Medical Biophysics
University of Toronto
Toronto, ON, CA
shawn.carere@mail.utoronto.ca

Abstract

In this paper we explore the use of data augmentation from an empirical approximation perspective. We identify that data augmentation can have two distinct roles in model generalization and show experimentally that these roles can be separated. Furthermore we analyze each role individually and provide various conclusions and hypotheses about their effects on model training dynamics. We include empirical evidence of behaviours elicited by data augmentation using a VGG based CNN on the CIFAR10 and COIL100 datasets. Our findings indicate that data augmentation not only selectively regularizes features but also removes uncertainty from the training loss estimate which prevents overfitting and allows the model to achieve a better optimum. We also suggest various directions for future work that we believe offer promising insights. Our work contributes to a public research project called the 'Algorithmic Efficiency Benchmark'

*Work done as part of final project for CSC2541 - Neural Network Training Dynamics - Department of Computer Science

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

1 Introduction

Data augmentation techniques are prevalent in a variety of supervised learning tasks. Of particular interest are computer vision tasks, which are often heavily reliant on large datasets and thus utilize a wide variety of image augmentation techniques to prevent overfitting of deep and complex models. Although these augmentation techniques have been shown to be very effective in improving model generalization [1, 2], the choice of which augmentations to use and the magnitudes by which to apply them is often empirically (but loosely) justified. The current approach to data augmentation is often essentially a series of heuristic guesses. Recent work introduced a new state of the art approach to data augmentation by trying to learn optimal data augmentation policies through reinforcement learning [3]. However this approach is expensive as it requires training multiple child models in order to sample a potential augmentation policy and generate an error signal for the controller. Furthermore this approach still lacks a detailed understanding of how augmentation policies are affecting model training, or why a particular policy works better than another. Although the ability to automate data augmentation policy selection is a powerful tool, a deeper understanding of how various augmentation operations affect model training is required in order to make these tools more intelligent.

Augmentation has been used to achieve nearly every state of the art result on image recognition and has become a standard practice in modern machine learning. Despite the obvious impacts of augmentation, a large focus of the machine learning community has remained on improving network architectures. Comparatively little attention has been given to improving data augmentation, or understanding why and how it works. This may be because in many cases, it is not difficult to find an augmentation policy that works relatively well. There may be little interest in finding an optimal augmentation policy due to the perception that it is not an integral part of learning. However data augmentation can be as effective as regularization or changes in model architecture when trying to improve model generalizability. In our experiments, just adding random translations and flipping to a CIFAR10 baseline increases validation accuracy by over 10% and effectively prevents overfitting. A data augmentation policy can be thought of as an input layer with fixed weights explicitly chosen using prior knowledge. Viewing data augmentation from this perspective highlights its importance and the need to understand it. In this paper we explore some common assumptions about data augmentation, and try to understand its effects on model training at a deeper level. Our work contributes to a public research project called the 'Algorithmic Efficiency Benchmark' [1], of which the primary goal is to rigorously compare machine learning algorithms on their ability to reach convergence faster. We have made our implementation publicly available [2].

2 Preliminaries

2.1 Summary

In this paper we explore data augmentation with 3 distinct experiments summarized in Figure 1. Our work contributes to a public research project called the 'Algorithmic Efficiency Benchmark' and hence we implement our experiments using the framework of this project. We summarize our contributions as follows

- We first provide a theoretical analysis of data augmentation and its distinct effects on generalization.
- We group the effects of data augmentation into categories, showing that one can improve model performance without increasing the number of training samples, and the other significantly reduces overfitting.
- We show that the benefit obtained from data augmentation is not uniform with respect to the number of samples in the original training set.
- We study how applying different Augmentation techniques affects the model reliance on *evocative* features. Our results indicate that given two sets of *evocative* (and sufficient) features, the model latches on one of the feature sets only. We show that augmentations which add noise to non-essential feature improve the model accuracy more.

[1] https://mlcommons.org/en/groups/research-algorithms/

[2] https://github.com/UofT-EcoSystem/algorithmic-efficiency/tree/augmentation

- Finally, we compare the impact of augmentation and dropout on the model reliance on a particular feature class.

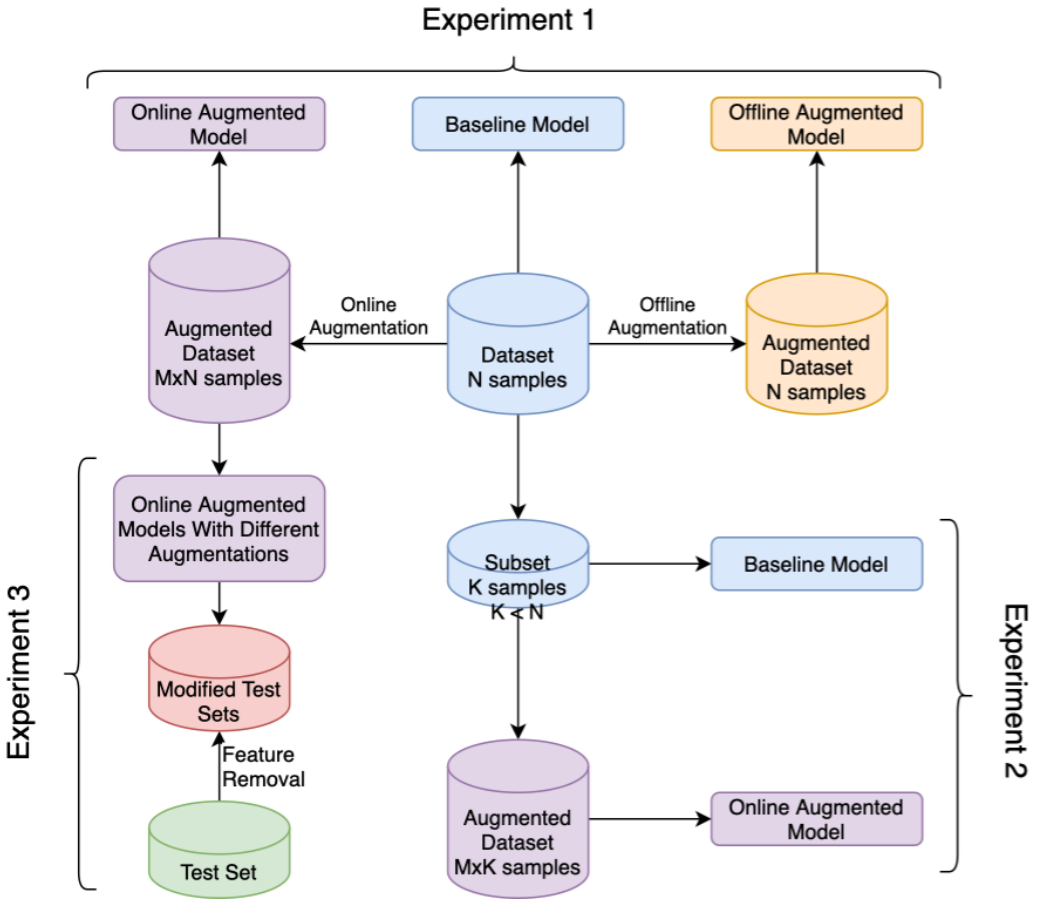


Figure 1: Summary of empirical experiments in this paper. Experiment 1 compares traditional online data augmentation to offline augmentation *with* replacement. Experiment 2 analyzes how the amount of original data affects data augmentation. Experiment 3 analyzes how different types of data augmentation affect the model's dependence on certain features.

