A Kernel Theory of Modern Data Augmentation

Tri Dao [1] **Albert Gu** [1] **Alexander J. Ratner** [1] **Virginia Smith** [2] **Christopher De Sa** [3] **Christopher Ré** [1]

Abstract

Data augmentation, a technique in which a training set is expanded with class-preserving transformations, is ubiquitous in modern machine learning pipelines. In this paper, we seek to establish a theoretical framework for understanding data augmentation. We approach this from two directions: First, we provide a general model of augmentation as a Markov process, and show that kernels appear naturally with respect to this model, even when we do not employ kernel classification. Next, we analyze more directly the effect of augmentation on kernel classifiers, showing that data augmentation can be approximated by first-order feature averaging and second-order variance regularization components. These frameworks both serve to illustrate the ways in which data augmentation affects the downstream learning model, and the resulting analyses provide novel connections between prior work in invariant kernels, tangent propagation, and robust optimization. Finally, we provide several proof-of-concept applications showing that our theory can be useful for accelerating machine learning workflows, such as reducing the amount of computation needed to train using augmented data, and predicting the utility of a transformation prior to training.

1. Introduction

The process of augmenting a training dataset with synthetic examples has become a critical step in modern machine learning pipelines. The aim of data augmentation is to artificially create new training data by applying transformations, such as rotations or crops for images, to input data while preserving the class labels. This practice has many potential benefits: Data augmentation can encode prior knowledge about data or task-specific invariances, act as regularizer to make the resulting model more robust, and provide resources to data-hungry deep learning models. As a testament to its growing importance, the technique has been used to achieve nearly all state-of-the-art results in image recognition (Cireşan et al., 2010; Dosovitskiy et al., 2016; Graham, 2014; Sajjadi et al., 2016), and is becoming a staple in many other areas as well (Uhlich et al., 2017; Lu et al., 2006). Learning augmentation policies alone can also boost the state-of-the-art performance in image classification tasks (Ratner et al., 2017; Cubuk et al., 2018).

Despite its ubiquity and importance to the learning process, data augmentation is typically performed in an ad-hoc manner with little understanding of the underlying theoretical principles. In the field of deep learning, for example, data augmentation is commonly understood to act as a regularizer by increasing the number of data points and constraining the model (Goodfellow et al., 2016; Zhang et al., 2017). However, even for simpler models, it is not well-understood how training on augmented data affects the learning process, the parameters, and the decision surface of the resulting model. This is exacerbated by the fact that data augmentation is performed in diverse ways in modern machine learning pipelines, for different tasks and domains, thus precluding a general model of transformation. Our results show that regularization is only part of the story.

In this paper, we aim to develop a theoretical understanding of data augmentation. First, in Section 3, we analyze data augmentation as a Markov process, in which augmentation is performed via a random sequence of transformations. This formulation closely matches how augmentation is often applied in practice. Surprisingly, we show that performing k-nearest neighbors with this model asymptotically results in a kernel classifier, where the kernel is a function of the base augmentations. These results demonstrate that kernels appear naturally with respect to data augmentation, regardless of the base model, and illustrate the effect of augmentation on the learned representation of the original data.

Motivated by the connection between data augmentation and kernels, in Section 4 we show that a kernel classifier on augmented data approximately decomposes into two components: (i) an averaged version of the transformed features, and (ii) a data-dependent variance regularization term. This suggests a more nuanced explanation of data augmentation—namely, that it improves generalization *both by inducing*

[1]Department of Computer Science, Stanford University, California, USA [2]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pennsylvania, USA [3]Department of Computer Science, Cornell University, New York, USA. Correspondence to: Tri Dao <trid@cs.stanford.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

invariance and by reducing model complexity. We validate the quality of our approximation empirically, and draw connections to other generalization-improving techniques, including recent work in invariant learning (Zhao et al., 2017; Mroueh et al., 2015; Raj et al., 2017) and robust optimization (Namkoong & Duchi, 2017).

Finally, in Section 5, to illustrate the utility of our theoretical understanding of augmentation, we explore promising practical applications, including: (i) developing a diagnostic to determine, prior to training, the importance of an augmentation; (ii) reducing training costs for kernel methods by allowing for augmentations to be applied directly to features—rather than the raw data—via a random Fourier features approach; and (iii) suggesting a heuristic for training neural networks to reduce computation while realizing most of the accuracy gain from augmentation.

2. Related Work

Data augmentation has long played an important role in machine learning. For many years it has been used, for example, in the form of *jittering* and *virtual examples* in the neural network and kernel methods literatures (Sietsma & Dow, 1991; Schölkopf et al., 1996; Decoste & Schölkopf, 2002). These methods aim to augment or modify the raw training data so that the learned model will be invariant to known transformations or perturbations. Given its importance, recent efforts have been made to apply data augmentation more efficiently, for example with subsampling (Kuchnik & Smith, 2019). There has also been significant work in incorporating invariance directly into the model or training procedure, rather than by expanding the training set (van der Wilk et al., 2018; Tai et al., 2019). One illustrative example is that of tangent propagation for neural networks (Simard et al., 1992; 1998), which proposes a regularization penalty to enforce local invariance, and has been extended in several recent works (Rifai et al., 2011; Demyanov et al., 2015; Zhao et al., 2017). However, while efforts have been made that loosely connect traditional data augmentation with these methods (Leen, 1995; Zhao et al., 2017), there has not been a rigorous study on how these sets of procedures relate in the context of modern models and transformations.

