A Data-Augmentation Is Worth A Thousand Samples

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data-Augmentation (DA) is known to improve performance across tasks and datasets. We propose a method to theoretically analyze the effect of DA and study questions such as: how many augmented samples are needed to correctly estimate the information encoded by that DA? How does the augmentation policy impact the final parameters of a model? We derive several quantities in close-form, such as the expectation and variance of an image, loss, and model's output under a given DA distribution. Up to our knowledge, we obtain the first explicit regularizer that corresponds to using DA during training for non-trivial transformations such as affine transformations, color jittering, or Gaussian blur. Those derivations open new avenues to quantify the benefits and limitations of DA. For example, we show that common DAs require tens of thousands of samples for the loss at hand to be correctly estimated and for the model training to converge. We show that for a training loss to be stable under DA sampling, the model's saliency map (gradient of the loss with respect to the model's input) must align with the smallest eigenvector of the sample variance under the considered DA augmentation, hinting at a possible explanation on why models tend to shift their focus from edges to textures.

1 Introduction

Data Augmentation (DA) is a prevalent technique in training deep learning models [8, 18]. These Deep Networks (DNs) models f_γ, governed by some parameters $\gamma \in \Gamma$, are trained on the train set and expected to generalize to unseen samples (test set). To combat the tendency of DNs to overfit [48], producing a large train-test performance gap, regularization and in particular DA is heavily employed, amongst other mechanisms such as weight-decay [28, 21]. The benefit of DA over those alternatives is that defining input transformations that preserve the semantics of their inputs is a relatively simple task, at least in computer vision or acoustic processing [17, 5, 35]. Furthermore, if the DA is well designed and rich enough, it can effectively bring the number of training samples close to the theoretical limits ensuring that any performance gap between the train and test set vanishes [42] as empirically observed in various scenarios [30, 39]. In fact, DA has proved so useful that novel training methods such as self-supervised learning (SSL) entirely rely on DA [24, 47] to learn a meaningful data representation.

Despite its empirical effectiveness, our understanding of DA has many open questions, three of which we propose to study: **(a)** how do different DAs impact the model's parameters during training?; **(b)** how sample-efficient is the DA sampling, i.e., how many DA samples a model must observe to converge?; and **(c)** how sensitive is a loss/model to the DA sampling and how this variance evolves during training as a function of the model's ability to minimize the loss at hand, and as a function of the model's parameters? Our goal is to analytically derive some preliminary answers and insights into those three areas enabled by a novel image transformation operator that we introduce in Section 3.2 coined Data-Space Transform (DST). DST employs a mathematical formulation of input transformations akin to the one of Chen et al. [4] although slightly more general as we do not

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

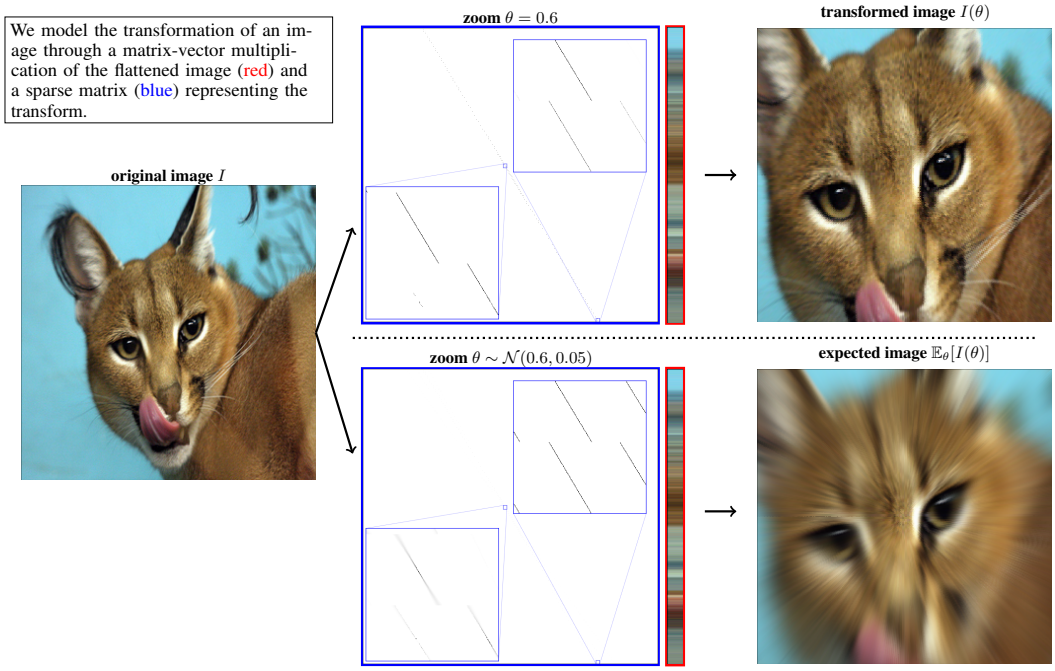

Figure 1: We propose a novel way to express Data Augmentation (DA) analytically that allows us to understand the impact of DA on the learned parameters of a model and quantify DA's sample efficiency. This allows us to compute the analytical expectation and variance of the transformed data (or any function computed with it) in close-form with respect to the transformation parameters.

require the DA to form a group. We study the zoom/translation/shearing/rotation DAs in the main text and Gaussian blur, grayscale, color jitter and random crop in Appendix D. We summarize our contributions below:

(a) we derive the analytical first order moments of augmented samples, and of the losses employing augmented samples (Section 3.3), effectively providing us with the explicit regularizer induced by each DA (Section 3.4)
(b) we quantify the number of DA samples that are required for a model/loss to obtain a correct estimate of the information conveyed by that DA (Section 4.1)
(c) we derive the sensitivity, i.e. variance, of a given loss and model under a DA policy (Section 4.2) leading us to rediscover from first principles a popular deep network regularizer: TangentProp, as being the natural regularization to employ to minimize the loss variance (Section 4.3).