2.2 Motivation

Existing work has theorized that the main mechanism by which data augmentation improves generalization is by effectively increasing the number of training samples. Here we show some motivation behind this belief as initially formulated by [4], and expand upon it to motivate our interest in whether or not this assumption is true.

Let $\mathcal{X}$ and $\mathcal{Y}$ be the data and label spaces respectively where each sample $(x, y) \sim \mathcal{P}$ is part of a joint distribution $\mathcal{P}$. The goal of learning is to find a function $f : \mathcal{X} \to \mathcal{Y}$ which minimizes the expected value of a loss function $\ell(f(x), y)$ over the distribution $\mathcal{P}$, which encompasses all possible samples a model may see throughout its lifetime. This is known as the *expected risk* seen in equation 1.

$$\mathcal{R}(f|\mathcal{P}) = \int \ell(f(x), y) d\mathcal{P}(x, y) \tag{1}$$

In practice we instead minimize an approximation of this known as the *empirical risk* seen in equation 2. This is known as empirical risk minimization (ERM). We expand upon the notation of the original authors in [4] by clarifying that the *empirical risk* of the distribution $\mathcal{P}$ is equivalent to the *expected*

risk of the distribution $\mathcal{P}_{train}$ derived from some finite training set with N samples. The accuracy of the empirical estimation of the expected risk of $\mathcal{P}$ goes up with N.

$$\mathcal{R}(f|\mathcal{P}_{train}) = \hat{\mathcal{R}}(f|\mathcal{P}) = \frac{1}{N}\sum_{n=1}^{N} \ell(f(x_n), y_n) \tag{2}$$

Using Vapnik-Chervonenkis (VC) Theory [5], one can bound the generalization of a model with probability $1-\delta$ using equation 3, where f is assumed to be a binary classifier with a finite VC-dimension of $|\mathcal{F}|_{VC}$. $\frac{1}{2} \leq \alpha \leq 1$ and is dependent on the complexity of the case.

$$\mathcal{R}(f|\mathcal{P}) - \mathcal{R}(f|\mathcal{P}_{train}) \leq \mathcal{O}\left(\left(\frac{|\mathcal{F}|_{VC} - log\delta}{N}\right)^{\alpha}\right) \tag{3}$$

To go one step further, in machine learning we empirically estimate the generalization proposed in equation 3 with a test dataset using the assumption in equation 4

$$\mathcal{R}(f|\mathcal{P}) - \mathcal{R}(f|\mathcal{P}_{train}) \approx \mathcal{R}(f|\mathcal{P}_{test}) - \mathcal{R}(f|\mathcal{P}_{train}) \tag{4}$$

Building off equation 3, the authors of [4] frame the generalization benefits of training with augmented data using equation 5. Again we replace the empirical risk of $\mathcal{P}$ with the expected risk of $\mathcal{P}_{train}$ for clarity. f_{aug} is the function that minimizes the empirical risk $\hat{\mathcal{R}}(f|\mathcal{P}_{aug})$ where $\mathcal{P}_{aug}$ is the joint distribution for the augmented data with $M \times N$ samples. It is important to note that $\hat{\mathcal{R}}(f|\mathcal{P}_{aug}) \neq \mathcal{R}(f|\mathcal{P}_{aug})$. ϵ is an error term that arises due to the distribution gap between $\mathcal{P}_{aug}$ and $\mathcal{P}$ and is assumed to be small for non-intensive data augmentation.

$$\mathcal{R}(f_{aug}|\mathcal{P}) \leq \mathcal{R}(f_{aug}|\mathcal{P}_{train}) + \mathcal{O}\left(\left(\frac{|\mathcal{F}|_{VC} - log\delta}{M \times N}\right)^{\alpha}\right) + \epsilon \tag{5}$$

From equation 5, the authors of [4] conclude, based on the first two terms of the equation, that data augmentation generalizes well as a result of two factors.

1-a) The empirical risk of $\mathcal{P}$ ($\mathcal{R}(f_{aug}|\mathcal{P}_{train}) = \hat{\mathcal{R}}(f_{aug}|\mathcal{P})$) is small

2-a) The augmented data contains a large number of samples

The authors in [4] interpret the above from the perspective that there is a limitation to increasing the number of samples with data augmentation without also increasing the empirical risk of $\mathcal{P}$. So long as $\mathcal{P} \approx \mathcal{P}_{aug}$, the empirical risk $\hat{\mathcal{R}}(f_{aug}|\mathcal{P})$ will remain small. This implies the conclusion that the data augmentation is a tradeoff between increasing the number of samples and preserving the distribution of the data. Rather than asking why f_{aug} generalizes well, we consider the function f^* which minimizes the empirical risk $\mathcal{R}(f|\mathcal{P}_{train})$ and ask why f_{aug} generalizes better than f^*. Below we re-frame the original authors hypotheses from equation 5 to reflect this change in perspective.

1-b) Augmentation reduces the empirical risk of $\mathcal{P}$ (ie. $\mathcal{R}(f_{aug}|\mathcal{P}_{train}) \leq \mathcal{R}(f^*|\mathcal{P}_{train})$)

2-b) The augmented data contains a larger number of samples

Notably, as denoted in 1-b), the empirical risk is no longer a term which must be kept from becoming large, but a term which may also be made smaller. While it is well established that increasing the number samples improves model generalization as suggested by 2-b), the effects of data augmentation on the empirical risk of $\mathcal{P}$ are often overlooked in existing literature. In Dao et al. [6], they show that for kernel classifiers, data augmentation can be explicitly approximated as first order feature averaging and second order variance regularization. This can be extrapolated to provide a hypothesis for how training with data augmentation might decrease the empirical risk $\mathcal{R}(f|\mathcal{P}_{train})$. For this reason, we will refer to factor 1 as the regularizing effects of augmentation. Conversely we refer to factor 2 as the sampling effects of augmentation since increasing MxN can be thought of as increasing the sampling resolution of the distribution $\mathcal{P}$. We make this simplification throughout the rest of the paper in order to increase readability. In this paper we aim to explore the regularizing and sampling effects of augmentation separately in order to better understand their effects on model training and generalizability.