In this work, we make explicit the connection between augmentation and modifications to the model, and show that prior work on tangent propagation can be derived as a special case of our more general theoretical framework (Section 5). Moreover, we draw connections to recent work on invariant learning (Mroueh et al., 2015; Raj et al., 2017) and robust optimization (Namkoong & Duchi, 2017), illustrating that data augmentation not only affects the model by increasing invariance to specific transformations, but also by reducing the variance of the estimator. These analyses lead to an important insight into how invariance can be most effectively applied for kernel methods and deep learning architectures (Section 5), which we show can be used to reduce training computation and diagnose the effectiveness of various transformations.

Prior theory also does not capture the complex process by which data augmentation is often applied. For example, previous work (Bishop, 1995; Chapelle et al., 2001) shows that adding noise to input data has the effect of regularizing the model, but these effects have yet to be explored for more commonly applied complex transformations, and it is not well-understood how the inductive bias embedded in complex transformations manifest themselves in the invariance of the model (addressed here in Section 4). A common recipe in achieving state-of-the-art accuracy in image classification is to apply a sequence of more complex transformations such as crops, flips, or local affine transformations to the training data, with parameters drawn randomly from hand-tuned ranges (Cireşan et al., 2010; Dosovitskiy et al., 2014). Similar strategies have also been employed in applications of classification for audio (Uhlich et al., 2017) and text (Lu et al., 2006). In Section 3, we analyze a motivating model reaffirming the connection between augmentation and kernel methods, even in the setting of complex and composed transformations.

Finally, while data augmentation has been well-studied in the kernels literature (Burges, 1999; Schölkopf et al., 1996; Muandet et al., 2012), it is typically explored in the context of simple geometrical invariances with closed forms. For example, van der Wilk et al. (2018) use Gaussian processes to learn these invariances from data by maximizing the marginal likelihood. Further, the connection is often approached in the opposite direction—by looking for kernels that satisfy certain invariance properties (Haasdonk & Burkhardt, 2007; Teo et al., 2008). We instead approach the connection directly via data augmentation, and show that even complicated augmentation procedures akin to those used in practice can be represented as a kernel method.

3. Data Augmentation as a Kernel

To begin our study of data augmentation, we propose and investigate a model of augmentation as a Markov process, inspired by the general manner in which the process is applied—via the composition of multiple different types of transformations. Surprisingly, we show that this augmentation model combined with a k-nearest neighbor (k-NN) classifier is asymptotically equivalent to a kernel classifier, where the kernel is a function of the base transformations. While the technical details of the section can be skipped on a first reading, the central message is that kernels appear naturally in relation to data augmentation, even when we do not start with a kernel classifier. This provides additional motivation to study kernel classifiers trained on augmented

data, as in Section 4.

Markov Chain Augmentation Process. In data augmentation, the aim is to perform class-preserving transformations to the original training data to improve generalization. As a concrete example, a classifier that correctly predicts an image of the number '1' should be able to predict this number whether or not the image is slightly rotated, translated, or blurred. It is therefore common to pick some number of augmentations (e.g., for images: rotation, zoom, blur, flip, etc.), and to create synthetic examples by taking an original data point and applying a sequence of these augmentations. To model this process, we consider the following procedure: given a data point, we pick augmentations from a set at random, applying them one after the other. To avoid deviating too far, with some probability we discard the point and start over from a random point in the original dataset. We formalize this below.

Definition 1 (Markov chain augmentation model)**.** Given a dataset of n examples $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, we *augment* the dataset via *augmentation matrices* $A_1, A_2, \ldots, A_m$, for $A_j \in \mathbb{R}^{\Omega \times \Omega}$, which are stochastic transition matrices over a finite state space of possible labeled (augmented) examples $\Omega := \mathcal{X} \times \mathcal{Y}$. We model this via a discrete time Markov chain with the transitions:

- With probability proportional to β_j, a *transition* occurs via augmentation matrix A_j.
- With probability proportional to γ_i, a *retraction* to the training set occurs, and the state resets to z_i.

For example, the probability of retracting to training example z_1 is $\gamma_1/(\gamma_1 + \cdots + \gamma_n + \beta_1 + \cdots + \beta_m)$. The augmentation process starts from any point and follows Definition 1 for an arbitrary amount of time. The retraction steps intuitively keep the final distribution grounded closer to the original training points.

From Definition 1, by conditioning on which transition is chosen, it is evident that the entire process is equivalent to a Markov chain whose transition matrix is the weighted average of the base transitions. Note that the transition matrices A_j do not need to be materialized but are implicit from the description of the augmentation. A concrete example is given in Section B.2. Without loss of generality, we let all rates β_j, γ_i be normalized with $\sum_j \gamma_i = 1$. Let $\{e_\omega\}_{\omega \in \Omega}$ be the standard basis of Ω, and let e_{z_i} be the basis element corresponding to z_i. The resulting transition matrix and stationary distribution are given below; proofs and additional details are provided in Appendix A. This describes the long-run distribution of the augmented dataset.

Proposition 1. *The described augmentation process is a Markov chain with transition matrix:*

$$R = \left(1 + \sum_{j=1}^m \beta_j\right)^{-1} \left[\sum_{i=1}^m \beta_j A_j + \sum_{i=1}^n \gamma_i \left(\mathbf{1} e_{z_i}^\top\right)\right] .$$

Lemma 1 (Stationary distribution)**.** *The stationary distribution is given by:*

$$\pi = \rho^\top \left(I(\beta + 1) - A\right)^{-1} , \tag{1}$$

where

$$A = \sum_{j=1}^m \beta_j A_j , \quad \beta = \sum_{j=1}^m \beta_j , \quad \rho = \sum_{i=1}^n \gamma_i e_{z_i} .$$

Lemma 1 agrees intuitively with the augmentation process: When all $\beta_j \approx 0$ (i.e., low rate of augmentation), Lemma 1 implies that the stationary distribution π is close to ρ, the original data distribution. As β_j increases, the stationary distribution becomes increasingly distorted by the augmentations.