Upshot of gained insights: (a) shows us that the explicit DA regularizer corresponds to a generalized Tikhonov regularizer that depends on each sample's covariance matrix largest eigenvectors; the kernel space of the model's Jacobian matrix aligns with the data manifold tangent space as modeled by the DA **(b)** shows us that the number of augmented samples required for a loss/model to correctly estimate the information provided by a DA of a single sample is on the order of 10^4. Even when combining the information of thousands of samples, we find that the entire train set must be augmented at least $50\times$ for a DA policy to be correctly learned by a model **(c)** quantifies the loss variance as a simple function of the model's Jacobian matrix, and the eigenvectors of the augmented sample variance matrix. Regardless of the model or task-at-hand, the loss sensitivity to random DA sampling goes down as the kernel of a model's Jacobian matrix aligns with the principal directions of the data manifold tangent space; as a by-product, we recover TangentProp as the natural regularizer minimizing the loss sensitivity to DA sampling.

All of our image-space results e.g. expectation and variance of images under DA are exact, no approximation is employed. Results on a model's output/loss are provided both in the case of a linear model and of a nonlinear model. For the former, results are exact, for the latter a third-order Taylor approximations is employed which is common and has been shown to provide accurate enough approximations to safely rely on the insights/quantities obtained e.g. see Sec. A.2 of Wei et al. [44]. All the proofs and implementation details are provided in the appendix. Full codebase will be publicly released upon completion of the review process, a sample is provided in the supplementary material. We emphasize that we are not proposing a tractable alternative to DA but a theoretical analysis on the type of regularization that DA implicitly puts on your model, and how to visualize that regularizer.

2 Background

Existing Explicit Regularizers From Data-Augmentation. It is widely accepted that data-augmentation (DA) regularizes a model towards the transformations that are modeled [28, 27], and this impacts performances significantly, possibly as much as the regularization offered by the choice of DN architecture [9] and optimizer [37].

To gain precious insights into the impact of DA onto the learned functional f_γ, the most common strategy is to derive the *explicit* regularizer that directly acts upon f_γ in the same manner as if one were to use DA during training. This explicit derivation is however challenging and so far has been limited to DA strategies such as additive white noise or multiplicative binary noise applied identically and independently throughout the image, as with dropout [38, 2]. In those settings, various works have studied in the linear regime the relation between such data-augmentation and its equivalence to using Tikhonov [41] or weight decay [49] as in $\min_{\boldsymbol{W}} \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\sigma)} \left[\|\boldsymbol{y}_n - \boldsymbol{W}(\boldsymbol{x}_n + \boldsymbol{\epsilon})\|_2^2\right] = \min_{\boldsymbol{W}} \sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{W}\boldsymbol{x}_n\|_2^2 + \lambda(\sigma)\|\boldsymbol{W}\|_F^2$ [1]. More recently, Wang et al. [43], LeJeune et al. [20] extended the case of additive white noise to nonlinear models and concluded that this DA corresponds to adding an explicit Frobenius norm regularization onto the Jacobian of f_θ evaluated at each data sample.

Going to more involved DA strategies e.g. translations or zooms of the input images is challenging and has so far only been studied from an empirical perspective. For example, Hernández-García and König [14, 13, 15] performed thorough ablation studies on the interplay between DAs and a collection of known explicit regularizers to find correlations between them. It was concluded that weight-decay (the explicit regularizer of additive white noise) does not relate to those more advanced DAs. This also led other studies to suggest that norm-based regularization might be insufficient to describe the implicit regularization of DAs involving advanced image transformations [31]. We debunk this last claim in Sec. 3.3.

Coordinate Space Transformation. Throughout this paper, we will consider a two-dimensional image $I(x, y)$ to be at least square-integrable $I \in L^2(\mathbb{R}^2)$ [22]. Multi-channel images are dealt with by treating each channel as its own single-channel image, as commonly done in practice [8]. As we are interested in practical cases, we will often assume that I has compact support e.g. has nonzero values only within a bounded domain such as $[0, 1]^2$. Visualizing this image thus corresponds to displaying the sampled values of I on a regular grid (pixel positions) of $[0, 1]^2$ [12]. The most common formulation to apply a transformation on the image I to obtain the transformed image T is to transform the image coordinates [34, 45, 25]. That is, a mapping $t : \mathbb{R}^2 \mapsto \mathbb{R}^2$ describes what coordinate $t(u, v)$ of I maps to the coordinate u, v of T as in

$$T(u, v) = I(t(u, v)). \tag{1}$$

This function t often has some parameter θ governing the underlying transformation as in

$$t_\theta(x, y) = [x - \theta_1, y - \theta_2]^T, \;\; t_\theta(x, y) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \;\; t_\theta(x, y) = [\theta_1 x, \theta_2 y]^T,$$

for translation, rotation and zoom respectively. We provide a visual depiction of the zoom transformation applied in coordinate space in Fig. 6, in the appendix. The formulation of Eq. 1 has two key benefits. First, it allows a simple and intuitive design of t to obtain novel transformations. Second, it is computationally efficient as the coordinate-space of images are $2/3$-dimensional. Those benefits have led to e.g. the Spatial Transformer Network [16]. On the other hand, Eq. 1 has one major drawback for our purpose: the exact moments of the transformed image under random θ parameters are not tractable due to the composition of t with the nonlinear mapping I. And as it will become clear in Section 3.1, computing such quantities is crucial for our study.

