On the Generalization Effects of Linear Transformations in Data Augmentation

Sen Wu [*1] **Hongyang R. Zhang** [*2] **Gregory Valiant** [1] **Christopher Ré** [1]

Abstract

Data augmentation is a powerful technique to improve performance in applications such as image and text classification tasks. Yet, there is little rigorous understanding of why and how various augmentations work. In this work, we consider a family of linear transformations and study their effects on the ridge estimator in an over-parametrized linear regression setting. First, we show that transformations which preserve the labels of the data can improve estimation by enlarging the span of the training data. Second, we show that transformations which mix data can improve estimation by playing a regularization effect. Finally, we validate our theoretical insights on MNIST. Based on the insights, we propose an augmentation scheme that searches over the space of transformations by how *uncertain* the model is about the transformed data. We validate our proposed scheme on image and text datasets. For example, our method outperforms RandAugment by 1.24% on CIFAR-100 using Wide-ResNet-28-10. Furthermore, we achieve comparable accuracy to the SoTA Adversarial AutoAugment on CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets.

1. Introduction

Data augmentation refers to the technique of enlarging the training dataset through pre-defined transformation functions. By searching over a (possibly large) space of transformations through reinforcement learning based techniques, augmentation schemes including AutoAugment (Cubuk et al., 2018) and TANDA (Ratner et al., 2017) have shown remarkable gains over various models on image classification tasks. Recent work has proposed random sampling (Cubuk et al., 2019) and Bayesian optimization (Hataya et al., 2019) to reduce the search cost, since RL-based techniques are computationally expensive. Despite the rapid progress of these transformation search methods, precisely understanding their benefits remains a mystery because of a lack of analytic tools. In this work, we study when and why applying a family of linear transformations helps from a theoretical perspective. Building on the theory, we develop methods to improve the efficiency of transformation search procedures.

A major challenge to understand the theory behind data augmentation is how to model the large variety of transformations used in practice in an analytically tractable framework. The folklore wisdom behind data augmentation is that adding more labeled data improves generalization, i.e. the performance of the trained model on test data (Shorten & Khoshgoftaar, 2019). Clearly, the generalization effect of an augmentation scheme depends on how it transforms the data. Previous work has analyzed the effect of Gaussian augmentation (Rajput et al., 2019) and feature averaging effect of adding transformed data (Dao et al., 2019; Chen et al., 2019). However, there is still a large gap between the transformations studied in these works and the ones commonly used in augmentation schemes.

In this work, we consider linear transformations which represent a large family of image transformations. We consider three categories: (i) Label-invariant (base) transformations such as rotation and horizontal flip; (ii) Label-mixing transformations including mixup (Zhang et al., 2017; Inoue, 2018), which produces new data by randomly mixing the features of two data points (e.g. a cat and a dog) – the labels are also mixed; (iii) Compositions of (label-invariant) transformations such as random cropping and rotating followed by horizontal flipping.

To gain insight into the effects of linear transformations, we consider a conceptually simple over-parametrized model proposed in Bartlett et al. (2019); Xie et al. (2020); Hastie et al. (2019) that captures the need to add more data as in image settings. Suppose we are given n training data points $x_1, \dots, x_n \in \mathbb{R}^p$ as $X \in \mathbb{R}^{n \times p}$ with labels $Y \in \mathbb{R}^n$. In this setting, the labels obey the linear model under ground truth parameters $\beta \in \mathbb{R}^p$, i.e. $Y = X\beta + \varepsilon$, where $\varepsilon \in \mathbb{R}^n$ denotes i.i.d. random noise with mean zero and variance σ^2.

Importantly, we assume that $p > n$, hence the span of the training data points does not include the entire space

[*] Equal contribution [1] Department of Computer Science, Stanford University [2] Department of Statistics, The Wharton School, University of Pennsylvania. Correspondence to: all authors <{senwu, hongyang, gvaliant, chrismre}@cs.stanford.edu>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

Transformations	Example	Improvement
Label-invariant	Rotate (F)	$\frac{(\beta^\top P_X^\perp Fx)^2}{n}$
Label-mixing	Mixup	$\frac{\lVert X\beta\rVert^2}{n^2}$
Composition	Rotate and flip $(F_1 F_2)$	$\frac{(\beta^\top P_X^\perp F_1 F_2 x)^2}{n}$

Table 1. Illustrative examples of our theoretical results. For label-invariant transformations and their compositions, they can reduce the estimation error at a rate proportional to the added information. Label-mixing reduces estimation error through a regularization effect.

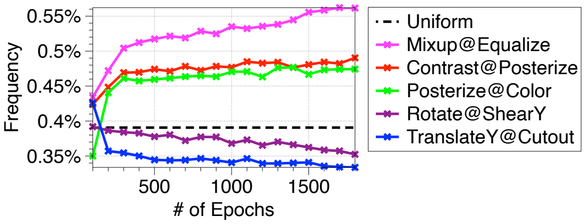

Figure 1. Our method learns and reduces the frequencies of the better performing transformations during training. Base model: PyramidNet+ShakeDrop (Han et al., 2017; Yamada et al., 2018). Dataset: CIFAR-10. See Section 5.3 for the details.

of $\mathbb{R}^p$. We consider the ridge estimator $\hat{\beta}$ with a fixed ℓ_2 regularization parameter and measure the estimation error of the ridge estimator by its distance to β.

Our first insight is that *label-invariant transformations* can *add new information* to the training data. We use our theoretical setup described above to present a precise statement. For a data point (x, y), a label-invariant transformation $F \in \mathbb{R}^{p \times p}$ in the regression setting produces a new data point Fx with label y. We show that by adding $P_X^\perp Fx$ which is outside the span of the training data, we reduce the estimation error of the ridge estimator at a rate proportional to $(\beta^\top P_X^\perp Fx)^2/n$ (see Theorem 3.1 for the result). Here $P_X^\perp$ denotes the projection to the orthogonal space of $X^\top X$. In Section 4, we validate the insight for classification settings on MNIST by showing that a label-invariant transformation can indeed add new information by reducing the bias (intrinsic error score) of the model.