Classification Yields a Kernel. Using our proposed model of augmentation, we can show that classifying an unseen example using augmented data results in a kernel classifier. In doing so, we can observe the effect that augmentation has on the learned feature representation of the original data. We discuss several additional uses and extensions of the result itself in Appendix A.1.

Theorem 1. *Consider running the Markov chain augmentation process in Definition 1 and classifying an unseen example $x \in \mathcal{X}$ using an asymptotically Bayes-optimal classifier, such as k-nearest neighbors. Suppose that the A_i are time-reversible with equal stationary distributions. Then in the limit as time $T \to \infty$ and $k \to \infty$, this classification has the following form:*

$$\hat{y} = \operatorname{sign} \sum_{i=1}^n y_i \alpha_{z_i} K_{x_i, x} , \tag{2}$$

where $\alpha \in \mathbb{R}^\Omega$ is supported only on the dataset $z_1, \ldots, z_n$, and $K \in \mathbb{R}^{\Omega \times \Omega}$ is a kernel matrix (i.e., K is symmetric positive definite and non-negative) depending only on all the augmentations A_j, β_j.

Theorem 1 follows from formulating the stationary distribution (Lemma 1) as $\pi = \alpha^\top K$ for a kernel matrix K and $\alpha \in \mathbb{R}^\Omega$. Noting that k-NN asymptotically acts as a Bayes classifier, selecting the most probable label according to this stationary distribution, leads to (2).[1] In Appendix A, we include a closed form for α and K along with the proof. We include details and examples, and elaborate on the strength of the assumptions.

Takeaways. This result has two important implications: First, kernels appear naturally in relation to complex forms

[1] We use k-NN as a simple example of a nonparametric classifier, but the result holds for any asymptotically Bayes classifier.

of augmentation, even when we do not begin with a kernel classifier. This underscores the connection between augmentation and kernels even with complicated compositional models, and also serves as motivation for our focused study on kernel classifiers in Section 4. Second, and more generally, data augmentation—a process that produces synthetic training examples from the raw data—can be understood more directly based on its effect on downstream components in the learning process, such as the features of the original data and the resulting learned model. We make this link more explicit in Section 4, and show how to exploit it in practice in Section 5.

4. Effects of Augmentation: Invariance and Regularization

In this section we build on the connection between kernels and augmentation in Section 3, exploring directly the effect of augmentation on a kernel classifier. It is commonly understood that data augmentation can be seen as a regularizer, in that it reduces generalization error but not necessarily training error (Goodfellow et al., 2016; Zhang et al., 2017). We make this more precise, showing that data augmentation has two specific effects: (i) increasing the invariance by averaging the features of augmented data points, and (ii) penalizing model complexity via a regularization term based on the variance of the augmented forms. These are two approaches that have been explicitly applied to get more robust performance in machine learning, though outside of the context of data augmentation. We demonstrate connections to prior work in our derivation of the feature averaging (Section 4.1) and variance regularization (Section 4.2) terms. We also validate our theory empirically (Section 4.3), and in Section 5, show the practical utility of our analysis to both kernel and deep learning pipelines.

General Augmentation Process. To illustrate the effects of augmentation, we explore it in conjunction with a general kernel classifier. In particular, suppose that we have an original kernel K with a finite-dimensional[2] feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$, and we aim to minimize some smooth convex loss $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with parameter $w \in \mathbb{R}^D$ over a dataset $(x_1, y_1), \ldots, (x_n, y_n)$. The original objective function to minimize is $f(w) = \frac{1}{n} \sum_{i=1}^{n} l\left(w^\top \phi(x_i); y_i\right)$, with two common losses being logistic $l(\hat{y}; y) = \log(1 + \exp(-y\hat{y}))$ and quadratic $l(\hat{y}; y) = (\hat{y} - y)^2$.

Now, suppose that we first augment the dataset using an augmentation kernel T. Whereas the augmentation kernel in Section 3 had a specific form based on the stationary distribution of the proposed Markov process, here we make this more general, simply requiring that for each data point x_i, $T(x_i)$ describes the distribution over data points into which x_i can be transformed. The new objective function becomes:

$$g(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{E}_{t_i \sim T(x_i)} \left[l\left(w^\top \phi(t_i); y_i\right) \right]. \quad (3)$$

4.1. Data Augmentation as Feature Averaging

We begin by showing that, to first order, objective (3) can be approximated by a term that computes the average augmented feature of each data point. In particular, suppose that the applied augmentations are "local" in the sense that they do not significantly modify the feature map ϕ. Using the first-order Taylor approximation, we can expand each term around any point ϕ_0 that does not depend on t_i:

$$\mathbf{E}_{t_i \sim T(x_i)} \left[l\left(w^\top \phi(t_i); y_i\right) \right] \approx$$
$$l\left(w^\top \phi_0; y_i\right) + \mathbf{E}_{t_i \sim T(x_i)} \left[w^\top (\phi_0 - \phi(t_i)) \right] l'(w^\top \phi_0; y_i).$$

Picking $\phi_0 = \mathbf{E}_{t_i \sim T(x_i)} \left[\phi(t_i)\right]$, the second term vanishes, yielding the *first-order approximation*:

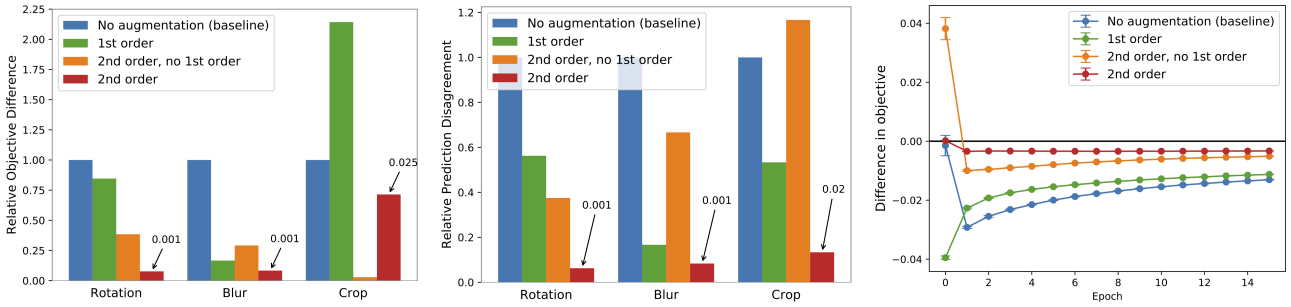
$$g(w) \approx \hat{g}(w) := \frac{1}{n} \sum_{i=1}^{n} l\left(w^\top \mathbf{E}_{t_i \sim T(x_i)} \left[\phi(t_i)\right]; y_i\right). \quad (4)$$

This is exactly the objective of a linear model with a new feature map $\psi(x) = \mathbf{E}_{t \sim T(x)} \left[\phi(t)\right]$, i.e., the average feature of all the transformed versions of x. If we overload notation and use $T(x, u)$ to denote the probability density of transforming x to u, this feature map corresponds to a new kernel:

$$\begin{aligned}
&\bar{K}(x, x') \\
&= \langle \psi(x), \psi(x') \rangle = \langle \mathbf{E}_{u \sim T(x)} \left[\phi(u)\right], \mathbf{E}_{u' \sim T(x')} \left[\phi(u')\right] \rangle \\
&= \int_{u \in \mathbb{R}^n} \int_{u' \in \mathbb{R}^n} \langle \phi(u), \phi(u') \rangle T(x, u) T(x', u') \, du' \, du \\
&= \int_{u \in \mathbb{R}^n} \int_{u' \in \mathbb{R}^n} K(u, u') T(x, u) T(x', u') \, du' \, du \\
&= (TKT^\top)(x, x').
\end{aligned}$$

That is, training a kernel linear classifier with a particular loss function plus data augmentation is equivalent, to first order, to training a linear classifier with the same loss on an *augmented kernel* $\bar{K} = TKT^\top$, with feature map $\psi(x) = \mathbf{E}_{t \sim T(x)} \left[\phi(t)\right]$. This feature map is exactly the embedding of the distribution of transformed points around x into the reproducing kernel Hilbert space (Muandet et al., 2017; Raj et al., 2017). This means that the first-order effect of training on augmented data is equivalent to training a support measure machine (Muandet et al., 2012), with the n input distributions corresponding to the n distributions of transformed points around $x_1, \ldots, x_n$. The new kernel $\bar{K}$ has the effect of increasing the invariance of the model, as averaging the features from transformed inputs that are not necessarily present in the original dataset makes the features less variable to transformation.

[2] We focus on finite-dimensional feature maps for ease of exposition, but the analysis still holds for infinite-dimensional feature maps.



(a) Absolute Objective Difference

(b) Prediction Disagreement

(c) Example Trace: Rotation

Figure 1. For the MNIST dataset, we validate that (a) the proposed approximate objectives $\hat{g}(w)$ and $\tilde{g}(w)$ are close to the true objective $g(w)$, and (b) training on the approximate objectives leads to similar predictions as training on the true objective. We plot the relative difference between the proposed approximations and the true augmented objective, in terms of difference in objective value (1a) and resulting test prediction disagreement (1b), using the non-augmented objective as a baseline. The 2nd-order approximation closely matches the true objective, particularly in terms of the resulting predictions. We observe that the accuracy of the approximations remains stable throughout training (1c). Full experiments are provided in Appendix E.

By Jensen's inequality, since the function l is convex, $\hat{g}(w) \leq g(w)$. In other words, if we solve the optimization problem that results from data augmentation, the resulting objective value using $\bar{K}$ will be no larger. Further, if we assume that the loss function is strongly convex and strongly smooth, we can quantify how much the solution to the first-order approximation and the solution of the original problem with augmented data will differ (see Proposition 3 in the appendix). We validate the accuracy of this first-order approximation empirically in Section 4.3.

4.2. Data Augmentation as Variance Regularization

Next, we show that the second-order approximation of the objective on an augmented dataset is equivalent to variance regularization, making the classifier more robust. We can get an exact expression for the error by considering the second-order term in the Taylor expansion, with ζ_i denoting the remainder function from Taylor's theorem:

$$
\begin{aligned}
&g(w) - \hat{g}(w) \\
&= \frac{1}{2n} \sum_{i=1}^{n} \mathbf{E}_{t_i \sim T(x_i)} \left[\left(w^\top (\phi(t_i) - \psi(x_i)) \right)^2 l''(\zeta_i (w^\top \phi(t_i)); y_i) \right] \\
&= w^\top \left(\frac{1}{2n} \sum_{i=1}^{n} \mathbf{E}_{t_i \sim T(x_i)} \left[\Delta_{t_i,x_i} \Delta_{t_i,x_i}^\top l''(\zeta_i (w^\top \phi(t_i)); y_i) \right] \right) w,
\end{aligned}
$$

where $\Delta_{t_i,x_i} := \phi(t_i) - \psi(x_i)$ is the difference between the features of the transformed image t_i and the averaged features $\psi(x_i)$. If (as is the case for logistic and linear regression) l'' is independent of y, the error term is independent of the labels. That is, the original augmented objective g is the modified objective $\hat{g}$ plus some regularization that is a function of the training examples, but not the labels. In other words, data augmentation has the effect of performing *data-dependent regularization*.