3 Analytical Moments of Transformed Images Enable Infinite Dataset Training

We first motivate this study by formulating the training process under DA sampling as doing a Monte-Carlo estimate of the true (unknown) expected loss under that DA distribution (Section 3.1). Going beyond this sampling/estimation procedures requires knowledge of the expectation and variance of a transformed sample, motivating the construction of a novel Data-Space Transformation (DTS) (Section 3.2) that allow for a close-form formula of those moments (Section 3.3). From those results, we will be able to remove the need to sample transformed images to train a model, by obtaining the

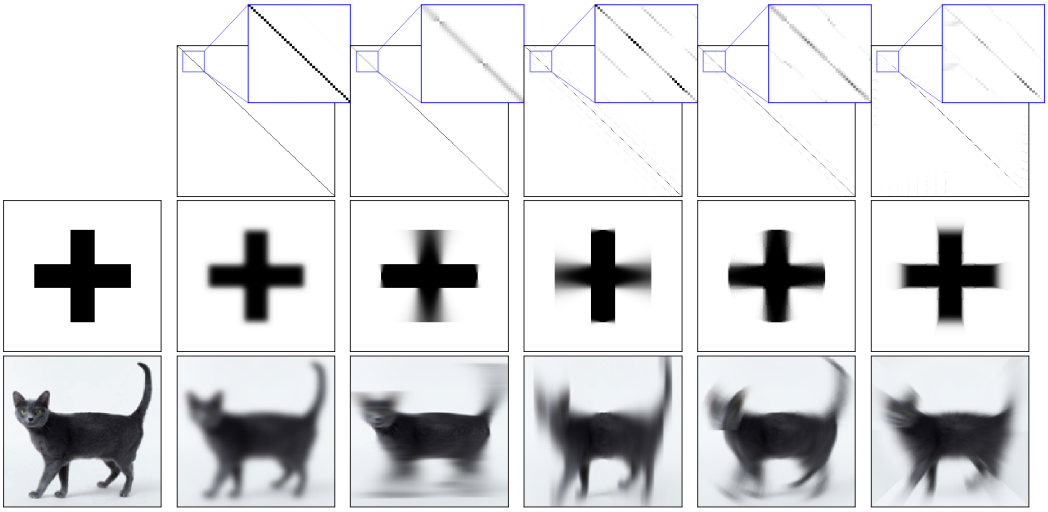

Figure 2: **Top row:** sparse matrix $\mathbb{E}_\theta[\boldsymbol{M}(\theta)]$ producing the expected image via $\mathbb{E}_\theta[\boldsymbol{t}(\theta)] = \mathbb{E}_\theta[\boldsymbol{M}(\theta)]\boldsymbol{x}$ (recall Eq. (6) and the discussion below Theorem 3.3), here with DA distributions of $\mathcal{N}(0, 0.04) \otimes \mathcal{N}(0, 0.04)$, $\mathcal{N}(0, 0.2)$, $\mathcal{N}(0, 0.1)$ and $\mathcal{N}(1, 0.1)$ for each column respectively. This *sparse* matrix is independent of the image and derived in close-form (Theorem 3.3). **Middle & Bottom row:** are two images that depict for each column the "expected image" under each DA. Estimating those images (and any loss employing them) only through sampling the DA augmentations would require tens of thousands of samples (see Fig. 4). *Gaussian blur, grayscale, color jitter and random crop DA are provided in Appendix D along with discussions on how to extend our results to CutOut and MixUp.*

close-form expected loss Section 3.4, under which training can be done directly, in close-form for linear model and with the usual gradient-based learning in general.

3.1 Motivation: Current Data-Augmentation Training Performs Monte-Carlo Estimation

Training a model with DA consists in (i) sampling transformed images at each training iteration for each sample $\boldsymbol{x}_n$ as in $\mathcal{T}_{\theta_n}(\boldsymbol{x}_n)$ with $\theta_n \sim \boldsymbol{\theta}$ a randomly sampled DA parameter e.g. the amount of translation to apply, (ii) evaluating the loss $\mathcal{L}$ on the transformed sample/mini-batch/dataset, and (iii) using some flavor of gradient descent to update the parameters γ of the model f_γ.

This training procedure corresponds to a one-sample Monte-Carlo (MC) estimate [23, 10] of the expected loss

$$\sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{\theta}} \left[\mathcal{L}\left(f_\gamma(\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\right)\right] \approx \sum_{n=1}^{N} \mathcal{L}(f_\gamma(\mathcal{T}_{\theta_n}(\boldsymbol{x}_n)), \tag{2}$$

with θ_n i.i.d samples of $\boldsymbol{\theta}$. In a supervised setting, the loss would also receive a per-sample target $\boldsymbol{y}_n$ as input. Although a one-sample estimate might be insufficient to apply the central limit theorem [33] and guarantee training convergence, the combination of multiple samples in mini-batch training does provide convergence in most cases. Yet, methods such as Self-Supervised Learning that heavily rely on DA, tend to be more sensitive and unstable [40]. To avoid such instabilities, and to incorporate all the DA transformations of $\boldsymbol{x}_n$ into the model's parameter update, one would be tempted to compute the model parameters' gradient on the expected loss (left-hand side of Eq. (2)) (see Section 3.4). Knowledge of the expected loss would also prove useful to measure the quality of the MC estimate (see Section 4.1). Note that even in the least-square setting, such approximation is employed as the expected loss under DAs is not know. Hence, one commonly sample many DA samples, and then apply the least-square formula on this augmented training set, which is highly inefficient. Alternatively, this study will obtain the optimum least-square parameters under DA sampling (Theorem 3.4) i.e. only using the given training set, we directly obtain the parameters that would correspond to sampling infinitely many DA samples and applying the usual least-square formula on this infinite training set.

The close-form expected loss requires knowledge of the expectation and variance of the transformed sample $\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$, hence, we first propose to formulate a novel and tractable augmentation model (Section 3.2) that will allow us to obtain those moments analytically in Section 3.3.

3.2 Proposed Data-Space Transformation

Instead of altering the coordinate positions of an image, as done in the coordinate-space transformation of Eq. (1), we propose to alter the image basis functions.

Going back to the construction of functions, one easily recalls that any (image) function can be expanded into a basis as in $I(u,v) = \int I(x,y)\delta(u-x, v-y)dxdy$ with δ the usual Dirac distributions. Suppose for now that we consider a horizontal translation by a constant θ. Then, one can obtain that translated image via

$$\mathcal{T}_\theta(I)(u,v) = \int I(x,y)\delta(u-x-\theta, v-y)dxdy, \tag{3}$$

hence by moving the basis functions onto which the image is evaluated, rather than by moving the image's coordinate. As the image is now constant with respect to the transformation parameter θ, and as the basis functions have some specific analytical forms, the derivation of the transformed images expectation and variance, under random θ, will become straightforward. Because the transformed image T is now obtained by combining its pixel values (recall Eq. (3)) we coin this transform as Data-Space Transform (DST) and we formally define it below.