Our second insight is that *label-mixing transformations* can *provide a regularization effect*. In the theoretical setup, given two data points (x_1, y_1) and (x_2, y_2), the mixup transformation with parameter α sampled from the Beta distribution produces $\alpha x_1 + (1-\alpha)x_2$ with label $\alpha y_1 + (1-\alpha)y_2$ (Zhang et al., 2017). Interestingly, mixup does not *add new information* since the mixed data lies in the span of the training data. However, we show that mixup plays a regularization effect through shrinking the training data relative to the ℓ_2 regularization. The final result is that adding the mixup sample reduces estimation error by $\Theta(\lVert X\beta\rVert^2/n^2)$ (see Theorem 3.3 for the result). In Section 4, we validate the insight on MNIST by showing that mixing same-class digits can reduce the variance (instability) of the model.

Finally, for *compositions of label-invariant transformations*, we can show their effect for adding new information as a corollary of the base case (see Corollary 3.4 for details and we provide the validation on MNIST in Section 4). We provide an illustration of the results in Table 1.

Algorithmic results. Building on our theory, we propose an uncertainty-based random sampling scheme which, among the transformed data points, picks those with the highest loss, i.e. "providing the most information". We find that there is a large variance among the performance of different transformations and their compositions. Unlike RandAugment (Cubuk et al., 2019), which averages the effect of all transformations, the idea behind our sampling scheme is to better select the transformations with strong performance.

We show that our proposed scheme applies to a wide range of datasets and models. First, our sampling scheme achieves higher accuracy by finding more useful transformations compared to RandAugment on three different CNN architectures. For example, our method outperforms RandAugment by 0.59% on CIFAR-10 and 1.24% on CIFAR-100 using Wide-ResNet-28-10, and 1.54% on ImageNet using ResNet-50. Figure 1 shows the *key insight* behind our proposed method. We compare the frequencies of transformations (cf. Section 5 for their descriptions) sampled by our method. As the training procedure progresses, our method gradually learns transformations providing new information (e.g. Rotate followed by ShearY) and *reduces* their frequencies. On the other hand, the frequencies of transformations such as Mixup followed by Equalize *increase* because they produce samples with large errors that the model cannot learn.

Second, we achieve similar test accuracy on CIFAR-10 and CIFAR-100 compared to the state-of-the-art Adversarial AutoAugment (Zhang et al., 2020). By contrast, our scheme is conceptually simpler and computationally more efficient; Since our scheme does not require training an additional adversarial network, the training cost reduces by at least 5x. By further enlarging number of augmented samples by 4 times, we achieve test accuracy 85.02% on CIFAR-100, which is higher than Adversarial AutoAugment by 0.49%. Finally, as an extension, we apply our scheme on a sentiment analysis task and observe improved performance compared to the previous work of Xie et al. (2019).

Notations. We use the big-O notation $f(n) \le O(g(n))$ to indicate that $f(n) \le C \cdot g(n)$ for a fixed constant C and large enough n. We use $f(n) \lesssim g(n)$ to denote that $f(n) \le O(g(n))$. For a matrix $X \in \mathbb{R}^{d_1 \times d_2}$, let $X^\dagger$ denote the Moore-Penrose psuedoinverse of X.

2. Preliminaries

Recall that $X = [x_1^\top, \dots, x_n^\top] \in \mathbb{R}^{n \times p}$ denotes the training data, where $x_i \in \mathbb{R}^p$ for $1 \le i \le n$. Let $\mathrm{Id}_p \in \mathbb{R}^{p \times p}$ denote the identity matrix. Let P_X denote the projection matrix onto the row space of X. Let $P_X^\perp = \mathrm{Id}_p - P_X$ denote the projection operator which is orthogonal to P_X. The ridge estimator with parameter $\lambda < 1$ is given by

$$\hat{\beta}(X, Y) = (X^\top X + n\lambda \,\mathrm{Id})^{-1} X^\top Y,$$

which arises from solving the mean squared loss

$$\frac{1}{2n} \min_{w \in \mathbb{R}^p} \|Xw - Y\|_F^2 + \frac{\lambda}{2}\|w\|^2.$$

We use $\hat{\beta}^F$ to denote the ridge estimator when we augment (X, Y) using a transformation function F. The estimation error of $\hat{\beta}$ is given by $e(\hat{\beta}) := \mathbb{E}_\varepsilon \left[\|\hat{\beta} - \beta\|^2\right]$. Next we define the label-invariance property for regression settings.

Definition 2.1 (Label-invariance). *For a matrix $F \in \mathbb{R}^{p \times p}$, we say that F is label-invariant over $\mathcal{X} \subseteq \mathbb{R}^p$ for $\beta \in \mathbb{R}^p$ if*

$$x^\top \beta = (Fx)^\top \beta, \text{ for any } x \in \mathcal{X}.$$

As an example, consider a 3-D setting where $\mathcal{X} = \{(a, b, 0) : \forall a, b \in \mathbb{R}\}$. Let $\beta = (1, -1/2, 1/2)^\top$ and

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\frac{\pi}{2} & \sin\frac{\pi}{2} \\ 0 & -\sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{pmatrix}.$$

Then $(\mathrm{Id} - F^\top)\beta = (0, 0, 1)$ is orthogonal to $\mathcal{X}$. Hence F is a label-preserving rotation with degree $\frac{\pi}{2}$ over $\mathcal{X}$ for β.

In addition to rotations and mixup which we have described, linear transformations are capable of modeling many image transformations. We list several examples below.

Horizontal flip. The horizontal flip of a vector along its center can be written as an orthonormal transformation where

$$F = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ 1 & 0 & \dots & 0 \end{pmatrix}.$$

Additive composition. For two transformations F_1 and F_2, their additive composition gives $x^{\text{aug}} = F_1 x + F_2 x$. For example, *changing the color of an image and adding Gaussian noise* is an additive composition with $F_1 x$ being a color transformation and $F_2 x$ being a Gaussian perturbation.