2.3 Methods

There are several approaches to the application of data augmentation to model training. The first we will consider is the difference between augmenting images online and offline. The latter is performed by selectively augmenting the dataset prior to training while the former applies data augmentation randomly as data is loaded through the training process. Offline augmentation is typically done without replacement in order to increase the number of training samples. Provided that identical augmentations procedures are used, both online and offline augmentation will expose the model to the same distribution of training data, with the key difference that after one epoch, the offline augmentation process will have seen this full distribution, whereas online augmentation will not have (depending on the probability of augmentation). We expect that this stochasticity in the online case will have a regularizing effect and accordingly be useful in preventing overfitting, although this is not explored in our experiments.

We also have the choice of applying augmentations on a per-batch or per-sample basis. The former is when augmentations are applied to all images in a batch, and the latter is when they are applied individually to all elements of the batch. We expect that augmenting images batch-wise may introduce some bias to the data, particularly for small datasets (such as the small subsets of CIFAR10 used in Experiment 2), so augments are applied per-sample throughout all experiments in this paper. Augmentations used in this paper include horizontal flipping with a probability of 0.5, translation of up to ± 2 pixels for CIFAR10 or ± 13 pixels for Coil100. Additionally, for some models in Experiment 3, we use random color inversion with a probability of 0.5 and gaussian noise with zero mean and a standard deviation of 0.1. In all experiments, images were normalized prior to training. In experiments 1 and 2, the input images were normalized to unit range by dividing by 255. In experiment 3, the input images were normalized to zero mean and unit variance.

The model used for all experiments was a CNN consisting of 3 VGG blocks [7] followed by a hidden layer with 128 units and an output layer. ReLU activation was used for all layers except the output layer where the logarithm of the softmax function was used. All models were trained with stochastic gradient descent with nesterov momentum of 0.9 and a learning rate of 0.01. Additionally, models that used dropout in experiment 3 had dropout layers with probability 0.2 after each VGG block as well as the hidden layer.

3 Experiment 1: Separating the Regularizing and Sampling Effects of Data Augmentation

3.1 Motivation and Experimental Setup

Data augmentation is typically thought to benefit model training primarily through artificially increasing the number of samples. Although this is true, the actual mechanisms by which data augmentation improves model performance are more nuanced. As discussed in Section 2.2, these mechanisms can be separated into two groups based on how they effect model generalization. For readability, we refer to these groups as the regularizing and sampling effects of data augmentation respectively. However this is likely to be an oversimplification. That being said, in this section we aim to separate these effects thus showing that data augmentation improves model generalization in multiple ways, many of which are often overlooked.

We test this by training 3 identical models using different augmentation procedures. We denote these procedures in the list below. The augmentations used were random pixel shifting and horizontal flipping. The details of model architecture, training and augmentation can be found in Section 2.3.

- **Baseline**: Trained with original unaugmented data
- **Offline Augmented**: Training data was augmented offline *with replacement*.
- **Online Augmented**: Training data was augmented online

The baseline serves to show how the model performs without the benefits of data augmentation. Notably, for the offline augmented model, the augmentation is done prior to training *with* replacement. This ensures that both the baseline and offline augmented model see the same amount of unique samples throughout training. We do this to isolate the regularizing effects of data augmentation. Since the sampling effects are dependent on the amount of unique samples in the training set, they

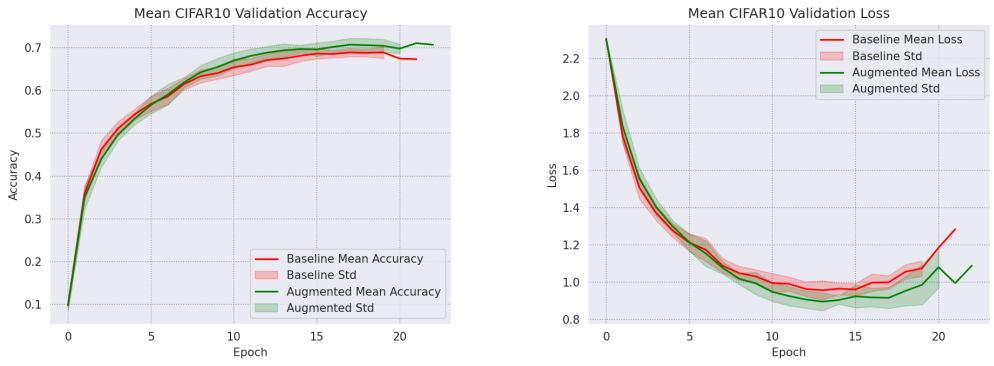
will not contribute to the differences between the baseline and offline augmented model. Lastly we also train a model with online data augmentation to add these sampling effects back in. It is worth noting however that although we can isolate the regularizing effects with the offline augmented model, we cannot isolate the sampling effects in a similar fashion. This is because we cannot rule out that additional samples seen by the online augmented model do not increase the magnitude of the regularization effects as well. However, we chose online augmentation because the number of unique samples seen by the model is significantly greater, allowing the sampling effects to be present as much as possible.

3.2 Results and Discussion

In order to analyze the regularizing effects of data augmentation, we first compare the offline augmented model to the baseline. The results from this experiment can be seen in Table 1. One can see that despite having the exact same number of unique samples, the offline augmented model achieves better generalization performance than the baseline with over 95% confidence. This shows that data augmentation does indeed improve model generalization by changing the loss landscape and decreasing the empirical risk $\hat{\mathcal{R}}(f|\mathcal{P})$. More specifically, it does this through means other than introducing additional samples. Augmentations are typically hand selected for the task using prior knowledge. Therefore it is possible that this prior knowledge is encoded into the training dataset via augmentation, altering the loss landscape in a meaningful way. He et al. [4] theorize that for intensive data augmentation, a distributional shift regularizes minor features, which reduces the amount of local minima. However the augmentations used in this experiment align with what the authors described as non-intensive, and therefore are unlikely to cause a significant distributional shift. Therefore even non-intensive data augmentations alter the loss landscape in such a way that is more complicated than simply smoothing the loss landscape.

Dataset	Validation Loss	Validation Accuracy (%)
Baseline	0.9490 ± 0.0361	67.93 ± 0.94
Offline Augmented	$\mathbf{0.8828} \pm 0.0386$	$\mathbf{70.11} \pm 0.87$

Table 1: The loss and accuracy for a CNN trained on CIFAR10. The samples in the augmented dataset were augmented using a random shift of up to 2 pixels in either direction for both the horizontal and vertical axes as well as random flipping in the horizontal direction with a probability of 0.5. Shown are the averages and standard deviation across 5 trials trained with the same hyperparameters



(a) Validation Accuracy

(b) Validation Loss

Figure 2: Metrics for a basic CNN with 3 VGG blocks trained on CIFAR10. Red is the regular dataset and green is the dataset augmented offline with replacement. Each model was trained 5 times with the same hyperparameters. The curve represents the mean across the 5 trials and the shaded region indicates one standard deviation.