Definition 3.1 (Data-Space Transform)**.** We define the data-space transformation of an image $I \in L^2(\mathbb{R}^2)$ producing the transformed image $\mathcal{T}_\theta(I) \in L^2(\mathbb{R}^2)$ as

$$\mathcal{T}_\theta(I)(u,v) = \int I(x,y)h_\theta(u,v,x,y)dxdy, \tag{4}$$

with $h_\theta(u,v,.,.) \in \mathbb{C}_0^\infty(\mathbb{R}^2)$ encoding the transformation.

In Definition 3.1, we only impose for $h_\theta(u,v,.,.)$ to be with compact support. In fact, one should interpret $h_\theta(u,v,.,.)$ as a distribution whose purpose is to evaluate I at a desired (coordinate) position on its domain. This evaluation -depending on the form of $h_\theta(u,v,.,.)$- can return a single pixel-value of the image I at a desired location (as in Eq. (3)), or it can combine multiple values e.g. with $h_\theta(u,v,.,.)$ being a bump function. Additionally, we do not impose any restriction on the invertibility of this mapping i.e. $\mathcal{T}_\theta(I)(u,v)$ can disregard parts of the original image.

Coordinate-space transformations as DSTs. The coordinate-space transformation and the proposed DST (Definition 3.1) act in different spaces: the image coordinates and the image pixel values, respectively. Nevertheless, this does not limit the range of transformations that can be applied to an image. The following statement provides a simple recipe to turn any already employed coordinate-space transformation into a data-space one.

Proposition 3.2. *Any coordinate-space transformation (1) using $t_\theta : \mathbb{R}^2 \mapsto \mathbb{R}^2$ can be expressed as a data-space transformation (4) by setting $h_\theta(u,v,x,y) = \delta(t_\theta(x,y) - [u,v]^T)$.*

We will focus in the main text on zoom/rotation/translation/shearing, and we propose in Appendix D the case of Gaussian blur, color jittering, random crop, along with a discussion on extending our results to CutMix [46] and MixUp [50]. Using Proposition 3.2, we obtain the DST operators $h_\theta(u,v,x,y)$ to be

$$\begin{gathered}\delta(u-x+\theta_1, v-y+\theta_2), \;\; \delta(u-x-\theta_1 y, v-y-\theta_2 x), \;\; \delta(u-\theta x, v-\theta y), \\ \delta(u-\cos(\theta)x+\sin(\theta)y, v-\sin(\theta)x-\cos(\theta)y),\end{gathered} \tag{5}$$

for the vertical/horizontal translation, vertical/horizontal shearing, zoom and rotation respectively. Before focusing on the analytical moments of the data-space transformed samples, we describe how those operators are applied in a discrete setting.

Discretized version. We now describe how any DST of a discrete image, flattened as a vector, can be expressed as a matrix-vector product with the matrix entries depending on the employed DA and its parameter θ, as in Figs. 1 and 2. The functional form of the proposed transform in Eq. (4) producing the target image at a specific position $\mathcal{T}_\theta(I)(u,v)$ is linear in the image I. The continuous integral over the image domain is replaced with a summation with indices based on the desired

sampling/resolution of I. Expressing linear operators as matrix-vector products will greatly ease our development, we will denote by $\boldsymbol{x} \in \mathbb{R}^{hw}$ the flattened $(h \times w)$ discrete images I. Hence, our data-space transformation, given some parameters θ, takes the form of

$$\boldsymbol{t}(\theta) = \boldsymbol{M}(\theta)\boldsymbol{x}, \tag{6}$$

with $\boldsymbol{t}(\theta) \in \mathbb{R}^{hw}$ the flattened transformed image (which can be reshaped as desired) and $\boldsymbol{M}(\theta) \in \mathbb{R}^{hw \times hw}$ the matrix whose rows encode the discrete and flattened $h_\theta(u, v, ., .)$. For example, and employing a uniform grid sampling for illustration, $\boldsymbol{M}(\theta)_{i,j} = h_\theta(i//w, i\%w, j//w, j\%w)$ with $//$ representing the floor division and $\%$ the modulo operation. For the case of multi-channel images, we consider without loss of generality the application of the matrix-vector operation on each channel separately. We depict this operation along with the exact form of $\boldsymbol{M}(\theta)$ for the case of the zoom transformation in Fig. 1.

3.3 Analytical Expectation and Variance of Transformed Images

The above construction (Definition 3.1) turns out to make the analytical form of the first two moments of an augmented sample much simpler to derive. As this derivation is at the core of our main contribution, we propose a step-by-step derivation of $\mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{T}_{\boldsymbol{\theta}}(I)\right]$ for the case of horizontal translation (as in Eq. (3)).

Let's consider again the continuous model (the discrete version is provided after Theorem 3.3). Using Fubini's theorem to switch the order of integration and recalling Eq. (4), we have that

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{T}_{\boldsymbol{\theta}}(I)(u, v)\right] = \int I(x, y)\mathbb{E}_{\boldsymbol{\theta}}\left[h_{\boldsymbol{\theta}}(u, v, x, y)\right]dxdy. \tag{7}$$

Using the definition for h_θ from Eq. (5) for horizontal translation, $\mathbb{E}_{\boldsymbol{\theta}}\left[h_{\boldsymbol{\theta}}(u, v, x, y)\right]$ becomes

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\delta(u - x - \boldsymbol{\theta}, v - y)\right] = \int \delta(u - x - \theta, v - y)p(\theta)d\theta = p(u - x)\delta(v - y),$$

with p the density function of $\boldsymbol{\theta}$ prescribing how the translation parameter is distributed. As a result, in this univariate translation case, the expected augmented image at coordinate (u, v) is given by

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{T}_{\boldsymbol{\theta}}(I)(u, v)\right] = \int I(x, y)p(u - x)\delta(v - y)dxdy = \int I(x, v)p(u - x)dx, \tag{8}$$

which can be further simplified into $\mathbb{E}_{\boldsymbol{\theta}}\left[\mathcal{T}_{\boldsymbol{\theta}}(I)(., v)\right] = I(., v) \star p$. Hence, the expected x-translated image is the convolution (on the x-axis only) between the original image I and the univariate density function p. We formalize this for the transformations of Eq. (5) below.

