Dreaming More Data: Class-dependent Distributions over Diffeomorphisms for Learned Data Augmentation

Søren Hauberg
Section for Cognitive Systems
Technical University of Denmark
sohau@dtu.dk

Oren Freifeld
Sensing, Learning and
Inference Group
MIT CSAIL
freifeld@csail.mit.edu

Anders Boesen Lindbo Larsen
Image Analysis and Computer
Graphics Section
Technical University of Denmark
abll@dtu.dk

John W. Fisher III
Sensing, Learning and Inference Group
MIT CSAIL
fisher@csail.mit.edu

Lars Kai Hansen
Section for Cognitive Systems
Technical University of Denmark
lkai@dtu.dk

Abstract

Data augmentation is a key element in training high-dimensional models. In this approach, one synthesizes new observations by applying pre-specified transformations to the original training data; e.g. new images are formed by rotating old ones. Current augmentation schemes, however, rely on manual specification of the applied transformations, making data augmentation an implicit form of feature engineering. With an eye towards true *end-to-end learning*, we suggest *learning the applied transformations on a per-class basis*. Particularly, we align image pairs within each class under the assumption that the spatial transformation between images belongs to a large class of diffeomorphisms. We then learn a class-specific probabilistic generative models of the transformations in a Riemannian submanifold of the Lie group of diffeomorphisms. We demonstrate significant performance improvements in training deep neural nets over manually-specified augmentation schemes. Our code and augmented datasets are available online.

1 Introduction

Variation in classification datasets typically reflects a variety of physical processes. Some are related to label differences and others inject variability which is irrelevant to the labeling. In visual object recognition we face different objects, but also different locations, postures, variable illumination conditions, and so forth. Mathematically this means that the image label is invariant to certain transformations of the image. Knowledge of these transformations can be used to drastically improve classification performance, either by picking invariant features or by augmenting the dataset with new observations generated by applying random transformations to the training data. This form of *data augmentation* is a rather old idea [6, 42], which is still considered part of "best practices" [43] for many tasks. Data augmentation was a critical component in the landmark image-classification paper by Krizhevsky et al. [30]: *"Without this scheme, our network suffers from substantial overfitting, which would have forced us to use much smaller networks"*.

In practice, data augmentation is a manual process, where a human specifies a small set of transformations for which an image classification task is believed to be invariant; for image classification tasks, these are most commonly chosen to be simple linear transformations such as *translations*, *rotations* and *scaling*. In essence, this manual process is a form of *feature engineering*, where the data itself plays only a secondary role. Moreover, as current augmentation schemes are manually specified, the same scheme is often used within all classes. This is potentially troublesome, e.g. while full *rotational invariance* is poorly suited when differ-

entiating images of digits "6" and "9", it may still help differentiate other digits. These concerns suggest that it is beneficial to *learn augmentation schemes* from training data rather than relying on manual specification, thus, getting closer to an *end-to-end learning scheme*.

We consider an unsupervised approach for learning image deformations that naturally appear within different classes. For each class, we align images in a pairwise fashion, assuming the latent spatial transformation between them is a (C^1) diffeomorphism. This gives a set of diffeomorphisms represented on a finite-dimensional nonlinear Riemannian manifold. We show that our data has a simple closed-form mean on this manifold, and approximate the distribution of diffeomorphisms with a per-class multivariate normal distribution in the tangent space at the mean. We then generate new data by sampling first an image from the training data, and second a diffeomorphism from the learned distribution. Applying the diffeomorphism to the image gives a new data point. With this data we train both a multilayer perceptron and a convolutional neural net. In both cases we observe a significant improvement over manually specified augmentation schemes. This is particularly evident for small datasets. Both the code and generated data are available online.[1]

2 Background and Related Work

We focus the discussion on image classification, though the fundamental ideas generalize to other domains. Let $T^{\boldsymbol{\theta}}$ be a spatial transformation, parametrized by $\boldsymbol{\theta}$, such that if an image $\boldsymbol{x}$ has label y, then so does $\boldsymbol{x} \circ T^{\boldsymbol{\theta}}$. For example, translating an object in an image does not change its label. If $T^{\boldsymbol{\theta}}$ is known then this conveys strong prior information, which may improve a classifier. This is a key consideration when training deep models, cf. [19].

One strategy for encoding knowledge of $T^{\boldsymbol{\theta}}$ is to design features that are invariant to $T^{\boldsymbol{\theta}}$, i.e. $\mathcal{F}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x} \circ T^{\boldsymbol{\theta}})$, where $\mathcal{F}$ denotes the feature map. Equivalently, invariant kernels [11] can be derived. This idea lies at the heart of Grenander's *Pattern Theory* [21], which encodes invariances by describing the data itself as transformations acting on reference objects. Alternatively, all instantiations of $T^{\boldsymbol{\theta}}$ may be applied to each observation to produce orbits [20, 31] which can then be matched. While mathematically elegant, these approaches tend to be laborious and computationally expensive, which hinder their applicability. Moreover, the resulting transformations generally have limited

expressiveness (partly to keep computations tractable, the transformation family is usually restricted to be simple; e.g., affine transformations).