The second-order approximation to the objective is:

$$
\begin{aligned}
\tilde{g}(w) := \hat{g}(w) + \\
\frac{1}{2n} \sum_{i=1}^{n} w^\top \mathbf{E}_{t_i \sim T(x_i)} \left[\Delta_{t_i,x_i} \Delta_{t_i,x_i}^\top \right] l''(w^\top \psi(x_i)) w \,.
\end{aligned} \tag{5}
$$

For a fixed w, this error term is exactly the variance of the output $w^\top \phi(X)$, where the true data X is assumed to be sampled from the empirical data points x_i and their augmented versions specified by $T(x_i)$, weighted by $l''(w^\top \psi(x_i))$. This data-dependent regularization term favors weight vectors that produce similar outputs $w^T \phi(x)$ and $w^T \phi(x')$ if x' is a transformed version of x.

4.3. Validation of Approximation

We empirically validate[3] the first- and second-order approximations, $\hat{g}(w)$ and $\tilde{g}(w)$, on MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets, performing rotation, crop, or blur as augmentations, and using either an RBF kernel with random Fourier features (Rahimi & Recht, 2007) or LeNet (details in Appendix E.1) as a base model. Our results show that while both approximations perform reasonably well, the second-order approximation indeed results in a better approximation of the actual objective than the first-order approximation alone, validating the significance of the variance regularization component of data augmentation.

In particular, in Figure 1a, we plot the difference after 10 epochs of SGD training, between the actual objective func-

[3]Code to reproduce experiments and plots: https://github.com/HazyResearch/augmentation_code

tion over augmented data $g(w)$ and: (i) the first-order approximation $\hat{g}(w)$, (ii) second-order approximation $\tilde{g}(w)$, and (iii) second-order approximation without the first-order term, $f(w) + (\tilde{g}(w) - \hat{g}(w))$. As a baseline, we plot these differences relative to the difference between the augmented and non-augmented objective (i.e., the original images), $f(w)$. In Figure 1b, to see how training on approximate objectives affect the predicted test values, we plot the prediction disagreement between the model trained on true objective and the models trained on approximate objectives. Finally, Figure 1c shows that these approximations are relatively stable in terms of performance throughout the training process. For the CIFAR-10 dataset and the LeNet model (Appendix E), the results are quite similar, though we additionally observe that the first-order approximation is very close to the model trained without augmentation for LeNet, suggesting that the data-dependent regularization of the second-order term may be the dominating effect in models with learned feature maps.

4.4. Connections to Prior Work

The approximations we have provided in this section unify several seemingly disparate works.

Invariant kernels. The derived first-order approximation can capture prior work in *invariant kernels* as a special case, when the transformations of interest form a group and averaging features over the group induces invariance (Mroueh et al., 2015; Raj et al., 2017). The form of the averaged kernel can then be used to learn the invariances from data (van der Wilk et al., 2018).

Robust optimization. Our work also connects to *robust optimization*. For example, previous work (Bishop, 1995; Chapelle et al., 2001) shows that adding noise to input data has the effect of regularizing the model. Maurer & Pontil (2009) bounds generalization error in terms of the empirical loss and the variance of the estimator. The second-order objective here adds a variance penalty term, thus optimizing generalization and automatically balancing bias (empirical loss) and variance with respect to the input distribution coming from the empirical data and their transformed versions (this is presumably close to the population distribution if the transforms capture the right invariance in the dataset). Though the resulting problem is generally non-convex, it can be approximated by a distributionally robust convex optimization problem, which can be efficiently solved by a stochastic procedure (Namkoong & Duchi, 2017; 2016).

Tangent propagation. In Section 5.3, we show that when applied to neural networks, the described second-order objective can realize classical *tangent propagation* methods (Simard et al., 1992; 1998; Zhao et al., 2017) as a special case. More precisely, the second-order only term (orange in Figure 1) is equivalent to the approximation described in Zhao et al. (2017), proposed there in the context of regularizing CNNs. Our results indicate that considering both the first- and second-order terms, rather than just this second-order component, in fact results in a more accurate approximation of the true objective, e.g., providing a 6–9x reduction in the resulting test prediction disagreement (Figure 1b). This suggests an approach to improve classical tangent propagation methods, explored in Section 5.3.

5. Practical Connections: Accelerating Training With Data Augmentation

We now present several proof-of-concept applications to illustrate how the theoretical insights in Section 4 can be used to accelerate training with data augmentation. First, we propose a kernel similarity metric that can be used to quickly predict the utility of potential augmentations, helping to obviate the need for guess-and-check work. Next, we explore ways to reduce training computation over augmented data, including incorporating augmentation directly in the learned features with a random Fourier features approach, and applying our derived approximation at various layers of a deep network to reduce overall computation. We perform these experiments on common benchmark datasets, MNIST and CIFAR-10, as well a real-world mammography tumor-classification dataset, DDSM.