Theorem 3.3. *The analytical form of $\mathbb{E}_{\boldsymbol{\theta}}\left[h_{\boldsymbol{\theta}}(u, v, x, y)\right]$, used to obtain the expected transformed image (recall Eq. (7)) is given by*

$$p(u - x, v - y) \;\; \textit{(translation)}, \quad \textit{and} \quad p(u/x)\delta(u/x - v/y) \;\; \textit{(rotation)}, \tag{9}$$

other cases and $\mathbb{E}_{\boldsymbol{\theta}}[\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)^T]$ are deferred to the proof. (Proof in Appendix G.)

Notice for example how the expected image under random 2-dimensional translations with (2-dimensional) density p is given by the convolution $I \star p$, providing a new portal to study and interpret convolutions with nonnegative, sum-to-one filters p. Again, and as per Eq. (6), the discretized version of the expected image takes the form of $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{t}(\boldsymbol{\theta})] = \mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{M}(\boldsymbol{\theta})]\boldsymbol{x}$ with the entries of $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{M}(\boldsymbol{\theta})]$ given by discretizing Eq. (9). We depict this expected matrix for various transformations as well as their application onto two different discrete images in Fig. 2, and we now proceed on deriving the left-hand side of Eq. (2) i.e. the explicit DA regularizer.

3.4 The Explicit Regularizer of Data-Augmentations

To keep notations as light as possible, we first consider a linear regression model with Mean Squared Error (MSE), the nonlinear case will be studied in Section 4. In that setting, the expected loss under DA sampling (recall Eq. (2)) becomes

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{\theta}}\left[\|\boldsymbol{y}_n - \boldsymbol{W}\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - \boldsymbol{b}\|_2^2\right], \tag{10}$$

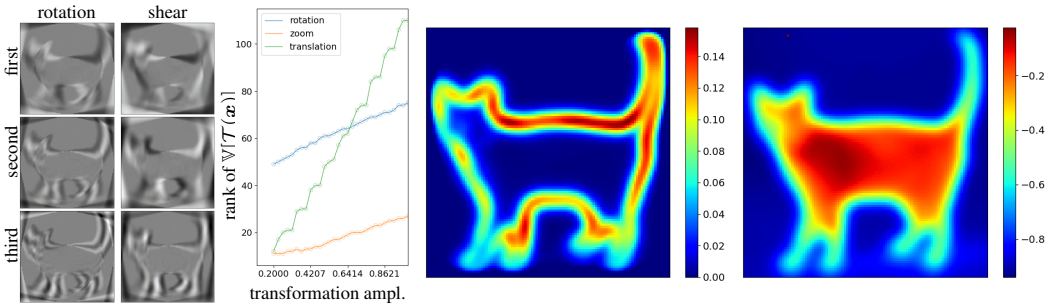

Figure 3: **First column:** top eigenvectors ($\boldsymbol{Q}$ matrix in Theorem 3.4) of the sample variance matrix $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ for rotation/shear augmentations with $\mathcal{U}(-15°, 15°)$ and $\mathcal{U}(-15°, 15°) \otimes \mathcal{U}(-15°, 15°)$ respectively. As per Theorem 4.1, training aligns the model's Jacobian matrix kernel to the largest eigenvectors of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ to reduce the loss variance under the respective DA sampling. **Second column:** number of nonzero eigenvalues of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ (nonzero elements in Λ in Theorem 3.4) for increasing transformation amplitude (the datum dimension is $114 \times 114 \times 3$). We observe that the dimension of the subspace spanned by augmented images constraint increases linearly with the amplitude of the transformation. **Third column:** Pixel variance i.e. diagonal of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ reshapes as an image. **Fourth column:** pixel covariance between one background pixel and all other pixels, i.e. 700th row of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ reshaped as an image for the cat of Fig. 2 under translations ($\mathcal{N}(0, 0.1) \otimes \mathcal{N}(0, 0.1)$). As per Theorem 3.4, the variance of the pixel values, seen on the left, being much higher for the edges of the cat than for its interior body/texture pushes the model to focus on the cat texture, a phenomenon empirically observed in deep networks [7].

with $\boldsymbol{x}_n \in \mathbb{R}^D, n = 1, \ldots, N$ the input (flattened) n^{th} image I_n, $\boldsymbol{y}_n \in \mathbb{R}^K$ the nth target vector, and $\boldsymbol{W} \in \mathbb{R}^{K \times D}, \boldsymbol{b} \in \mathbb{R}^K$ the model's parameters. Recall from Section 3.1 that to learn $\boldsymbol{W}$ under DA, the current method consists in performing a MC estimation. Instead, let's derive (detailed derivation in Appendix E) the exact loss of Eq. (10) as a function of the sample mean and variance under the consider DA. We will drop the $\boldsymbol{\theta}$ subscript for clarity to obtain

$$\mathcal{L} = \sum_{n=1}^{N} \|\boldsymbol{y}_n - \boldsymbol{W}\mathbb{E}\left[\mathcal{T}(\boldsymbol{x}_n)\right] - \boldsymbol{b}\|_2^2 + \|\boldsymbol{W}\boldsymbol{Q}(\boldsymbol{x})\Lambda(\boldsymbol{x})^{\frac{1}{2}}\|_F^2, \tag{11}$$

with the spectral decomposition $\boldsymbol{Q}(\boldsymbol{x})\Lambda(\boldsymbol{x})\boldsymbol{Q}(\boldsymbol{x})^T = \mathbb{V}\left[\mathcal{T}(\boldsymbol{x})\right]$. The right term in Eq. (11) is the explicit DA regularizer. It pushes the kernel space of $\boldsymbol{W}$ to align with the largest principal directions of the data manifold tangent space, as modeled by the DA. In fact, the largest eigenvectors in $\boldsymbol{Q}(\boldsymbol{x})$ represent the principal directions of the data manifold tangent space at $\boldsymbol{x}$, as encoded via $\mathbb{V}\left[\mathcal{T}(\boldsymbol{x})\right]$.