Multiplicative composition. In this case, $x^{\text{aug}} = F_1 F_2 x$. For example, a *rotation followed by a cutout* of an image is a multiplicative composition with F_2 being a rotation matrix and F_1 being a matrix which zeros out certain regions of x.

3. Analyzing the Effects of Transformation Functions in an Over-parametrized Model

How should we think about the effects of applying a transformation? Suppose we have an estimator $\hat{\beta}$ for a linear model $\beta \in \mathbb{R}^p$. The bias-variance decomposition of $\hat{\beta}$ is

$$e(\hat{\beta}) = \underbrace{\left\|\mathop{\mathbb{E}}_{\varepsilon}\left[\hat{\beta}\right] - \beta\right\|^2}_{\text{bias}} + \underbrace{\left\|\hat{\beta} - \mathop{\mathbb{E}}_{\varepsilon}\left[\hat{\beta}\right]\right\|^2}_{\text{variance}} \tag{1}$$

In the context of data augmentation, we show the following two effects from applying a transformation.

Adding new information. The bias part measures the error of $\hat{\beta}$ after taking the expectation of ε in $\hat{\beta}$. Intuitively, the bias part measures the intrinsic error of the model after taking into account the randomness which is present in $\hat{\beta}$. A transformation may improve the bias part if x^{aug} is outside P_X. We formalize the intuition in Section 3.1.

Regularization. Without adding new information, a transformation may still reduce $e(\hat{\beta})$ by playing a regularization effect. For example with mixup, x^{aug} is in P_X. Hence adding x^{aug} does not add new information to the training data. However, the mixup sample reweights the training data and the ℓ_2 regularization term in the ridge estimator. We quantify the effect in Section 3.2.

3.1. Label-Invariant Transformations

We quantify the effect of label-invariant transformations which add new information to the training data.

Example. Given a training data point (x, y), let $(x^{\text{aug}}, y^{\text{aug}})$ denote the augmented data where $y^{\text{aug}} = y$. Intuitively, based on our training data (X, Y), we can infer the label of any data point within P_X (e.g. when σ is sufficiently small). If x^{aug} satisfies that $P_X x^{\text{aug}} = 0$, then add $(x^{\text{aug}}, y^{\text{aug}})$ does not provide new information. On the other hand if $P_X^\perp x^{\text{aug}} \neq 0$, then adding x^{aug} expands the subspace over which we can obtain accurate estimation. Moreover, the added direction corresponds to $P_X^\perp x^{\text{aug}}$. Meanwhile, since $y^{\text{aug}} = y$, it contains a noise part which is correlated with ε. Hence the variance of $\hat{\beta}$ may increase.

To derive the result, we require a technical twist: instead of adding the transformed data Fx directly, we are going to add its projection onto $P_X^\perp$. This is without loss of generality since we can easily infer the label of $P_X^\perp F x$. Let $\hat{\beta}^F$ denote the ridge estimator after adding the augmented data to the (X, Y). Our result is stated as follows.

Theorem 3.1. *Suppose we are given a set of n covariates $X \in \mathbb{R}^{n \times p}$ with labels $Y = X\beta + \varepsilon$, where $\beta \in \mathbb{R}^p$ and $\varepsilon \in \mathbb{R}^n$ has mean 0 and variance σ^2. Let $F \in \mathbb{R}^{p \times p}$ be a label-invariant transformation over X for β. Let (x, y) be a data point from (X, Y).*

Algorithm 1 Uncertainty-based sampling of transformations

```
1: **Input.** a batch of $B$ data point $(x_1, y_1), (x_2, y_2), \ldots, (x_B, y_B)$
   **Require.** $K$ transformations $F_1, F_2, \ldots, F_K$, the current model $\hat{\beta}$ and the associated loss $l_{\hat{\beta}}$. Default transformations $G_1, \ldots, G_H$.
   **Param.** $L$: number of composition steps; $C$: number of augmented data per input data; $S$: number of selected data points used for training.
   **Return:** a list of $B \cdot S$ transformed data $T$.
2: **for** $i = 1, \ldots, B$ **do**
3:     **for** $k = 1, \ldots, C$ **do**
4:         Let $F_{j_1}, \ldots, F_{j_L}$ be $L$ transformations sampled uniformly at random without replacement from $F_1, \ldots, F_K$.
5:         Compute $x_k^{\text{aug}}$ and $y_k^{\text{aug}}$ by applying $F_{j_1}, \ldots, F_{j_L}$ and the default ones $G_1, \ldots, G_H$ sequentially on $(x_i, y_i)$.
6:         Infer the loss of $(x_k^{\text{aug}}, y_k^{\text{aug}})$ as $l^k = l_{\hat{\beta}}(x_k^{\text{aug}}, y_k^{\text{aug}})$.
7:     **end for**
8:     Choose the $S$ data points from $\{x_k^{\text{aug}}, y_k^{\text{aug}}\}_{k=1}^{C}$ that have the highest losses $l^k$ and add them to $T$.
9: **end for**
```

Let $z = P_X^{\perp} F x$ and $y^{\text{aug}} = y - \text{Diag}\left[(X^{\top})^{\dagger} F x\right] Y$. Suppose that we augment (X, Y) with (z, y^{aug}). Then, we have that

$$e(\hat{\beta}) - e(\hat{\beta}^F) \geq \frac{(2\lambda(n+1) - (1-2\lambda)\|z\|^2)}{\lambda^2(n+1+\|z\|^2)^2} \cdot \langle z, \beta \rangle^2 - \frac{(2 + 2\frac{\|P_X F x\|^2}{\mu_{\min}(X)^2})}{\lambda^2(n+1)^2} \cdot \sigma^2 \|z\|^2. \quad (2)$$

Moreover, when $\|z\|^2 = o(n)$ and $\frac{\langle z, \beta \rangle^2}{\|z\|^2} \geq \log n (1 + \frac{\|P_X F x\|^2}{\mu_{\min}(X)^2}) \frac{\sigma^2}{\lambda n}$ (including $\sigma = 0$), we have

$$0 \leq e(\hat{\beta}) - e(\hat{\beta}^F) - (2 + o(1)) \frac{\langle z, \beta \rangle^2}{\lambda n} \leq \frac{\text{poly}(\gamma/\lambda)}{n^2}. \quad (3)$$

In the above, $\mu_{\min}(X)$ denotes the smallest singular value of X. For a vector v, $\text{Diag}\,[v] \in \mathbb{R}^{d \times d}$ denotes a diagonal matrix with the i-th diagonal entry being the i-th entry of v. poly(γ/λ) denotes a polynomial of γ/λ.