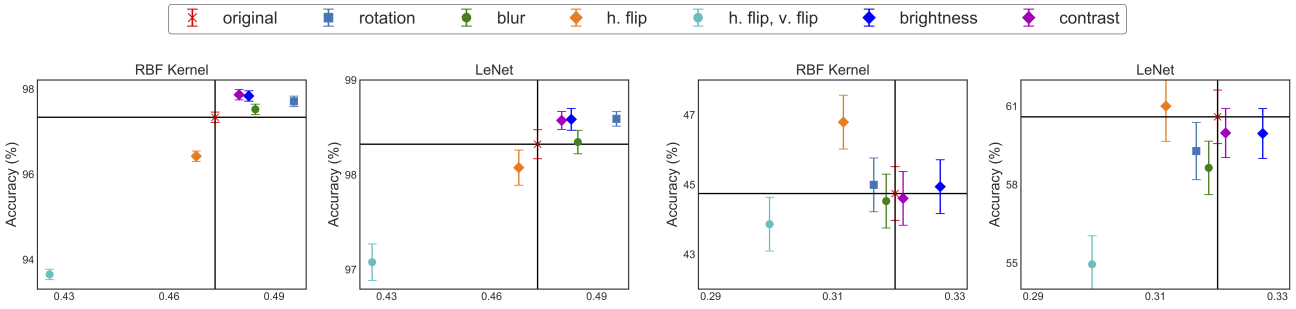
5.1. A Fast Kernel Metric for Augmentation Selection

For new tasks and datasets, manually selecting, tuning, and composing augmentations is one of the most time-consuming processes in a machine learning pipeline, yet is critical to achieving state-of-the-art performance. Here we propose a kernel alignment metric, motivated by our theoretical framework, to quickly estimate if a transformation is likely to improve generalization performance *without performing end-to-end training*.

Kernel alignment metric. Given a transformation T, and an original feature map $\phi(x)$, we can leverage our analysis in Section 4.1 to approximate the features for each data point x as $\psi(x) = \mathbf{E}_{t\sim T(x)}[\phi(t)]$. Defining the feature kernel $\bar{K}(x, x') = \psi(x)^\top \psi(x')$ and the label kernel $K_Y(y, y') = \mathbf{1}\{y = y'\}$, we can compute the *kernel target alignment* (Cristianini et al., 2002) between the feature kernel $\bar{K}$ and the target kernel K_Y *without training*:

$$\hat{A}(X, \bar{K}, K_Y) = \frac{\langle \bar{K}, K_Y \rangle}{\sqrt{\langle \bar{K}, \bar{K} \rangle \langle K_Y, K_Y \rangle}},$$

where $\langle K_a, K_b \rangle = \sum_{i,j}^{n} K_a(x_i, x_j) K_b(x_i, x_j)$. This alignment statistic can be estimated quickly and accurately from subsamples of the data (Cristianini et al., 2002). In our case, we use random Fourier features (Rahimi & Recht, 2007) as an approximate feature map $\phi(x)$ and sample $t \sim T(x)$ to estimate the averaged feature $\psi(x) = \mathbf{E}_{t\sim T(x)}[\phi(t)]$.



(a) MNIST: Kernel Target Alignment

(b) CIFAR-10: Kernel Target Alignment

Figure 2. Accuracy vs. kernel target alignment for RBF kernel and LeNet models, for MNIST (left) and CIFAR-10 (right) datasets. This alignment metric can be used to quickly select transformations (e.g., MNIST: rotation) that improve performance and avoid bad transformations (e.g., MNIST: flips).

The kernel target alignment measures the extent to which points in the same class have similar features. If this alignment is larger than that between the original feature kernel $K(x, x') = \phi(x)^\top \phi(x)$ and the target kernel, we postulate that the transformation T is likely to improve generalization. We validate this method on MNIST and CIFAR-10 with numerous transformations (rotation, blur, flip, brightness, and contrast). In Figure 2, we plot the accuracy of the kernel classifier and LeNet against the kernel target alignment. We see that there is indeed a correlation between kernel alignment and accuracy, as points tend to cluster in the upper right (higher alignment, higher accuracy) and lower left (lower alignment, lower accuracy) quadrants, indicating that this approach may be practically useful to detect the utility of a transformation prior to training.

5.2. Efficient Augmentation via Random Features

Beyond predicting the utility of an augmentation, we can also use our theory to reduce the computation required to perform augmentation on a kernel classifier—resulting, e.g., in a 4x speedup while achieving the same accuracy (MNIST, Table 1). For affine transforms (e.g., rotation, translation, scaling, shearing), we can perform transforms *directly on the approximate kernel features*, rather than the raw data, thus gaining efficiency while maintaining accuracy.

Recall from Section 4 that the first-order approximation of the new feature map is given by $\psi(x) = \mathbf{E}_{t\sim T(x)}[\phi(t)]$, i.e., the average feature of all the transformed versions of x. Suppose that the transform is linear in x of the form $A_\alpha x$, where the transformation is parameterized by α. For example, a rotation by angle α has the form $T(x) = R_\alpha x$, where R_α is a $d \times d$ matrix that 2D-rotates the image x. Further, assume that the original kernel $k(x, x')$ is shift-invariant (say an RBF kernel), so that it can be approximated by random Fourier features (Rahimi & Recht, 2007). Instead of transforming the data point x itself, we can transform the averaged feature map for x directly as:

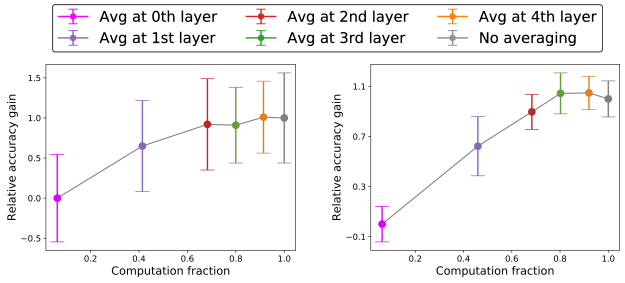
$$\tilde{\psi}(x)_k = \frac{1}{s\sqrt{D}} \sum_{j=1}^{s} \exp(i(A_{\alpha_j}^\top \omega_k)^\top x), \quad k = 1, \ldots, D,$$