We propose in Fig. 3 visualization of $\boldsymbol{Q}$ and Λ for different DAs, illustrating how each DA policy impacts the model's parameter $\boldsymbol{W}$ through the regularization of Eq. (11). The knowledge of $\mathbb{E}\left[\mathcal{T}(\boldsymbol{x})\right]$ and $\mathbb{V}\left[\mathcal{T}(\boldsymbol{x})\right]$ from Theorem 3.3 finally enables to train a (linear) model on the true expected loss (Eq. (11)) as we formalize below.

Theorem 3.4. *Training a linear model with MSE and infinite DA sampling is equivalent to minimizing Eq.* (11) *and produces the optimal $\boldsymbol{W}^*$ model's parameter*

$$\boldsymbol{W}^* = \left(\sum_n (\boldsymbol{y}_n - \boldsymbol{b})\mathbb{E}[\mathcal{T}(\boldsymbol{x}_n)]^T\right)\left(\sum_n \mathbb{E}[\mathcal{T}(\boldsymbol{x}_n)]\mathbb{E}[\mathcal{T}(\boldsymbol{x}_n)]^T + \mathbb{V}[\mathcal{T}(\boldsymbol{x}_n)]\right)^{-1}. \tag{12}$$

Not surprisingly, whenever the DA is identity, $\mathbb{E}[\mathcal{T}(\boldsymbol{x}_n)] = \boldsymbol{x}_n, \mathbb{V}[\mathcal{T}(\boldsymbol{x}_n)] = \boldsymbol{0}$ and thus Eq. (12) recovers exactly the standard least-square solution. We visualize $\mathbb{V}[\mathcal{T}(\boldsymbol{x}_n)]$ for the translation DA in Fig. 3. The same line of result can be derived in the nonlinear setting by assuming that the DA is restricted to small transformations. In that case, one leverages a truncated Taylor approximation of the nonlinear model[1] and recovers that (local) DA applies the same regularization as in Eq. (11) but with the model's Jacobian matrix $\boldsymbol{J}f_\gamma(\boldsymbol{x}_n)$ in-place of $\boldsymbol{W}$ (more details in Section 4.3).

Given the above derivation of the exact expected loss, we can now turn to precisely measure how accurate is the MC estimate commonly used to train models under DA sampling.

4 Data-Augmentation Sampling Efficiency and Loss Sensitivity

In this section we study the convergence of the MC estimate (Section 4.1), and provide variance analysis of that estimate as a function of the model's Jacobian matrix and the sample variance

[1] as commonly done, see e.g. Sec. A.2 from Wei et al. [44] for justification and approximation error analysis

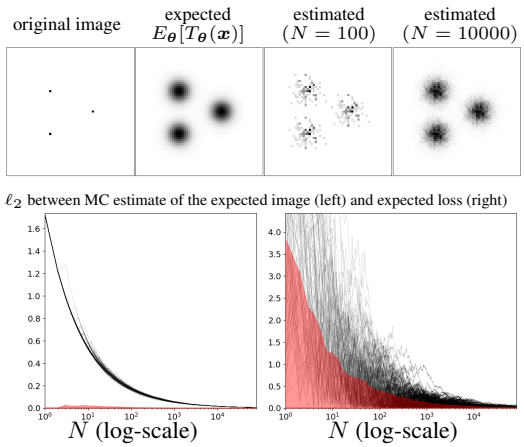

Figure 4: Analytical expected synthetic image under translation ($\mathcal{N}(0, 0.1) \otimes \mathcal{N}(0, 0.1)$) against its N-sample Monte-Carlo estimate (**top row**), ℓ_2 distance between the true and estimated images (**bottom left**) and the true and estimated MSE loss with a random Gaussian $\boldsymbol{y}, \boldsymbol{W}$ (**bottom right**). In red is depicted the standard deviation of the independent Monte-Carlo runs. Clearly we observe that even on a simple (64×64) image and using the translation transformation, thousands of sampled images are necessary to provide an accurate estimate of the loss at hand. That is, sampling based data-augmentations are a rather inefficient medium to employ for injecting prior information into a model as this prior information will only emerge after tens of thousands of images have been sampled.

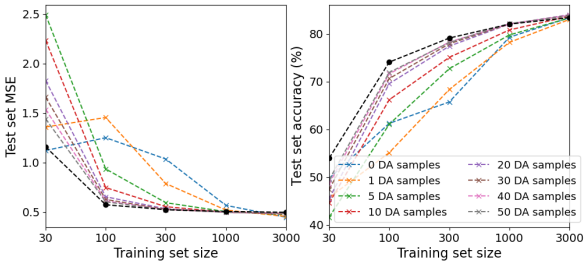

Figure 5: MNIST with linear model and translation DA with varying number of samples (**colored lines**) against the use of the expected loss (**black line**). We depict the MSE loss (**left**) and the accuracy (**right**) on the test set. At first, even a $50\times$ DA-based increase in dataset size fails to provide the performances reached by employing the analytical expected loss. As the training size increases, as the DA samples get redundant with the added samples closing the performance gap (the case of nonlinear model is provided in Fig. 7).See Tables 2 to 5 for additional table of results with standard deviation.

eigenvectors (Section 4.2), concluding by the (re-)discovery of TangentProp from first principles (Section 4.3). Those results hold regardless of the type of DA employed. Additionally, the analysis is done on a single sample to ease notation, in the i.i.d. setting, simply sum over all the training samples to analyze the training set.

4.1 Empirical Monte-Carlo Convergence of Transformed Images

Given the close-form *average image* and *average loss*, we empirically measure how efficient is the MC estimation from Eq. (2).