Theorem 3.1 shows that the reduction of estimation error scales with $\langle z, \beta \rangle^2$, the correlation between the new signal and the true model. Intuitively, as long as n is large enough, then equation (3) will hold. The proof is by carefully comparing the bias and variance of $\hat{\beta}$ and $\hat{\beta}^{\text{aug}}$ after adding the augmented data point. On the other hand, we remark that adding Fx directly into X does not always reduce $e(\hat{\beta})$, even when $\langle z, \beta \rangle^2 = \Theta(1)$ (cf. (Xie et al., 2020)).

Another remark is that for augmenting a sequence of data points, one can repeated apply Theorem 3.1 to get the result. We leave the details to Appendix A.1.

Connection to augmentation sampling schemes. We derive a corollary for the idea of random sampling used in RandAugment. Let $\{F_i\}_{i=1}^K$ be a set of K label-invariant transformations. We consider the effect of randomly sampling a transformation from $\{F_i\}_{i=1}^K$.

Corollary 3.2. *In the setting of Theorem 3.1, let $\{F_i\}_{i=1}^K$ be K label-invariant transformations. For a data point (x, y) from (X, Y) and $i = 1, \ldots, K$, let $z_i = P_X^{\perp} F_i x$ and $y_i^{\text{aug}} = y - \text{Diag}\left[(X^{\top})^{\dagger} F_i x\right] Y$.*

Suppose that (z, y^{aug}) is chosen uniformly at random from $\{z_i, y_i^{\text{aug}}\}_{i=1}^K$. Then we have that

$$\mathop{\mathbb{E}}_{z, y^{\text{aug}}} \left[e(\hat{\beta}) - e(\hat{\beta}^{\text{unif}\{F_i\}_{i=1}^K}) \right] = \frac{2 + o(1)}{K} \left(\sum_{i=1}^{K} \frac{\langle z_i, \beta \rangle^2}{\lambda n} \right) + \frac{\text{poly}(\gamma/\lambda)}{n^2}.$$

The proof follows directly from Theorem 3.1. Corollary 3.2 implies that the effect of random sampling is simply an average over all the transformations. However, if there is a large variance among the effects of the K transformations, random sampling could be sub-optimal.

Our proposed scheme: uncertainty-based sampling. Our idea is to use the sampled transformations more efficiently via an *uncertainty-based* sampling scheme. For each data point, we randomly sample C (compositions of) transformations. We pick the ones with the highest losses after applying the transformation. This is consistent with the intuition of Theorem 3.1. The larger $\langle z, \beta \rangle^2$, the higher the loss of $(x^{\text{aug}}, y^{\text{aug}})$ would be under $\hat{\beta}$.

Algorithm 1 describes our procedure in detail. In Line 4-6, we compute the losses of C augmented data points. In Line 8, we select the S data points with the highest losses for training. For each batch, the algorithm returns $S\dot{B}$ augmented samples.

3.2. Label-Mixing Transformations: Mixup

We show that mixup plays a regularization effect through reweighting the training data and the ℓ_2 regularization term.

Specifically, we analyze the following procedure. Let $\alpha \in [0, 1]$ be sampled from a Beta distribution with fixed parameters (Zhang et al., 2017). Let (x_i, y_i) and (x_j, y_j) be two data points selected uniformly at random from the training data. We add the mixup sample $(x^{\text{aug}}, y^{\text{aug}})$ into (X, Y), with $x^{\text{aug}} = \alpha x_i + (1-\alpha)x_j$ and $y^{\text{aug}} = \alpha y_i + (1-\alpha)y_j$.

We illustrate that adding the mixup sample is akin to shrinking the training data relative to the regularization term. Assuming that $\sum_{i=1}^n x_i = 0$, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x^{\text{aug}}}\left[x^{\text{aug}}{x^{\text{aug}}}^\top\right] = \frac{(1-2\alpha)^2}{n}\cdot X^\top X\\
\Rightarrow &\mathop{\mathbb{E}}_{x^{\text{aug}}}\left[X^\top X + x^{\text{aug}}{x^{\text{aug}}}^\top + (n+1)\lambda\,\mathrm{Id}\right]\\
&= (1 + \frac{(1-2\alpha)^2}{n})X^\top X + (n+1)\lambda\,\mathrm{Id},
\end{aligned}$$

Hence the mixup sample shrinks the $X^\top X$ term relative to the ℓ_2 regularization term in $\hat{\beta}^{\text{aug}}$! Below we describe our result formally.

Theorem 3.3. *Let $\gamma > 1$ be a fixed constant which does not grow with n. Let $\{x_k\}_{k=1}^n$ be n training samples which satisfy that $\sum_{k=1}^n x_k = 0$ and $\|x_k\| \le \gamma$, for all $1 \le k \le n$. Assume that n is large enough (e.g. $n \gtrsim \frac{(\gamma+\lambda)^3}{\lambda^4}\max(\|\beta\|^2\gamma^4, \sigma^2\gamma^2)$). In expectation over the randomness of α, x_i, x_j, we have that*

$$\mathop{\mathbb{E}}_{\alpha, x_i, x_j}\left[e(\hat{\beta}) - e(\hat{\beta}^{\text{mixup}})\right] \gtrsim \frac{\lambda^2}{\gamma^3}\frac{\|X\beta\|^2}{n^2}.$$

We remark that the assumption that $\sum_{i=1}^n x_i = 0$ is indeed satisfied in image classification settings. This is because a normalization step, which normalizes the mean of every RGB channel to be zero, is applied on all the images. The intuition behind the proof of Theorem 3.3 is that the mixup sample shrinks the training data, which reduces the bias of the estimator. The proof can be found in Appendix A.2.