We can start to analyze how the regularizing effects of these non-intensive data augmentations are affecting model generalization throughout training by looking at the validation metrics in Figure

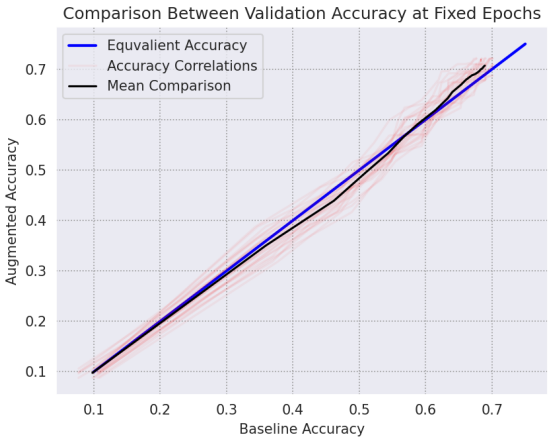


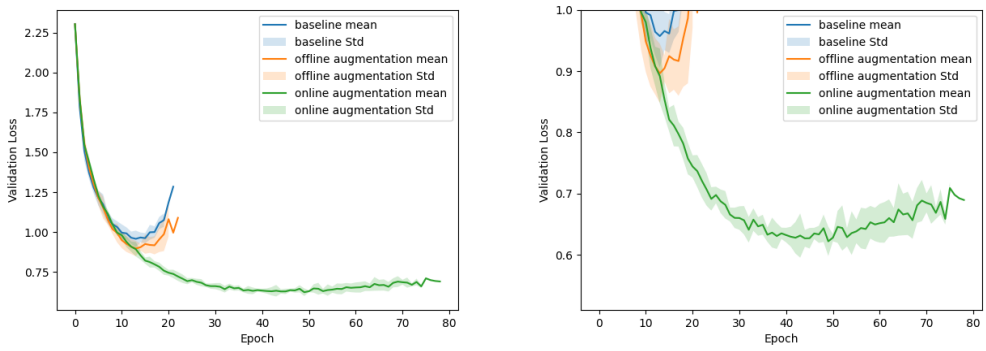
Figure 3: Accuracy of baseline and offline augmented models compared on a fixed-epoch basis to view relative convergence rates throughout training.

2. One can see that although the offline augmented model achieves better performance overall, it actually takes a longer time to converge than the baseline model. This is further demonstrated in Figure 3. At the very beginning of training, both models converge at similar rates. In this stage the convergence is nearly linear and dominated by the large magnitude of the gradients due to high curvature directions. We refer to this as the high-curvature domain. Conversely, we define the low curvature domain as the region in which the magnitude of the gradients are comparatively small, and convergence is dominated by the optimizer's ability to navigate low curvature directions. In this region, near the end of training, the offline augmented model finds a better optimum than the baseline model. Lastly, what we are most interested in is what we refer to as the mixed curvature domain, the transitional state between the high and low curvature domains. Although the bounds between all three domains is currently subjective, we direct the reader to the region of convergence between epochs 2 and 5/8 for the baseline and offline augmented model respectively. Notably, the offline augmented model converges more slowly through this region than the baseline. That being said it remains in the mixed curvature domain for longer surpassing the baseline's performance before entering the low-curvature domain. Since both models use the same optimization algorithm, this difference in convergence is a reflection of changes to the loss landscape. We propose two possibilities:

1. Directions previously from the low-curvature domain have increased in curvature
2. Directions previously from the high-curvature domain have decreased in curvature

Given that data augmentation is believed to have regularizing effects [2, 6, 8, 9], we believe the second possibility to be more likely. This somewhat contradicts the findings from [4] which theorized that data augmentation regularized *minor features* allowing the model to focus on more important features. However that was for intensive data augmentations which significantly change the distribution of the training data. That being said, how data augmentation effects the loss landscape and the rate of convergence requires further exploration.

Lastly, in order to analyze the sampling effects of data augmentation, we compare both the baseline and offline augmented models to a model trained with online data augmentation. One can see from Figure 4 that the online augmented model achieves a significantly better optimum. The reason for this is likely a combination of introducing the sampling effects of augmentation by increasing the number of unique training samples, as well as an increase in magnitude of the regularizing effects observed with the offline augmented model. We note however that the online augmented model is significantly more robust to overfitting. In fact the magnitude of overfitting exhibited by the online augmented model is so small that it is only clearly visible after increasing the y-axis resolution in Figure 4b. Conversely, the offline augmented model exhibits a similar amount of overfitting to the baseline. Therefore we hypothesize that robustness to overfitting is primarily a result of the sampling effects of augmentation as opposed to the regularizing effects. We explain this using the two terms



(a) Online vs Offline Augmentaion

(b) Online vs Offline Augmentation Zoomed

Figure 4: Validation loss and accuracy comparisons of baseline with both online and offline augmentation a) Validation loss curves for baseline, offline augmented and online augmented models b) Same as figure 4a but with a higher resolution y-axis

that form the upper bound for the expected risk in Equation 5. Note we estimate the expected risk throughout training using validation loss (see Equation 4).

Empirical risk dominates: The first term is the empirical risk which, for non-intensive data augmentation, corresponds to the training loss [4]. At the beginning of training, the magnitude of the training loss is still high and therefore it dominates the upper bound in Equation 5. This means that at the beginning of training one can be confident that the model's performance on the validation set will be approximately equivalent or better than it's performance on the training set. Hence we say our training loss throughout this phase is a *semi-accurate* estimation of the validation loss.

VC-error dominates: We refer to the second term of Equation 5 as the VC-error since it is derived from VC theory [5]. As training proceeds, the empirical risk will continue to decrease in magnitude until eventually the VC-error dominates Equation 5. In this phase, the magnitude of the expected risk ($\mathcal{R}(f|\mathcal{P})$)) is no longer bounded by the empirical risk ($\mathcal{R}(f|\mathcal{P}_{train})$)). This creates the necessary conditions for overfitting, wherein our empirical estimation of the expected risk is underestimated ($\mathcal{R}(f|\mathcal{P}_{train}) \ll \mathcal{R}(f|\mathcal{P})$) and hence generalization of the model is poor. Therefore we say that our training loss throughout this phase is an *inaccurate* estimation of the validation loss.

To bring this all together, we hypothesize that the online augmented model exhibits significantly less overfitting because the magnitude of the VC-error is significantly lower. Increasing the number of unique samples decreases VC-error and allows the empirical risk to dominate Equation 5. Furthermore, the online augmented model might reach a better optimum because it's training loss remains semi-accurate for longer, allowing the model to extract more information from the original data. Conversely, since the baseline and offline augmented model have higher VC-error, the minimum training loss they can achieve, before the VC-error dominates and the estimation of the expected risk becomes inaccurate, is higher. Therefore they may not be able to use all the information available in the training set due to an empirical 'glass ceiling' (or more accurately perhaps, an empirical 'glass floor').

4 Experiment 2: Do Larger Datasets Benefit more from Data Augmentation?

4.1 Motivation and Experimental Setup

In Experiment 1, we explore the sampling effects of data augmentation by increasing the variable M in Equation 5 to a large value. This is done by using online augmentation, which for our implementation corresponds to $M = 48$ and comparing it to the baseline and offline augmented models with $M = 1$. In this experiment, we explore how the variable N from Equation 5 affects sampling by modifying the amount of original data.