To counter this, many approaches are only approximately invariant to the transformations. For example, scale invariance is approximated via image-pyramid representations [17, 28], assuming a known set of scales which the classifier must be invariant to. The classic *Tangent Prop* [42] locally linearizes $T^{\boldsymbol{\theta}}$ with finite differences, and forces the back-propagated gradient of a neural net to respect the invariance. The linearization, however, implies that invariance can be learned with respect to only infinitesimal transformations. General linear invariances are also used for restricted Boltzmann machines [29, 44], but again the linearity implies that the invariance is only infinitesimal.

A widely-used approach is to synthesize new observations by applying the known $T^{\boldsymbol{\theta}}$ to the training data, and then train a classifier on the augmented data set [6, 30, 32, 43]. Eigen et al. [16] consider multiple invariances including length scaling, rotation, translation, horizontal flip and color scaling in a deep multiscale network for single-image depth estimation. In work on speech recognition, Jaitly and Hinton [27] apply *Vocal Tract Normalization (VTLN)* as a way to artificially transform utterances of one speaker to the voice of another. The transform is applied in the spectral domain and corresponds in the VTLN model to a simple parameterized warp of the frequency axis [27]. In astrophysics, in the context of galaxy redshift prediction, Hoyle et al. [26] augmented data by use of redshift models. The *Infinite MNIST* data set [32] was generated by considering horizontal and vertical translations, rotations, horizontal and vertical scalings, hyperbolic transformations, and random Gaussian perturbations. Using the ideas of Tangent Prop, infinitesimal transformations are then applied to the training data to produce a total of 8 million observations. This is the current most elaborate augmentation strategy.

Semi-supervised learning [10] in data with a neighborhood graph can be related to model-based data augmentation, the modeling assumption being that neighbors share the same label; for examples in text classification, see [33]. In an interesting twist on data augmentation, Dosovitskiy et al. [14] create various augmented data, and learn features by training networks to *distinguish between* the differently augmented datasets.

The above approaches all assume that the transformations $T^{\boldsymbol{\theta}}$ are known *a priori*, which is generally not the case. The specification of *which* transformation should be considered is an implicit form of feature engineering, and it is worth investigating whether the transfor-
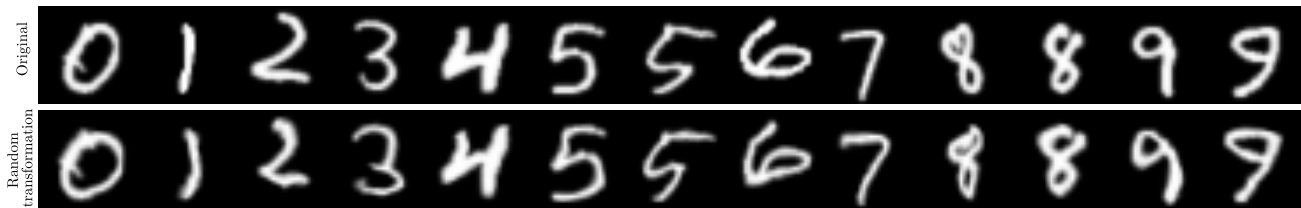
---

[1] http://www2.compute.dtu.dk/~sohau/augmentations/

Figure 1: Example transformations. *Top:* original images. *Bottom:* random transformations from our model applied to the original images.

mations themselves should be learned from data. *This is the objective of the current manuscript.*

Transformations between observations can also be used to define a similarity measure between observations by quantifying the "size" of the transformation. This was used with a *k*-nearest neighbors classifier by Amit et al. [2] to create the current state-of-the-art MNIST classifier relying on only a low number of training images. Similarly, Duchenne et al. [15] use the size of transformations to build a kernel for SVM classification. To speed up the estimation of the transformations, both Winn & Jojic [48] and Hariharan et al. [22] learn models of transformations to guide the search.

3 Diffeomorphisms: Representation, Inference, and Learning

To learn a model $p(T^{\boldsymbol{\theta}}|y)$ of the transformations within each class, we need a well-behaved and sufficiently-expressive mathematical representation of transformations. To avoid a manual specification of the transformations of interest (such as rotations, translations, etc.), we focus our attention on a large class of first-order diffeomorphisms, i.e. we consider $T^{\boldsymbol{\theta}}$ that are once differentiable, and whose inverse, $(T^{\boldsymbol{\theta}})^{-1}$, exists and is differentiable as well.