3.3. Compositions of Label-Invariant Transformations

Our theory can also be applied to quantify the amount of new information added by compositions of transformations. We first describe an example to show that taking compositions expands the search space of transformation functions.

Example. Consider two transformations F_1 and F_2, e.g. a rotation and a horizontal flip. Suppose we are interested in finding a transformed sample $(x^{\text{aug}}, y^{\text{aug}})$ such that adding the sample reduces the estimation error of $\hat{\beta}^{\text{aug}}$ the most. With additive compositions, the search space for x^{aug} becomes

$$\{Fx : x \in \mathcal{X}, F \in \{F_1, F_2, F_1 + F_2\}\},$$

which is strictly a superset compared to using F_1 and F_2.

Based on Theorem 3.1, we can derive a simple corollary which quantifies the incremental benefit of additively composing a new transformation.

Corollary 3.4. *In the setting of Theorem 3.1, let F_1, F_2 be two label-invariant transformations. For a data point (x, y) from (X, Y) and $i \in \{1, 2\}$, let $z_i = P_X^\perp F_i x$ and $y_i^{\text{aug}} = y - \mathrm{Diag}\left[(X^\top)^\dagger F_i x\right] Y$. The benefit of composing F_2 with F_1 is given as follows*

$$\begin{aligned}
e(\hat{\beta}^{F_1}) - e(\hat{\beta}^{F_1+F_2}) = \;&(2 + o(1))\frac{\langle z_1, \beta\rangle^2 - \langle z_1 + z_2, \beta\rangle^2}{\lambda n}\\
&+ \frac{\mathrm{poly}(\gamma/\lambda)}{n^2}.
\end{aligned}$$

Corollary 3.4 implies that the effect of composing F_2 with F_1 may either be better or worse than applying F_1. This is consistent with our experimental observation (described in Section 4). We defer the proof of Corollary 3.4 to Appendix A.3. We remark that the example and the corollary also apply to multiplicative compositions.

4. Measuring the Effects of Transformation Functions

We validate the theoretical insights from Section 3 on MNIST (LeCun et al., 1998). To extend our results from the regression setting to the classification setting, we propose two metrics that correspond to the bias and the variance of a linear model. Our idea is to decompose the average prediction accuracy (over all test samples) into two parts similar to equation (1), including an *intrinsic error score* which is deterministic and an *instability score* which varies because of randomness.

We show three claims: i) Label-invariant transformations such as rotations can add new information by reducing the intrinsic error score. ii) As we increase the fraction of same-class mixup digits, the instability score decreases. iii) Composing multiple transformations can either increase the accuracy (and intrinsic error score) or decrease it. Further, we show how to select a core set of transformations.

Metrics. We train k independent samples of multi-layer perceptron with hidden layer dimension 100. Let $\hat{\beta}_i$ denote the predictor of the i-th sample, for $1 \le i \le k$. For each data point x, let $M(x)$ denote the majority label in $\{\hat{\beta}_i\}_{i=1}^k$ (we break ties randomly). Clearly, the majority label is the best estimator one could get, given the k independent predictors, without extra information. Then, we define the *intrinsic error score* as

$$\frac{1}{n}\cdot\left(\sum_{(x,y)\in(X,Y)} \mathbb{1}\{M(x) \neq y\}\right). \tag{4}$$

We define the *instability score* as

$$\frac{1}{n}\cdot\left(\sum_{(x,y)\in(X,Y)}\left(\frac{1}{k}\cdot\sum_{i=1}^k \mathbb{1}\{\hat{\beta}_i(x) \neq M(x)\}\right)\right). \tag{5}$$

	Avg. Acc.	Error Score	Instab. Score
Baseline	98.08%	1.52%	0.95%
Cutout	98.31%	1.43%	0.86%
RandCrop	98.61%	**1.01%**	0.88%
Rotation	**98.65%**	1.08%	**0.77%**

Table 2. Measuring the intrinsic error and instability scores of individual transformations on MNIST. The three transformations all reduce the intrinsic error score compared to the baseline.

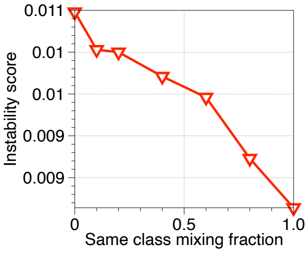

Figure 2. The instability score decreases as we increase the fraction of same-class mixup digits on MNIST.

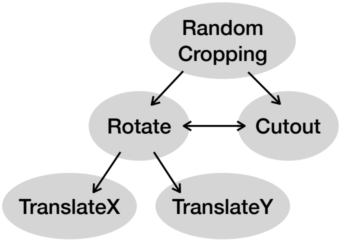

Figure 3. Visualizing which transformations are beneficial for each other. On MNIST, the translations do not provide additional benefit beyond the rest three transformations.

Compared to equation (1), the first score corresponds to the bias of $\hat{\beta}$, which can change if we add new information to the training data. The second score corresponds to the variance of $\hat{\beta}$, which measures the stability of the predicted outcomes in the classification setting. For the experiments we sample 9 random seeds, i.e. set $k = 9$.

Label-invariant transformations. Recall that Section 3.1 shows that label-invariant transformations can reduce the bias of a linear model by adding new information to the training data. Correspondingly, our hypothesis is that the intrinsic error score should decrease after augmenting the training dataset with suitable transformations.

Table 2 shows the result for applying rotation, random cropping and the cutout (i.e. cutting out a piece of the image) respectively. The baseline corresponds to using no transformations. We observe that the three transformations all reduce the model error score and the instability score, which confirms our hypothesis. There are also two second-order effects: i) A higher instability score hurts the average predication accuracy of random cropping compared to rotation. ii) A higher intrinsic error score hurts the average prediction accuracy of cutout compared to random cropping.