This experiment is performed in the same setting as Experiment 1. To investigate the relation between the amount of original data and the improvement gained from data augmentation, we selectively load subsets (10%, 20%, ..., 100%) of the full CIFAR10 dataset (of 50,000 training examples), and train our CNN on this data. For each of the sub-datasets, the model is trained firstly on the original data, and secondly where augmentations are applied randomly in an online manner through training. The augmentations chosen for this experiment are identical to before: random translations up to two pixels horizontally/vertically, and random horizontal flipping. For each of the experimental runs, we hold all hyperparameters constant and perform 5 trials, training until overfitting to ensure we find an optimum for each model.

4.2 Results and Discussion

Figure 5 shows the validation loss associated with our model trained on varying amounts of original data, with and without data augmentations applied. As expected, increasing the amount of original data significantly increases the amount of iterations before the minimum validation loss is reached (and consequent overfitting, in the unaugmented case). We observe that the loss curves are initially all incident, when the models have seen none of the data, before diverging at varying points. The predominant effect of data augmentation is the lower minimum the optimization achieves, as well as a flattening of the loss as training progresses.

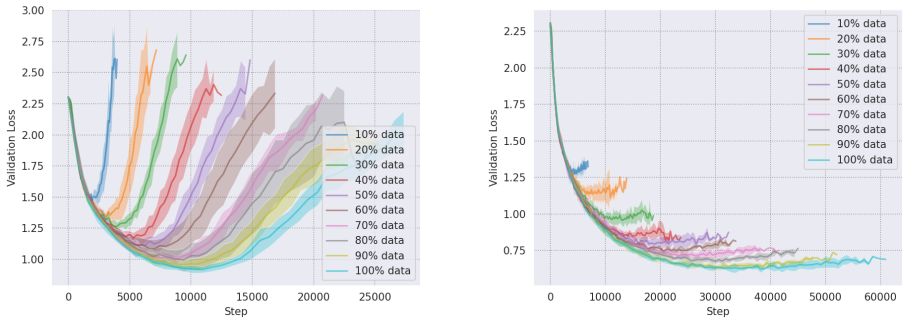
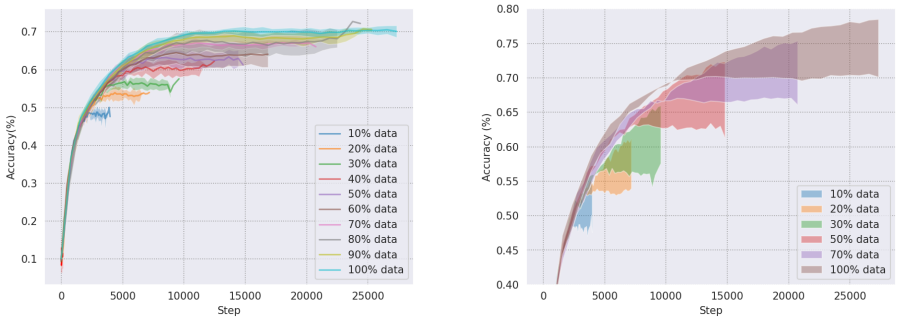


Figure 5: Validation loss for subsets of varying sizes for unaugmented data (left) and augmented data (right). Models were overfit purposefully to emphasize effectiveness of increased original data amounts. Minimum loss occurs further into the training process as amount of original data increases.

Figure 6 shows the corresponding validation accuracy through the training cycle. Again, as would be expected, the accuracy convergence for unaugmented data improves as we increase the amount of original data, from around 50% to 70% as we vary the dataset size from 10% to 100%. The addition of augmented data provided a sizable improvement to model accuracy. As in the observations of validation loss, the accuracy progresses incidentally for all dataset sizes before each subset converges independently.

The addition of (online) augmented data means the model can train for significantly longer before being exposed to repeated data and can delay overfitting. For this reason, the validation loss (Figure 5) does not significantly indicate overfitting for the augmented data, while the unaugmented data does). This observation is closely related to the findings in Figure 4

Figure 7 displays the improvement which the addition of data augmentation provides when applied to each original data subset. We see three regimes of note: the first that applying data augmentation on the quantity of original data from 10% to 30% results in a rapid improvement to the improvement of validation accuracy. In this case, it is likely that the model generalizes better simply because there are too few data points without augmentation and overfitting occurs quickly. Secondly, when the amount of original data approaches the full dataset, this results in *less* improvement as compared to the 30% - 60% data range. Curiously, this might not be expected behavior as increasing the amount of original data should only serve to increase the variety of data the model is exposed to. We believe this is occurring as the model + optimizer (SGD) is not powerful enough to train this model to optimality, so that the optimization on augmented data is already approaching the best accuracy this VGG/SGD setup can converge to. Thus, data augmentation has a reduced effect as the addition of further data is



(a) Unaugmented Validation Accuracy (b) Improvements as a result of augmentation

Figure 6: Accuracy metrics averaged over 5 trials for baseline and online augmented models with varying amounts of data from CIFAR10. (a) Mean validation accuracy for baseline models. Shaded region is standard deviation across 5 trials. (b) Improvement over baseline due to augmentation. The lower and upper bounds of the shaded region are the mean accuracy across 5 trials for the baseline and online augmented models respectively.

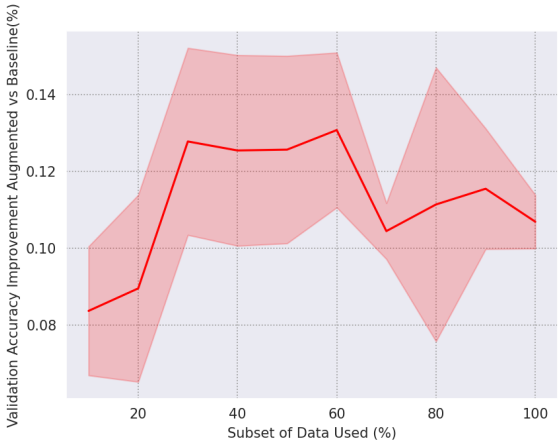


Figure 7: Improvement of validation accuracy for varying original data sizes. Improvements are calculated when model reaches minimum validation loss. Shaded region indicates standard deviation over 5 trials.

no longer as useful as in cases where the model was primarily constrained by lack of model expressive power. Finally, there is a plateau of improvement between 30% and 60% of original data. This is the region where the effect of data augmentation is maximized, and can be interpreted as when the optimization is neither limited by amount of training data nor the experimental setup, so we see the full benefit of data augmentation. The full training results are included in the Appendix in Figure 6

5 Experiment 3: The Impact of Data Augmentation and Dropout on Model Robustness

5.1 Motivation

CNN models have shown to achieve high performance in the standard settings where the training and test data are i.i.d. However, subtle distributional shifts to the input can degrade their performance significantly. Why is it that models are not robust to such distributional shifts?