We first propose in Fig. 4 a constructed (64×64) image for which we compute the expected loss (Eq. (11)) and the MC estimate (right-hand side of Eq. (2)). Surprisingly, we obtain that even for such a simple image and translation DA, between 1000 and 10000 samples are required to correctly estimate the MSE loss from the augmented samples. In a more practical scenario, one could rightfully argue that the combination of the DA samples from different images allows to obtain a better estimate with a smaller amount of augmentations per sample. Hence, we provide in Fig. 5 that experiment using a linear model on MNIST [19] with varying train set size. We observe that as the number of samples grows as the required number of augmentation per sample reduces. Nevertheless, even with thousands of samples, at least 50 augmentations per sample are required to provide an accurate estimate (reproduction with nonlinear models given in Fig. 7).

We now propose to specifically quantify the sensitivity of the MC estimate to DA sampling.

4.2 Loss Sensitivity Under Data-Augmentation Sampling in the Linear and Nonlinear Regime

Recalling Section 3.1, current DA training is performed on a MC estimate of the loss. The estimator's variance [26] is proportional to the variance of the quantity being estimated: $\mathbb{V}[(\mathcal{L} \circ f)(\mathcal{T}(\boldsymbol{x}))]$. We now characterize when, and why, would an MC estimator converge for a given model.

By leveraging the delta method [6, 29] i.e. a truncated Taylor expansion of the model and loss function mapping as $\mathcal{L} \circ f$, we have

$$\mathbb{V}[(\mathcal{L} \circ f)(\mathcal{T}(\boldsymbol{x}))] \approx \|\nabla(\mathcal{L} \circ f)(\mathbb{E}[\mathcal{T}(\boldsymbol{x})])\|^2_{\mathbb{V}[\mathcal{T}(\boldsymbol{x})]}, \tag{13}$$

with $\|\boldsymbol{u}\|^2_{\boldsymbol{A}} \triangleq \boldsymbol{u}^T \boldsymbol{A} \boldsymbol{u}$. Noticing that $\mathbb{V}[\mathcal{T}(\boldsymbol{x})] = \mathbb{E}[\mathcal{T}(\boldsymbol{x})\mathcal{T}(\boldsymbol{x})^T] - \mathbb{E}[\mathcal{T}(\boldsymbol{x})]\mathbb{E}[\mathcal{T}(\boldsymbol{x})]^T$ and using the closed-form moments of augmented samples from Theorem 3.3, it is possible to write out Eq. (13) explicitly for model and DA specific analysis. We visualize $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$ in Fig. 3.

Linear regression case. To gain some insight into Eq. (13), let's first consider the linear regression case leading to $\mathbb{V}[(\mathcal{L} \circ f)(\mathcal{T}(\boldsymbol{x}))] \approx \|\boldsymbol{y} - \boldsymbol{W}\mathbb{E}[\mathcal{T}(\boldsymbol{x})] - \boldsymbol{b}\|^2_{\boldsymbol{W}\mathbb{V}[\mathcal{T}(\boldsymbol{x})]\boldsymbol{W}^T}$. Hence, the estimated loss variance depends on the model predicting the correct output when observing the expected sample $\mathbb{E}[\mathcal{T}(\boldsymbol{x})]$, and on the smallest right singular vectors of $\boldsymbol{W}$ to align with the largest eigenvectors of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$, echoing our observation below Eq. (11).

General case. Since the result from Eq. (13) holds in the general setting, we can generalize the above observation to the more general setting.

Theorem 4.1. *The variance of the loss's MC estimate (recall Eq.* (2)*) for an input $\boldsymbol{x}_n$ goes to* 0 *if the loss gradient $\nabla\mathcal{L}(f(\mathbb{E}[\mathcal{T}(\boldsymbol{x})]))$ goes to* 0*, or if the kernel of the model's Jacobian matrix $\boldsymbol{J}f(\mathbb{E}[\mathcal{T}(\boldsymbol{x})])$ aligns with the largest eigenvectors of the sample variance $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$.*

The proof of the above statement simply starts from Eq. (13) and applies the Cauchy-Schwarz inequality two times to obtain

$$\mathbb{V}[(\mathcal{L} \circ f)(\mathcal{T}(\boldsymbol{x}))] \leq \|\nabla\mathcal{L}(f(\mathbb{E}[\mathcal{T}(\boldsymbol{x})]))\|_2^4 \times \|\boldsymbol{J}f(\mathbb{E}[\mathcal{T}(\boldsymbol{x})])\boldsymbol{Q}(\boldsymbol{x})\Lambda^{\frac{1}{2}}(\boldsymbol{x})\|_F^4, \tag{14}$$

with $\mathbb{V}[\mathcal{T}(\boldsymbol{x})] = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T$. From Eq. (14), it is quite realistic to assume that the model's Jacobian does not collapse to 0 (this would not minimize the training loss in general). Hence, the most realistic case of Theorem 4.1 concerns the alignment between the model's Jacobian matrix kernel space and the top eigenvectors of $\mathbb{V}[\mathcal{T}(\boldsymbol{x})]$. However, for models like ResNet DNs [11], the loss variance can only reduce to 0 if the model minimizes the loss at the expected input since the Jacobian matrix is always full-rank. *We recall that the above relied on the delta method to approximate the intractable LHS of Eq.* (13) *(see footnote on Page 7).*

4.3 Explicit Loss Sensitivity Minimization Provably Recovers TangentProp

The expected loss (Eq. (2)) has been derived in the linear regression setting (Eq. (11)). But in a more general scenario, and as discussed below Theorem 3.4, this expectation might not be tractable and thus needs to be approximated e.g. based on a Taylor expansion of the loss and model. Alternatively to using the approximated expectation, one could employ the usual MC estimate of the expectation (right-hand side of Eq. (2)), and leverage the MC estimator variance obtained in Eq. (14) as a regularizer. We show here that both approaches are equivalent, and recover a popular regularizer known as TangentProp [36]. Although various extensions of TangentProp have been introduced [3, 32] no principled derivation of it has yet been proposed. Using the same argument as in Section 4.2, the expectation of a nonlinear transformation of a random variable ($\mathcal{T}(\boldsymbol{x}_n)$) can be approximated from the expectation of the Taylor expansion (detailed derivations in Appendix F) from which an upper-bound is obtained by applying the Cauchy-Schwarz inequality