To further validate that rotated data "adds new information", we show that the intrinsic error score decreases by correcting data points that are "mostly incorrectly" predicted by the baseline model. Among the data points that are predicted correctly after applying the rotation but are wrong on the baseline model, the average accuracy of these data points on the baseline model is 25%. In other words, for 75% of the time, the baseline model makes the wrong prediction for these data. We leave the details to Appendix B.

Label-mixing transformations. Section 3.2 shows that in our regression setting, adding a mixup sample has a regularization effect. What does it imply for classification problems? We note that our regression setting comprises a single parameter class β. For MNIST, there are 10 different classes of digits. Therefore, a plausible hypothesis following our result is that mixing same-class digits has a regularization effect.

We conduct the following experiment to verify the hypothesis. We vary the fraction of mixup data from mixing same-class vs. different-class digits. We expect that as we increase the same class mixing fraction, the instability score decreases. The results of figure 2 confirm our hypothesis. We also observe that mixing different class images behaves differently compared to mixing same class images. The details are left to Appendix D.

Compositions of transformations. Recall that Section 3.3 shows the effect of composing two transformations may be positive or negative. We describe examples to confirm the claim. By composing rotation with random cropping, we observe an accuracy of 98.72% with intrinsic error score 0.86%, which are both lower than rotation and random cropping individually. A negative example is that composing translating the X and Y coordinate (acc. 98.0%) is worse than translating X (acc. 98.38%) or Y (acc. 98.19%) individually. Same for the other two scores.

Identifying core transformations. As an in-depth study, we select five transformation including rotation, random cropping, cutout, translation of the x-axis and the y-axis. These transformations can all change the geometric position of the digits in the image. Our goal is to identify a subset of core transformations from the five transformations. For example, since the rotation transformation implicitly changes the x and the y axis of the image, a natural question is whether composing translations with rotation helps or not.

To visualize the results, we construct a directed graph with the five transformations. We compose two transformations to measure their model error scores. An edge from transformation A to transformation B means that applying B after A reduces the model error score by over 10%. Figure 3 shows the results. We observe that rotation, random cropping and cutout can all help some other transformations, whereas the translations do not provide an additional benefit for the rest.

5. Experiments

We test our uncertainty-based transformation sampling scheme on both image and text classification tasks. First, our sampling scheme achieves more accurate results by finding more useful transformations compared to RandAugment. Second, we achieve comparable test accuracy to the SoTA Adversarial AutoAugment on CIFAR-10, CIFAR-100, and ImageNet with less training cost because our method is conceptually simpler. Finally, we evaluate our scheme in text augmentations to help train a sentiment analysis model.

5.1. Experimental Setup

Datasets and models. We consider the following datasets and models in our experiments.

CIFAR-10 and CIFAR-100: The two datasets are colored images with 10 and 100 classes, respectively. We evaluate our proposed method for classifying images using the following models : Wide-ResNet-28-10 (Zagoruyko & Komodakis, 2016), Shake-Shake (26 2x96d) (Gastaldi, 2017), and PyramidNet+ShakeDrop (Han et al., 2017; Yamada et al., 2018).

Street view house numbers (SVHN): This dataset contains color house-number images with 73,257 core images for training and 26,032 digits for testing. We use Wide-ResNet-28-10 model for classifying these images.

ImageNet Large-Scale Visual Recognition Challenge (ImageNet): This dataset includes images of 1000 classes, and has a training set with roughly 1.3M images, and a validation set with 50,000 images. We select ResNet-50 (He et al., 2016) to evaluate our method.

Comparison methods. For image classification tasks, we compare Algorithm 1 with AutoAugment (AA) (Cubuk et al., 2018), Fast AutoAugment (Fast AA) (Lim et al., 2019), Population Based Augmentation (PBA) (Ho et al., 2019), RandAugment (RA) (Cubuk et al., 2019), and Adversarial AutoAugment (Adv. AA) (Zhang et al., 2020). We also include the baseline model with the following default setting. For CIFAR-10 and CIFAR-100, we flip each image horizontally with probability 0.5 and then randomly crop a 32×32 sub-image from the padded image. For SVHN, we apply the cutout to every image. For ImageNet, we randomly resize and crop a 224×224 sub-image from the original image and then flip the image horizontally with probability 0.5.

Training procedures. Recall that Algorithm 1 contains three parameters, how many composition steps (L) we take, how many augmented data points (C) we generate and how many (S) we select for training. We set $L = 2$, $C = 4$ and $S = 1$ for our experiments on CIFAR datasets and SVHN. We set $L = 2$, $C = 8$ and $S = 4$ for our experiments on ImageNet. We consider $K = 16$ transformations in Algorithm 1, including AutoContrast, Brightness, Color, Contrast, Cutout, Equalize, Invert, Mixup, Posterize, Rotate, Sharpness, ShearX, ShearY, Solarize, TranslateX, TranslateY. See e.g. (Shorten & Khoshgoftaar, 2019) for descriptions of these transformations.

As is common in previous work, we also include a parameter to set the probability of applying a transformation in Line 4 of Algorithm 1. We set this parameter and the magnitude of each transformation randomly in a suitable range. We apply the augmentations over the entire training dataset. We report the results averaged over four random seeds.

5.2. Experimental Results

We apply Algorithm 1 on three image classification tasks (CIFAR10, CIFAR100, SVHN, and ImageNet) over several models. Table 3 summarizes the result. We highlight the comparisons to RandAugment and Adversarial AutoAugment since they dominate the other benchmark methods.

Improving classification accuracy over RandAugment. For Wide-ResNet-28-10, we find that our method outperforms RandAugment by 0.59% on CIFAR-10 and 1.24% on CIFAR-100. If we do not use Mixup, our method still outperforms RandAugment by 0.45% on CIFAR-10. For Shake-Shake and PyramidNet+ShakeDrop, our method improves the accuracy of RandAugment by 0.27% and 0.16% on CIFAR-10, respectively. For ResNet-50 on ImageNet dataset, our method achieves top-1 accuracy 79.14% which outperforms RandAugment by 1.54%.