Answering the above question is a crucial step towards developing techniques to improve model robustness. Yin et al. [10] show that in naturally occurring data there are many correlations between

the input and target that models can utilize to generalize well. In the image domain, key features such as local textures, colors, shapes, and even invisible high frequency patterns can all be leveraged to achieve i.i.d generalization. We will call these key features *evocative*, because the model can latch on them to get high accuracy. Utilizing evocative features will lead to dramatic reduction in model performance if the same evocative features become corrupted at test time. Hence, relying on evocative features could be a main reason for a model's lack of robustness [10]. Previous works have shown that models trained on natural image datasets overly rely on texture and shape [11, 12, 13]. [11] provided evidence that CNNs rely more heavily on object textures rather than global object shapes as commonly assumed.

In this section, we aim to answer the following two questions:

- *How does applying different Augmentation techniques affect the model reliance on evocative features?* While the significance of evocative features in model performance has been well demonstrated, the impact of standard augmentation techniques on model reliance on evocative features has not been carefully studied. Here, we turn our focus on texture and shape, which are the two most significant evocative features for natural image classification [11].
- *Does the implicit regularization effect of dropout lead the model to rely less on evocative features?* We recognize that dropout can be interpreted as random noise injected to each layer, and data augmentation can be interpreted as noise of a particular type targeted towards specific features. We aim to study the impact of dropout on the model reliance on evocative features.

5.2 Experimental Setup

To better study the impact of data augmentation on evocative features, we give the following definition to classify different augmentation types:

An augmentation A highlights a set of features F1 over another set of features F2 if it applies a transformation to the features in F2 but not the features in F1.

For example, flipping an image would highlight the texture features of an image over the shape features, because the flipped picture reserves its texture but not shape. We can further categorize augmentations based on if they highlight shape or texture. For example, rotation translation and resizing/cropping highlight texture over shape while Gaussian noise, color inversion, and color jittering highlight shape over texture.

To explore the impact of different augmentations to model reliance on evocative features, we train the same model architecture in multiple ways:

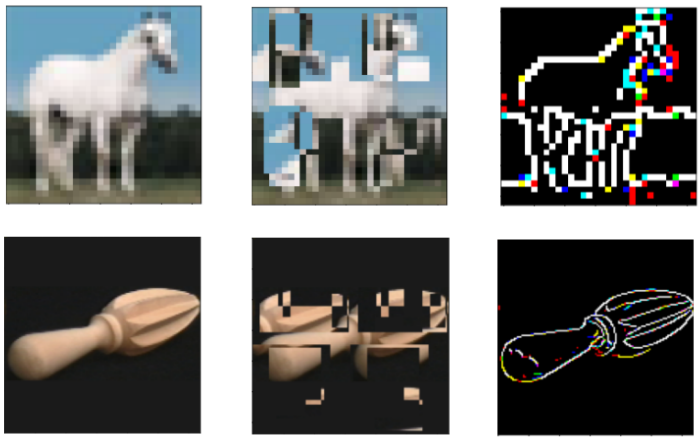


Figure 8: Sample images from CIFAR10 (top) and COIL100 (buttom). left: original image, centre: shuffled image right: edges

- The `Baseline Model` is trained on the original dataset with no augmentations
- The `Texture Model` is trained on the original dataset + augmentations that highlight texture (we pick flipping and translation)
- The `Shape Model` is trained on the original dataset + augmentations that highlight shape (we pick noise and color inversion)
- The `AllAug Model` is trained on the original dataset + augmentations that highlight texture and augmentations that highlight shape (flipping, translation, noise, color inversion)

To measure the reliance of model on texture and shape information, we create two test sets by *removing the texture and shape features* from each image (see 8). We then test each of the models on three datasets:

- `Unaugmented dataset`: this is the original dataset with no additional augmnetations.
- `Shuffled dataset`: to remove the shape information, we *Shuffled* each image by taking multiple pairs of random blocks and switching them. We do this until the shape of the object in the image is hardly recognizable.
- `Edges dataset`: The texture information are removed from the image by extracting the image cutout (edges).

We run our experiments across two dataset; CIFAR10 and COIL100. CIFAR10 is a naturalistic dataset where the variance of both texture and shape information is high across the same class (two pictures among the same class may have different shape and texture information). Each class in the COIL100 dataset consists of 72 images of the same object taken from different angels. Thus, the texture information across the same class is (almost) unchanged. As a result, the model can achieve high accuracy by simply latching on the texture information. Comparatively, the shape information has higher variance across the same class. However, the shape features are still informative enough for a human to classify the image by only looking at the edges/cutout in each image.

5.3 Results and Discussion

Model\Test Set	`Baseline Model`	`Texture Model`	`Shape Model`	`AllAug Model`
`Unaugmented dataset`	0.99 ± 0.00 0.99 ± 0.00	0.99 ± 0.00 0.98 ± 0.00	0.99 ± 0.00 0.99 ± 0.00	0.98 ± 0.01 0.95 ± 0.03
`Shuffled dataset`	0.69 ± 0.02 0.80 ± 0.02	0.76 ± 0.03 0.85 ± 0.06	0.71 ± 0.04 0.78 ± 0.01	0.74 ± 0.05 0.80 ± 0.05
`Edges dataset`	0.06 ± 0.01 0.05 ± 0.01	0.03 ± 0.04 0.03 ± 0.01	0.04 ± 0.00 0.03 ± 0.01	0.03 ± 0.00 0.03 ± 0.03

Table 3: Results on COIL100. The top row in each cell is the accuracy of the model on the corresponding test set. The bottom row is the accuracy of the model trained with dropout.

Model\Test Set	`Baseline Model`	`Texture Model`	`Shape Model`	`AllAug Model`
`Unaugmented dataset`	0.60 ± 0.01 0.75 ± 0.01	0.76 ± 0.01 0.82 ± 0.01	0.66 ± 0.01 0.73 ± 0.01	0.73 ± 0.01 0.77 ± 0.00
`Shuffled dataset`	0.31 ± 0.02 0.30 ± 0.01	0.28 ± 0.02 0.31 ± 0.01	0.24 ± 0.01 0.25 ± 0.02	0.24 ± 0.02 0.25 ± 0.01
`Edges dataset`	0.13 ± 0.00 0.12 ± 0.00	0.12 ± 0.00 0.13 ± 0.00	0.12 ± 0.01 0.12 ± 0.00	0.13 ± 0.01 0.13 ± 0.00

Table 5: Results on CIFAR10. The top row in each cell is the accuracy of the model on the corresponding test set. The bottom row is the accuracy of the model trained with dropout.