where $\omega_1, \ldots, \omega_D$ are sampled from the spectral distribution, and $\alpha_1, \ldots, \alpha_s$ are sampled from the distribution of the parameter α (e.g., uniformly from $[-15, 15]$ if the transform is rotation by α degrees). One could also approximate the expectation over α by Gaussian quadrature (Dao et al., 2017), which could be more accurate than Monte Carlo sampling when α is low-dimensional. This type of random feature map has been suggested by Raj et al. (2017) in the context of kernels invariant to actions of a group. Our theoretical insights in Section 4 thus connect data augmentation to invariant kernels, allowing us to leverage the approximation techniques in this area. Our framework highlights additional ways to improve this procedure: if we view augmentation as a modification of the feature map, we naturally apply this feature map to test data points as well, implicitly reducing the variance in the features of different versions of the same data point. This variance regularization is the second goal of data augmentation discussed in Section 4.

We validate this approach on standard image datasets MNIST and CIFAR-10, along with a real-world mammography tumor-classification dataset called Digital Database for Screening Mammography (DDSM) (Heath et al., 2000; Clark et al., 2013; Lee et al., 2016). DDSM comprises 1506 labeled mammograms, to be classified as benign versus malignant tumors. In Table 1, we compare: (i) a baseline model trained on non-augmented data, (ii) a model trained on the true augmented objective, and (iii) a model that uses augmented random Fourier features. We augment via rotation between -15 and 15 degrees. All models are RBF kernel classifiers with 10,000 random Fourier features, and we report the mean accuracy and standard deviation over 10 trials. To make the problem more challenging, we also randomly rotate the test data points. The results show that augmented

Table 1. Performance of augmented random Fourier features on MNIST, CIFAR-10, and DDSM.

Model	MNIST		CIFAR-10		DDSM	
	Acc. (%)	Time	Acc. (%)	Time	Acc. (%)	Time
No augmentation	96.1 ± 0.1	34s	39.4 ± 0.5	51s	57.3 ± 6.7	27s
Traditional augmentation	97.6 ± 0.2	220s	45.3 ± 0.5	291s	59.4 ± 3.2	61s
Augmented RFFs	97.6 ± 0.1	54s	45.2 ± 0.4	124s	58.8 ± 5.1	34s



(a) MNIST (b) CIFAR-10

Figure 3. Accuracy gain relative to baseline (no augmentation) when averaging at various layers of a LeNet network. Approximation at earlier layers saves computation but can reduce the fidelity of the approximation.

random Fourier features can retain 70-100% of the accuracy boost of augmentation, with 2-4x faster training time.

5.3. Intermediate-Layer Feature Averaging for Deep Learning

Finally, while our theory does not hold exactly given the non-convexity of the objective, we show that our theoretical framework also suggests ways in which augmentation can be efficiently applied in deep learning pipelines. Let the first k layers of a deep neural network define a feature map ϕ, and the remaining layers define a non-linear function $f(\phi(x))$. The loss on each data point is then of the form $\mathbf{E}_{t_i \sim T(x_i)}\left[l(f(\phi(t_i)); y_i)\right]$. The 2nd-order Taylor expansion around $\psi(x_i) = \mathbf{E}_{t_i \sim T(x_i)}\left[\phi(t_i)\right]$ yields the objective:

$$\frac{1}{n}\sum_{i=1}^{n} l(f(\psi(x_i)); y_i) + \frac{1}{2}\mathbf{E}_{t_i \sim T(x_i)}\Big[(\phi(t_i) - \psi(x_i))^\top \nabla^2_{\psi(x_i)} l(f(\psi(x_i)); y_i)(\phi(t_i) - \psi(x_i))\Big] .$$

If $f(\phi(x)) = w^\top \phi(x)$, we recover the result in Section 4 (Equation 5). Operationally, we can carrying out the forward pass on all transformed versions of the data points up to layer k (i.e., computing $\phi(t_i)$), and then averaging the features and continuing with the remaining layers using this averaged feature, thus reducing computation.

We train with this approach, applying the approximation at various layers of a LeNet network using rotation as the augmentation. To get a rough measure of tradeoff between accuracy of the model and computation, we record the fraction of time spent at each layer in the forward pass, and use this to measure the expected reduction in computation when approximating at layer k. In Figure 3, we plot the relative accuracy gain of the classifier when trained on approximate objectives against the fraction of computation time, where 0 corresponds to accuracy (averaged over 10 trials) of training on original data, and 1 corresponds to accuracy of training on true augmented objective $g(w)$. These results indicate, e.g., that this approach can reduce computation by 30%, while maintaining 92% of the accuracy gain (red, Figure 3a). In Appendix E.4, we demonstrate similar results in terms of the test prediction distribution throughout training.

Connection to tangent propagation. If we perform the described averaging *before the very first layer* and use the analytic form of the gradient with respect to the transformations (i.e., *tangent vectors*), this procedure recovers *tangent propagation* (Simard et al., 1992). The connection between augmentation and tangent propagation in this special case was recently observed in Zhao et al. (2017). However, as we see in Figure 3, applying the approximation at the first layer (standard tangent propagation) can in fact yield very poor accuracy results—similar to performing no augmentation—showing that our more general approximation can improve this approach in practice.

6. Conclusion

We have taken steps to establish a theoretical base for modern data augmentation. First, we analyze a general Markov process model and show that the k-nearest neighbors classifier applied to augmented data is asymptotically equivalent to a kernel classifier, illustrating the effect that augmentation has on downstream representation. Next we show that local transformations for data augmentation can be approximated by first-order feature averaging and second-order variance regularization components, having the effects of inducing invariance and reducing model complexity. We use our insights to suggest ways to accelerate training for kernel and deep learning pipelines. Generally, a tension exists between incorporating domain knowledge more naturally via data augmentation, or through more principled kernel approaches. We hope our work will enable easier translation between these two paths, leading to simpler and more theoretically grounded applications of data augmentation.