Improving training efficiency over Adversarial AutoAugment. Our method achieves comparable accuracy to the current state-of-the-art on CIFAR-10, CIFAR-100, and ImageNet. Algorithm 1 uses additional inference cost to find the uncertain samples. And we estimate that the additional cost equals half of the training cost. However, the inference cost is 5x cheaper compared to training the adversarial network of Adversarial AutoAugment, which requires generating 8 times more samples for training.

Further improvement by increasing the number of augmented samples. Recall that Algorithm 1 contains a parameter S which controls how many new labeled data we generate per training data. The results in Table 3 use $C = 4$ and $S = 1$, but we can further boost the prediction accuracy by increasing C and S. In Table 4, we find that by setting $C = 8$ and $S = 4$, our method improves the accuracy of Adversarial AutoAugment on CIFAR-100 by 0.49% on Wide-ResNet-28-10.

5.3. Ablation Studies

Histgram of selected transformations. We examine the transformations selected by Algorithm 1 to better understand its difference compared to RandAugment. For this

Dataset	Model	Baseline	AA	Fast AA	PBA	RA	Adv. AA	Ours
CIFAR-10	Wide-ResNet-28-10	96.13	97.32	97.30	97.42	97.30	**98.10**	97.89%(±0.03%)
	Shake-Shake (26 2x96d)	97.14	98.01	98.00	97.97	98.00	98.15	**98.27%(±0.05%)**
	PyramidNet+ShakeDrop	97.33	98.52	98.30	98.54	98.50	98.64	**98.66%(±0.02%)**
CIFAR-100	Wide-ResNet-28-10	81.20	82.91	82.70	83.27	83.30	84.51	**84.54%(±0.09%)**
SVHN	Wide-ResNet-28-10	96.9	98.1	-	-	**98.3**	-	**98.3%(±0.03%)**
ImageNet	ResNet-50	76.31	77.63	77.60	-	77.60	**79.40**	79.14%

Table 3. Test accuracy (%) on CIFAR-10, CIFAR-100, and SVHN. We compare our method with default data augmentation (Baseline), AutoAugment (AA), Fast AutoAugment (Fast AA), Population Based Augmentation (PBA), RandAugment (RA), and Adversarial AutoAugment (Adv. AA). Our results are averaged over four random seeds except ImageNet experiments.

Dataset	Adv. AA	Ours ($S = 4$)
CIFAR-10	98.10(±0.15%)	**98.16%(±0.05%)**
CIFAR-100	84.51(±0.18%)	**85.02%(±0.18%)**

Figure 4. Increasing the number of augmented data points per training sample can further improve accuracy.

	RA	Adv. AA	Ours ($S = 1$)
Training (×)	1.0	8.0	~ 1.5

Figure 5. Comparing the training cost between our method, RA and Adv. AA on CIFAR-10 relative to RA. The training cost of Adv. AA is cited from the authors (Zhang et al., 2020).

purpose, we measure the frequency of transformations selected by Algorithm 1 every 100 epoch. If a transformation generates useful samples, then the model should learn from these samples. And the loss of these transformed data will decrease as a result. On the other hand, if a transformation generates bad samples that are difficult to learn, then the loss of these bad samples will remain large.

Figure 1 shows the sampling frequency of five compositions. We test the five compositions on a vanilla Wide-ResNet model. For transformations whose frequencies are decreasing, we get: Rotate and ShearY, 93.41%; TranslateY and Cutout, 93.88%. For transformations whose frequencies are increasing, we get: Posterize and Color, 89.12% (the results for other two are similar and we omit the details). Hence the results confirm that Algorithm 1 learns and reduces the frequencies of the better performing transformations.

5.4. Extension to Text Augmentations

While we have focused on image augmentations throughout the paper, we can also our ideas to text augmentations. We extend Algorithm 1 to a sentiment analysis task as follows. We choose $\text{BERT}_{\text{LARGE}}$ as the baseline model, which is a 24 layer transformer network from (Devlin et al., 2018). We apply our method to three augmentations: back-translation (Yu et al., 2018), switchout (Wang et al., 2018), and word replace (Xie et al., 2019).

Dataset. We use the Internet movie database (IMDb) with 50,000 movie reviews. The goal is to predict whether the sentiment of the review is positive or negative.

Comparison methods. We compare with pre-BERT SoTA, $\text{BERT}_{\text{LARGE}}$, and unsupervised data augmentation (UDA) (Xie et al., 2019). UDA uses $\text{BERT}_{\text{LARGE}}$ initialization and training on 20 supervised examples and DBPedia (Lehmann et al., 2015) as an unsupervised source.

Results. We find that our method achieves test accuracy 95.96%, which outperforms all the other methods by at least 0.28%. For reference, the result of using Pre-BERT SoTA is 95.68%. The result of using $\text{BERT}_{\text{LARGE}}$ is 95.22%. The result of using UDA is 95.22%.

6. Related Work

Image augmentations. We describe a brief summary and refer interested readers to the excellent survey by (Shorten & Khoshgoftaar, 2019) for complete references.

Data augmentation has become a standard practice in computer vision such as image classification tasks. First, individual transformations such as horizontal flip and mixup have shown improvement over strong vanilla models. Beyond individual transformations, one approach to search for compositions of transformations is to train generative adversarial networks to generate new images as a form of data augmentation (Sixt et al., 2018; Laine & Aila, 2016; Odena, 2016; Gao et al., 2018). Another approach is to use reinforcement learning based search methods (Hu et al., 2019). The work of Cubuk et al. (2018) searches for the augmentation schemes on a small surrogate dataset. While this idea reduces the search cost, it was shown that using a small surrogate dataset results in sub-optimal augmentation policies (Cubuk et al., 2019).

The work of Kuchnik & Smith (2018) is closely related to ours since they also experiment with the idea of uncertainty-based sampling. Their goal is different from our work in that they use this idea to find a representative sub-sample of the training dataset that can still preserve the performance of applying augmentation policies. Recently, the idea of mixing data has been applied to semi-supervised learning by mixing feature representations as opposed to the input data (Berthelot et al., 2019).