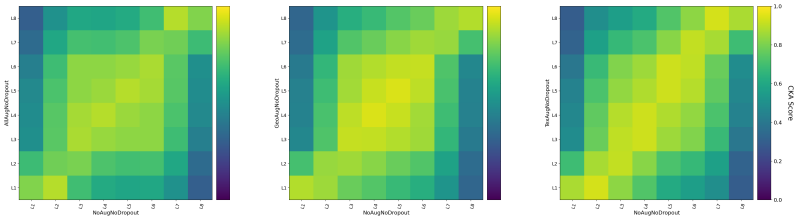
- *How does applying different Augmentation techniques affect the model reliance on evocative features?*

  Texture augmentations result in higher accuracies than shape augmentations: In general, the `Texture Model` performs the best out of all the models across all datasets. By comparing the `Texture Model` and the `Baseline Model`, we can see that training the model with augmentations which highlight texture results in higher accuracy on the `Shuffled dataset` for COIL100. This is as expected, since the texture information is sufficient to get high accuracy on COIL100. On CIFAR10, we witness a significant improvement on the original dataset (from 0.6 to 0.76) when the model is trained with texture augmentation.

  Shape Augmentations do not have a significant impact on model accuracy: We notice that the `Shape Model` performs worse than the `Texture` and the `AllAug` Models on all datasets. This is expected, since latching on the texture information is more rewarding. To our surprise, however, the `Shape Model` performs exactly the same as the other models on the `Edges Dataset`. This suggests that the augmentations highlighting shape information did not make the model rely on shape information. For COIL100, we know that getting high accuracy by simply latching on shape is possible. However, the models prefer to latch on the texture information, since the variance of texture infromation across the same class is lower. Thus, the models hardly learn anything from shape. To further demonstrate this, we note that even adding dropout to the models does not affect the results on the `Edges Dataset`. Thus, we hypothesize that *when two sets of evocative features redundantly predict the correct label, the model only latches on one of the evocative features*. We encourage future research to investigate this idea.

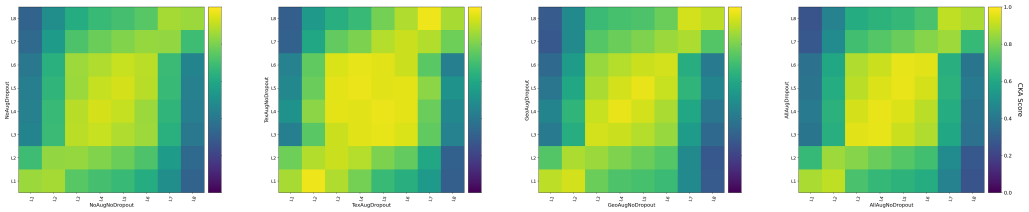
- *Does Applying Data Augmentation force the model to learn radically different representations?*
  To answer this question, we use the CKA metric [14] to measure the similarity between the learned representations of different models. In figure 9, we demonstrate that the models trained with different augmentations learn similar representations to the `Baseline Model` (with CKA similarity index above 0.8 between the same two layers). Each row/column corresponds to one layer in our CNN network. There are 2 layers per VGG block (for 3 VGG blocks in total), as well as two linear layers.

- *Does the implicit regularization effect of dropout lead the model to rely less on evocative features?*
  It is clear that dropout has helped the models achieve better performance (for COIL100, the model was already getting near perfect accuracy and dropout did not affect the accuracy results). How did the model with dropout achieve a performance boost? Did dropout have an impact on the model latching less on the texture information and more on shape?
  The second row of Table 4 indicates that dropout amplified the model reliance on texture information on the COIL100 dataset. On the other hand, dropout did not have any impact on the model learning from the shape information. We hypothesize that dropout does not have an impact on the model relying more or less on a particular feature type. Instead, dropout increases model accuracy via (i) explicit regularization (ii) adding noise to the gradient and implicit regularization. These regularization techniques, however, seem to *almost* have no effect on the model latching on a particular feature type.



(a) `AllAug Model` (b) `Texture Model` (c) `Shape Model`

Figure 9: Linear CKA between the layers of the `Baseline Model` vs the other models. The network architecture is same but the models are trained with different augmentation strategies on CIFAR10



(a) `Baseline Model` (b) `Shape Model` (c) `Texture Model` (d) `AllAug Model`

Figure 10: Linear CKA between layers of each model with and without dropout. All models are trained on CIFAR10.

6 Limitations and Future Work

In this section we discuss some of the limitations of our experiments as well as possible directions for future work. It is worth noting that many of the limitations are in part due computational constraints as well as time constraints imposed by the course with which this report is associated. One such limitation is that we are only able to indirectly observe the regularization and sampling effects of augmentation. Due to our projects integration with the algorithmic efficiency benchmark framework, it was not feasible to obtain training loss estimates throughout training. However, future work should compare the difference between the training and validation losses of both the offline augmented and online augmented models in Experiment 1. This could provide a quantifiable estimate of the magnitude of the regularizing effects in proportion to the sampling effects for the online augmented model. Furthermore, to quantify the change in VC-error between models in Experiment 1, future work should explore the maximum amount of overfitting obtainable by each model. The maximum difference between training and validation error could provide an empirical estimate of VC-error as seen in Equation 6 (Note that the ϵ error term is omitted).

$$\mathcal{O}\left(\left(\frac{|\mathcal{F}|_{VC} - log\delta}{M \times N}\right)^{\alpha}\right) \geq \mathcal{R}(f_{aug}|\mathcal{P}) - \mathcal{R}(f_{aug}|\mathcal{P}_{train}) \approx ValLoss - TrainLoss \quad (6)$$

Additionally, in experiment 1, we only explore the sampling effects in the extreme case of online augmentation where M in Equation 5 is large. Future work should consider training models using offline augmentation *without replacement* and manually vary the factor M to further quantify the sampling effects of data augmentation.

We also note that due to the stochastic nature of online data augmentation, the model continues to see new samples throughout training even after it has passed the threshold where it could have seen all possible variations of an input sample. It is possible that the stochastic introduction of never-before-seen samples has some sort of regularizing effect. One direction for future work would be to explore whether or not this has an effect on model training by comparing an online augmented model trained on MxN samples to an offline augmented model *without replacement* trained with the same MxN samples.

Lastly we draw similarities between the work done by Dao et al [6] to analyze the effects of data augmentation, and the work done by Wei et al [15] to analyze the effects of dropout. We discuss

the former in Section 8. Dao et al explicitly model data augmentation as kernel feature averaging and variance regularization. Conversely Wei et al model dropout as the combination of it's explicit and implicit regularization effects. Both data augmentation and dropout are stochastic processes that improve model generalization. Firstly, we note that the 1st order feature averaging effect of data augmentation is strikingly similar to the explicit regularization effects of dropout. The main difference being the loss with data augmentation is averaged over the distribution of possible transformations whereas the loss with dropout is averaged over the dropout noise. Secondly, we note that the 2nd order variance regularization effects of data augmentation are similar to the implicit regularization effects of dropout. The implicit effects of dropout are a result of the noise added to the gradients throughout training. Although the variance regularization effects of data augmentation are modelled explicitly for a linear kernel classifier, they can also be interpreted as adding noise to the gradients. The main difference being that the gradient noise introduced by data augmentation is selectively added along certain directions corresponding to the features it is regularizing. Future work should explore whether there is a connection between these effects. One could explicitly model the potential implicit regularization effects of data augmentation using noise as well as remove the effects by averaging the loss estimates for a particular sample as seen in [15].