Acknowledgments

We thank Fred Sala and Madalina Fiterau for helpful discussion, and Avner May for providing detailed feedback on earlier versions.

We gratefully acknowledge the support of DARPA under Nos. FA87501720095 (D3M) and FA86501827865 (SDH), NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity) and CCF1563078 (Volume to Velocity), ONR under No. N000141712266 (Unifying Weak Supervision), the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, Google, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, and American Family Insurance, Google Cloud, Swiss Re, Stanford Bio-X SIG Fellowship, and members of the Stanford DAWN project: Intel, Microsoft, Teradata, Facebook, Google, Ant Financial, NEC, SAP, 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.

References

Bishop, C. M. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Burges, C. J. C. Geometry and invariance in kernel based methods. *Advances in Kernel Methods*, pp. 89–116, 1999.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. Vicinal risk minimization. In Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13*, pp. 416–422. MIT Press, 2001.

Cireşan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010.

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al. The cancer imaging archive (TCIA): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. On kernel-target alignment. In *Neural Information Processing Systems*, 2002.

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.

Dao, T., De Sa, C. M., and Ré, C. Gaussian quadrature for kernel features. In *Advances in neural information processing systems*, pp. 6107–6117, 2017.

Decoste, D. and Schölkopf, B. Training invariant support vector machines. *Machine Learning*, 46(1):161–190, 2002.

Demyanov, S., Bailey, J., Kotagiri, R., and Leckie, C. Invariant backpropagation: how to train a transformation-invariant neural network. *arXiv:1502.04434*, 2015.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *Neural Information Processing Systems*, 2014.

Dosovitskiy, A., Fischer, P., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1734–1747, 2016.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Graham, B. Fractional max-pooling. *arXiv:1412.6071*, 2014.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

Haasdonk, B. and Burkhardt, H. Invariant kernel functions for pattern analysis and machine learning. *Machine Learning*, 68(1):35–61, 2007.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016.

Heath, M., Bowyer, K., Kopans, D., Moore, R., and Kegelmeyer, P. The digital database for screening mammography. *Digital mammography*, pp. 431–434, 2000.

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

Kuchnik, M. and Smith, V. Efficient augmentation via data subsampling. In *International Conference on Learning Representations*, 2019.

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.

Lee, R. S., Gimenez, F., Hoogi, A., and Rubin, D. Curated breast imaging subset of ddsm. *The Cancer Imaging Archive*, 2016.

Leen, T. From data distributions to regularization in invariant learning. In *Neural Information Processing Systems*, 1995.

Lu, X., Zheng, B., Velivelli, A., and Zhai, C. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association*, 13(5):526–535, 2006.

Maurer, A. and Pontil, M. Empirical Bernstein bounds and sample variance penalization. In *Conference on Computational Learning Theory*, 2009.

Mroueh, Y., Voinea, S., and Poggio, T. A. Learning with group invariant features: A kernel perspective. In *Neural Information Processing Systems*, 2015.

Muandet, K., Fukumizu, K., Dinuzzo, F., and Schölkopf, B. Learning from distributions via support measure machines. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 10–18. Curran Associates, Inc., 2012.

Muandet, K., Fukumizu, K., Sriperumbudur, B., Schölkopf, B., et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Namkoong, H. and Duchi, J. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Neural Information Processing Systems*, 2016.

Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. In *Neural Information Processing Systems*, 2017.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Neural Information Processing Systems*, 2007.

Raj, A., Kumar, A., Mroueh, Y., Fletcher, P. T., and Schölkopf, B. Local group invariant representations via orbit embeddings. *International Conference on Artificial Intelligence and Statistics*, 2017.

Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., and Ré, C. Learning to compose domain-specific transformations for data augmentation. In *Neural Information Processing Systems*, 2017.

Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y., and Muller, X. The manifold tangent classifier. In *Neural Information Processing Systems*, 2011.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Neural Information Processing Systems*, 2016.

Schölkopf, B., Burges, C., and Vapnik, V. Incorporating invariances in support vector learning machines. In *International Conference on Artificial Neural Networks*, 1996.

Sietsma, J. and Dow, R. J. Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79, 1991.

Simard, P., Victorri, B., LeCun, Y., and Denker, J. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *Neural Information Processing Systems*, 1992.

Simard, P., LeCun, Y., Denker, J., and Victorri, B. Transformation invariance in pattern recognition—tangent distance and tangent propagation. *Neural Networks: Tricks of the Trade*, pp. 549–550, 1998.

Tai, K. S., Bailis, P., and Valiant, G. Equivariant transformer networks. In *International Conference on Machine Learning*, 2019.

Teo, C. H., Globerson, A., Roweis, S. T., and Smola, A. J. Convex learning with invariances. In *Neural Information Processing Systems*, 2008.

Tibshirani, R. and Wasserman, L. Nonparametric regression and classification, 2018. URL http://www.stat.cmu.edu/~larry/=sml/NonparametricPrediction.pdf.

Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. Improving music source separation based on deep neural networks through data augmentation and network blending. In *International Conference on Acoustics, Speech and Signal Processing*, 2017.

van der Wilk, M., Bauer, M., John, S., and Hensman, J. Learning invariances using the marginal likelihood. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9960–9970. Curran Associates, Inc., 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Zhao, J., Li, J., Zhao, F., Yan, S., and Feng, J. Marginalized CNN: Learning deep invariant representations. In *British Machine Vision Conference*, 2017.