Theoretical studies. Dao et al. (2019) propose a kernel theory to show that label-invariant augmentations are equivalent to transforming the kernel matrix in a way that incorporates the prior of the transformation. Chen et al. (2019) use group theory to show that incorporating the label-invariant property into an empirical risk minimization framework reduces variance.

Our theoretical setup is related to Xie et al. (2020); Raghunathan et al. (2020), with several major differences. First, in our setting, we assume that the label of an augmented data is generated from the training data, which is deterministic. In their setting, the label of an augmented data includes new information because the random noise is freshly drawn. Second, we consider the ridge estimator as opposed to the minimum norm estimator, since the ridge estimator includes an ℓ_2 regularization which is commonly used in practice. Finally, it would be interesting to extend our theoretical setup beyond linear settings (e.g. (Li et al., 2018; Zhang et al., 2019; Montanari et al., 2019)).

7. Conclusions and Future Work

In this work, we studied the theory of data augmentation in a simplified over-parametrized linear setting that captures the need to add more labeled data as in image settings, where there are more parameters than the number of data points. Despite the simplicity of the setting, we have shown three novel insights into three categories of transformations. We verified our theoretical insights on MNIST. And we proposed an uncertainty-based sampling scheme which outperforms random sampling. We hope that our work can spur more interest in developing a better understanding of data augmentation methods. Below, we outline several questions that our theory cannot yet explain.

First, one interesting future direction is to further uncover the mysterious role of mixup (Zhang et al.17). Our work has taken the first step by showing the connection between mixup and regularization. Meanwhile, Table 7 (in Appendix) shows that mixup can reduce the model error score (bias) on CIFAR-10. Our theory does not explain this phenomenon because our setup implicit assumes a single class (the linear model β) for all data. We believe that extending our work to a more sophisticated setting, e.g. mixed linear regression model, is an interesting direction to explain the working of mixup augmentation over multiple classes. Second, it would be interesting to consider the theoretical benefit of our proposed uncertainty-based sampling algorithm compared to random sampling, which can help tighten the connection between our theory and the proposed algorithm. We would like to remark that addressing this question likely requires extending the models and tools that we have developed in this work. Specifically, one challenge is how to come up with a data model that will satisfy the label-invariance property for a large family of linear transformations. We leave these questions for future work.

Acknowledgements. We gratefully acknowledge the support of DARPA under Nos. FA86501827865 (SDH) and FA86501827882 (ASED); NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ONR under No. N000141712266 (Unifying Weak Supervision); the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, American Family Insurance, Google Cloud, Swiss Re, the HAI-AWS Cloud Credits for Research program, and members of the Stanford DAWN project: Teradata, Facebook, Google, Ant Financial, NEC, VMWare, and Infosys. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of DARPA, NIH, ONR, or the U.S. Government.

Gregory Valiant's contributions were supported by NSF awards 1804222, 1813049 and 1704417, DOE award DE-SC0019205 and an ONR Young Investigator Award. Our experiments are partly run on Stanford University's SOAL cluster hosted in the Department of Management Science and Engineering (cf. https://5harad.com/soal-cluster/).

References

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Chen, S., Dobriban, E., and Lee, J. H. Invariance reduces variance: Understanding data augmentation in deep learning and beyond. *arXiv preprint arXiv:1907.10905*, 2019.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

Dao, T., Gu, A., Ratner, A. J., Smith, V., De Sa, C., and Ré, C. A kernel theory of modern data augmentation. *Proceedings of machine learning research*, 97:1528, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Gao, H., Shou, Z., Zareian, A., Zhang, H., and Chang, S.-F. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems*, pp. 975–985, 2018.

Gastaldi, X. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.

Han, D., Kim, J., and Kim, J. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5927–5935, 2017.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. Faster autoaugment: Learning augmentation strategies using backpropagation. *arXiv preprint arXiv:1911.06987*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ho, D., Liang, E., Stoica, I., Abbeel, P., and Chen, X. Population based augmentation: Efficient learning of augmentation policy schedules. *arXiv preprint arXiv:1905.05393*, 2019.

Hu, Z., Tan, B., Salakhutdinov, R. R., Mitchell, T. M., and Xing, E. P. Learning data manipulation for augmentation and weighting. In *Advances in Neural Information Processing Systems*, pp. 15738–15749, 2019.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kuchnik, M. and Smith, V. Efficient augmentation via data subsampling. *arXiv preprint arXiv:1810.05222*, 2018.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

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

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47, 2018.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. In *Advances in Neural Information Processing Systems*, pp. 6665–6675, 2019.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

Montanari, A., Ruan, F., Sohn, Y., and Yan, J. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

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

Rajput, S., Feng, Z., Charles, Z., Loh, P.-L., and Papailiopoulos, D. Does data augmentation lead to positive margin? *arXiv preprint arXiv:1905.03177*, 2019.

Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., and Ré, C. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pp. 3236–3246, 2017.

Riedel, K. S. A sherman–morrison–woodbury identity for rank augmenting matrices with application to centering. *SIAM Journal on Matrix Analysis and Applications*, 13 (2):659–662, 1992.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Sixt, L., Wild, B., and Landgraf, T. Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI*, 5:66, 2018.

Wang, X., Pham, H., Dai, Z., and Neubig, G. Switchout: an efficient data augmentation algorithm for neural machine translation. *arXiv preprint arXiv:1808.07512*, 2018.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

Xie, S. M., Raghunathan, A., Yang, F., Duchi, J. C., and Liang, P. When covariate-shifted data augmentation increases test error and how to fix it, 2020.

Yamada, Y., Iwamura, M., Akiba, T., and Kise, K. Shakedrop regularization for deep residual learning. *arXiv preprint arXiv:1802.02375*, 2018.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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

Zhang, H., Sharan, V., Charikar, M., and Liang, Y. Recovery guarantees for quadratic tensors with limited observations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Zhang, X., Wang, Q., Zhang, J., and Zhong, Z. Adversarial autoaugment. In *International Conference on Learning Representations*, 2020.