7 Conclusion

In this paper, we viewed model generalizability as an empirical estimation problem. Then, similar to [4], we used this perspective to analyze the effects of data augmentation. However, we go one step further by acknowledging these effects can be categorized into two distinct groups or pathways, which we refer to as regularizing and sampling effects for simplicity.

In Experiment 1 (Section 3) we validate these categories and explore their differences. First we saw that data augmentation can improve model performance even without increasing the number of training samples. This shows that the regularizing effects of data augmentation can if fact be separated from the sampling effects as hypothesized in Section 2.2. We then discussed how the regularizing effects might change the loss landscape altering the training dynamics and convergence rate of the augmented model. Lastly, based on empirical evidence, we discussed how the sampling effects of data augmentation might reduce overfitting by lowering an empirical bound that prevents other models from achieving better performance.

In Experiment 2 we take a different approach to explore the sampling effects of data augmentation. We modify the amount of original data, denoted by variable N in Equation 5, as opposed the the amount of synthetic data, denoted by the variable M. We show that the improvement due to data augmentation is dependent on the amount of original data, and notably lesser for particularly small datasets. We also observe progressively reduced overfitting as the number of unique samples increases, which agrees with our findings on the sampling effects of data augmentation in Experiment 1.

In Experiment 3 we focus on the regularizing effects of data augmentation. We introduce the concept of *evocative features* and explore how different types of data augmentation selectively regularize different features. We show, for CIFAR10 and COIL100 datasets, that augmentations that highlight texture by adding variance to shape and position result in better performance than augmentations that highlight shape by adding variance to texture features. We also find that dropout does not seem affect model reliance on evocative features in order to improve model performance.

In conclusion we've proposed a new perspective from which to explore the effects of data augmentation and conducted preliminary experiments with this perspective in mind. However we recognize that further validation and exploration is required, and provide several opportunities for future work in Section 6.

8 Related Works

Rethinking the Distribution Gap between Clean and Augmented Data [4] In this paper the authors make a distinction between intensive data augmentation, which causes a significant distributional shift between $\mathcal{P}_{train}$ and $\mathcal{P}_{aug}$, and non-intensive data augmentation, which does not. This analysis forms the basis of our motivation in Section 2.2. The authors argue there is a trade-off between the amount of augmented data and the distribution gap. They argue this distribution gap is less important at the beginning of training, showing that intensive data augmentation regularizes the

weights of minor features. This forces the model to focus on major features, keeping the direction of convergence consistent and reducing local minima. However, they hypothesize this distribution gap is significant at the end of training when near an optimum. Hence they propose refined data augmentation, wherein a model is fine-tuned with non-intensive data augmentation at the end of training. They show that this approach improves model performance, however we note that the effect size is small and the authors do not provide confidence intervals.

A Kernel Theory of Modern Data Augmentation [6] In this paper the authors first motivate the use of kernels in their analysis by showing that augmentation modelled as a Markov process, when combined with a *k*-nearest neighbour classifier, asymptotically acts as a kernel classifier. Then, using linear kernel classifiers, they explicitly model augmentation using the first and second order Taylor approximations of the objective function. In the first order approximation, they show that training a linear classifier with kernel K, augmentation transformation T and feature map ϕ is equivalent to having a kernel TKT^T and feature map $E_{t\sim T(x)}[\phi(t)]$. That is to say that augmentation has a feature averaging effect. In the second order approximation, they show that augmentation regularizes features that are changed significantly by the augmentation transformation T. They validate these findings empirically by showing that they can nearly match regular augmentation by modelling the 1st and 2nd order effects explicitly in the objective function. We note however that they only validate their approximation for a fixed number of steps, and it is not shown whether or not the models used for comparison reached an optimum. Therefore it is unclear whether or not their approximation remains accurate throughout all stages of training.

References

[1] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, July 2019.

[2] Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv:1806.03852 [cs]*, November 2020. arXiv: 1806.03852.

[3] 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 2019. arXiv: 1805.09501.

[4] 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 2019. arXiv: 1909.09148.

[5] Vladmir N. Vapnik. *Statistical learning theory*. Springer-Verlag New York, Inc., 1998.

[6] 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. ISSN: 2640-3498.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[8] Alex Hernández-García and Peter König. Further advantages of data augmentation on convolutional neural networks. *arXiv:1906.11052 [cs]*, 11139:95–103, 2018. arXiv: 1906.11052.

[9] Sen Wu, Hongyang Zhang, Gregory Valiant, and Christopher Re. On the Generalization Effects of Linear Transformations in Data Augmentation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10410–10420. PMLR, November 2020. ISSN: 2640-3498.

[10] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[12] Samuel Ritter, David Barrett, Adam Santoro, and Matt Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. 06 2017.

[13] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019.

[14] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019.

[15] Colin Wei, Sham Kakade, and Tengyu Ma. The Implicit and Explicit Regularization Effects of Dropout. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10181–10192. PMLR, November 2020. ISSN: 2640-3498.

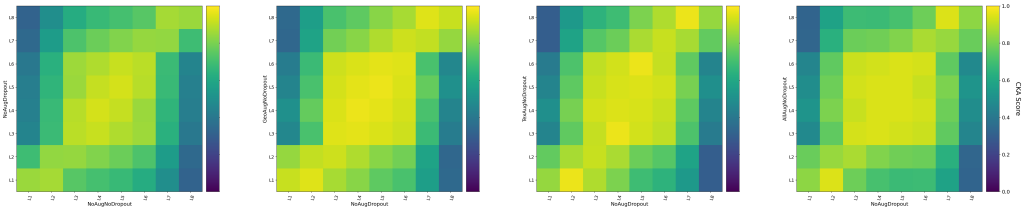
Appendix

9 Additional Results for Experiment 2

	Baseline		Augmented	
Subset of Original Data (%)	Loss	Accuracy (%)	Loss	Accuracy (%)
10	1.4521	48.78	1.2463	57.15
20	1.3194	53.82	1.1113	62.78
30	1.2559	56.32	0.9372	69.08
40	1.1555	59.75	0.8299	72.28
50	1.1015	62.30	0.7740	74.86
60	1.0774	63.22	0.7263	76.28
70	0.9908	66.78	0.6980	77.22
80	0.9839	67.27	0.6671	78.40
90	0.9488	68.07	0.6252	79.62
100	0.9031	69.88	0.6080	80.57

Table 6: Full set of experimental results for Experiment 2. Validation loss and accuracy reported when model reaches minimum validation loss.

10 Additional CKA Results for Experiment 3



(a) `Baseline Model + dropout` (b) `Texture Model + dropout` (c) `Shape Model + dropout` (d) `AllAug Model + dropout`

Figure 11: Linear CKA between layers of the `Baseline Model`, and networks trained with different augmentation strategies with dropout. All models are trained on CIFAR10.