$$\mathbb{E}[(\mathcal{L} \circ f)(\mathcal{T}(\boldsymbol{x}))] \leq (\mathcal{L} \circ f)(\mathbb{E}[\mathcal{T}(\boldsymbol{x})]) + \kappa(\boldsymbol{x}) \underbrace{\|\boldsymbol{J}f(\mathbb{E}[\mathcal{T}(\boldsymbol{x})])\boldsymbol{Q}(\boldsymbol{x})\Lambda(\boldsymbol{x})^{\frac{1}{2}}\|_F^2}_{\text{TangentProp regularization}},$$

with $\kappa(\boldsymbol{x}) \geq 0$ and the notations from Eq. (14). As a result, the TangentProp regularizer naturally appears when using a Taylor approximation of the expected loss, and it corresponds to adding an explicit loss variance regularization term (compare the TangentProp with Eq. (14)). We thus obtained from first principles that TangentProp emerges naturally when considering the second order Taylor approximation of the expected loss given a DA.

5 Conclusions and Limitations

In this paper, we proposed a novel set of mathematical tools (Sections 3.2 and 3.3) under which it is possible to study DA and to provably answer some of the open questions around the efficiency and impact of DA to train a model. We first obtained the explicit regularizer produced by different DAs in Section 3.4. This led to the following observation: the kernel space of the $\boldsymbol{W}$ matrix is pushed to align with the largest eigenvectors of the sample covariance matrix. This was then studied in a more general setting in Section 4.2 for nonlinear models when characterizing the loss variance under DA sampling. We also observed that Monte-Carlo sampling of transformed images is highly inefficient (Section 4.1), even if similar training samples combine their underlying information within a dataset. Lastly, those derivations led us to provably derive a known regularizer —TangentProp— as being the natural minimizer of a model's loss variance. The main limitation of the current analysis concerns nonlinear models for which we rely on a third-order Taylor expansion. We hope to explore in more details the impact of DA on nonlinear models, possibly with a per-architecture analysis in future work.

References

[1] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26:2814–2822, 2013.

[2] Xavier Bouthillier, Kishore Konda, Pascal Vincent, and Roland Memisevic. Dropout as data augmentation. *arXiv preprint arXiv:1506.08700*, 2015.

[3] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, pages 416–422, 2001.

[4] Shuxiao Chen, Edgar Dobriban, and Jane Lee. A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33:21321–21333, 2020.

[5] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23 (9):1469–1477, 2015.

[6] Joseph L Doob. The limiting distributions of certain statistics. *The Annals of Mathematical Statistics*, 6(3):160–169, 1935.

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bygh9j09KX`.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[9] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

[10] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[12] Paul Heckbert. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics*, 16(3):297–307, 1982.

[13] Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*, 2018.

[14] Alex Hernández-García and Peter König. Do deep nets really need weight decay and dropout? *arXiv preprint arXiv:1802.07042*, 2018.

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

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[17] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*, 2015.

[18] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[19] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[20] Daniel LeJeune, Randall Balestriero, Hamid Javadi, and Richard G Baraniuk. Implicit rugosity regularization via data augmentation. *arXiv preprint arXiv:1905.11639*, 2019.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[22] Stephane G Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2091–2110, 1989.

[23] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[24] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[25] Ramakrishnan Mukundan, SH Ong, and Poh Aun Lee. Image analysis by tchebichef moments. *IEEE Transactions on image Processing*, 10(9):1357–1364, 2001.

[26] Mark EJ Newman and Gerard T Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.

[27] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[28] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[29] Gary W Oehlert. A note on the delta method. *The American Statistician*, 46(1):27–29, 1992.

[30] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[31] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *arXiv preprint arXiv:2005.06398*, 2020.

[32] Salah Rifai, Yann N Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. *Advances in neural information processing systems*, 24:2294–2302, 2011.

[33] Murray Rosenblatt. A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, 42(1):43, 1956.

[34] Alexander A Sawchuk. Space-variant image restoration by coordinate transformations. *JOSA*, 64(2):138–144, 1974.

[35] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[36] Patrice Simard, Bernard Victorri, Yann LeCun, and John S Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *NIPS*, volume 91, pages 895–903. Citeseer, 1991.

[37] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

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

[39] Arwa Mohammed Taqi, Ahmed Awad, Fadwa Al-Azzo, and Mariofanna Milanova. The impact of multi-optimizers and data augmentation on tensorflow convolutional neural network performance. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 140–145. IEEE, 2018.

[40] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[41] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

[42] Michael Unser. A representer theorem for deep neural networks. *J. Mach. Learn. Res.*, 20(110): 1–30, 2019.

[43] Zichao Wang, Randall Balestriero, and Richard Baraniuk. A max-affine spline perspective of recurrent neural networks. In *International Conference on Learning Representations*, 2018.

[44] Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout, 2020.

[45] George Wolberg and Siavash Zokai. Robust image registration using log-polar transform. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 493–496. IEEE, 2000.

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[47] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[48] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

[49] Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

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

Checklist

1. For all authors...

(a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We have precisely given the scope of our results in the end of the introduction. Additionally, we recall the limitations and assumptions of our results within each section carefully.

(b) Did you describe the limitations of your work? [Yes] The main limitations comes from the approximation done for nonlinear models which we have repeatedly highlight in both the introductions and the corresponding sections. We also highlight those in the conclusion along with future work.

(c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not see any potential negative societal impact for this work

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] we have carefully given the assumptions each result relies on. Especially, at the end of the introduction, we have highlighted a reference paper where it was shown that our assumptions are common in the literature, and that they are good enough to gain insights into nonlinear models.

(b) Did you include complete proofs of all theoretical results? [Yes] Each formal statement and derivation is linked to a proof in the appendix. We also provide details derivations in the appendix of all the equations we have obtained in the main paper, even if they are not part of a formal result.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide samples of the code to reproduce figures. The entire codebase will be released publicly upon completion of the review-process

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We only explore MNIST and provided the training details in the caption of the corresponding figure

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We did not provide error bars but multiple runs with various settings

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The employed experiment is small scale and can be performed without GPU

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] MNIST only

(b) Did you mention the license of the assets? [No]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]