3.1 Representing Diffeomorphisms

The space of *all* diffeomorphisms is an infinite-dimensional Lie group [4]. To reduce the implied computational burden when aligning images, we restrict our attention to a large finite-dimensional subset of this Lie group. Here we rely on recent developments in the image transformation literature [18]:

These developments specify a transformation through the integration of a *velocity field with a simple structure* that, while being highly-expressive, leads to fast and accurate computations, thus rendering inference tractable. Let $\Omega \subset \mathbb{R}^2$ denote the image domain and let $\mathcal{P}$ be a triangular tessellation of Ω (see Fig. 2a for an example). In what follows, an $\Omega \to \mathbb{R}^2$ velocity field is called *piecewise-affine (PA)* if it is affine when re-

stricted to each triangle of $\mathcal{P}$. Let $\mathcal{V}$ denote the space of all *Continuous Piecewise-Affine (CPA)* $\Omega \to \mathbb{R}^2$ velocity fields that vanish on $\partial\Omega$, the boundary of Ω:

$$
\mathcal{V} \triangleq \{\boldsymbol{v}^{\boldsymbol{\theta}} : \boldsymbol{v}^{\boldsymbol{\theta}} \text{ is an } \Omega \to \mathbb{R}^2 \text{ CPA map (w.r.t. } \mathcal{P}) \\
\text{and } \boldsymbol{v}^{\boldsymbol{\theta}}(\boldsymbol{\xi}) = \boldsymbol{0}_{2\times1} \, \forall \boldsymbol{\xi} \in \partial\Omega \}. \tag{1}
$$

It can be shown [18] that $\mathcal{V}$ is a finite-dimensional linear space, where $d \triangleq \dim(\mathcal{V})$ is determined by how finely $\mathcal{P}$ is tessellated; thus, $\mathcal{V}$ is isomorphic to $\mathbb{R}^d$. In our experiments we use the tessellation shown in Fig. 2a. This choice, together with the boundary condition, implies that $d = 50$ (25 non-boundary vertices times 2 degrees of freedom in each one). In Eq. 1, the superscripted $\boldsymbol{\theta}$ is an element of $\mathbb{R}^d$, and we say that $\boldsymbol{\theta}$ parametrizes $\mathcal{V}$ in the sense that any element of $\mathcal{V}$ can be written as a linear combination of d orthonormal CPA fields with weights $\boldsymbol{\theta}$ [18].

A spatial transformation $T^{\boldsymbol{\theta}} : \Omega \to \Omega$ can then be derived by integrating a velocity field $\boldsymbol{v}^{\boldsymbol{\theta}} \in \mathcal{V}$ [18]; in other words, the space of the resulting transformations, denoted by M, is given by

$$
M \triangleq \{T^{\boldsymbol{\theta}} : \Omega \to \Omega , \, T^{\boldsymbol{\theta}} : \boldsymbol{\xi} \mapsto \phi^{\boldsymbol{\theta}}(\boldsymbol{\xi}, 1) \text{ where} \\
\phi^{\boldsymbol{\theta}}(\boldsymbol{\xi}, \cdot) \text{ solves Eq. 3 for some } \boldsymbol{v}^{\boldsymbol{\theta}} \in \mathcal{V}\}; \tag{2}
$$

$$
\phi^{\boldsymbol{\theta}}(\boldsymbol{\xi}, t) = \boldsymbol{\xi} + \int_0^t \boldsymbol{v}^{\boldsymbol{\theta}}(\phi^{\boldsymbol{\theta}}(\boldsymbol{\xi}, \tau)) \, d\tau , \\
\phi^{\boldsymbol{\theta}}(\cdot, \cdot) : \Omega \times \mathbb{R} \to \Omega . \tag{3}
$$

It can be shown [18] that elements of M are diffeomorphisms, that M contains the identity transformation $T^{\boldsymbol{0}} : \boldsymbol{\xi} \mapsto \boldsymbol{\xi}$, and that M is closed under inversion with $(T^{\boldsymbol{\theta}})^{-1} = T^{-\boldsymbol{\theta}}$. While these elements are obtained, via the solution of Eq. 3, from CPA velocity fields, they are generally *not* CPA themselves. Thus, elements of M are called CPA-*based* (CPAB) transformations [18].

It can further be shown that M is a nonlinear *d*-dimensional connected Riemannian manifold [18], and that $\mathcal{V}$ is a tangent space of M, where the identity transformation $T^{\boldsymbol{0}}$ is the point of tangency. To build statistical models over M we need mappings back and forth between the manifold and its tangent spaces [39].
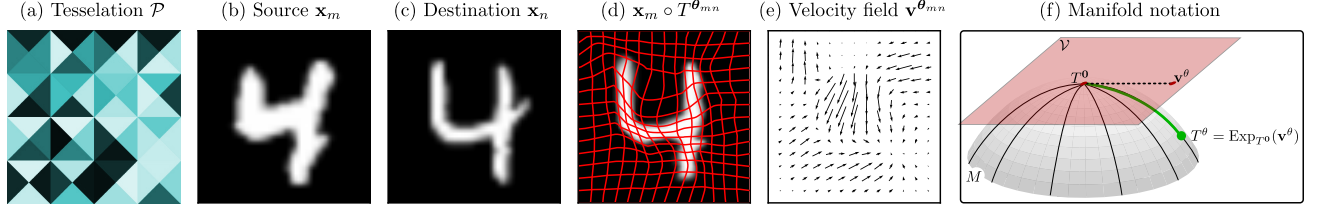
Figure 2: (a) An example tesselation. (b–c) Two images from the MNIST dataset. (d) The transformation between the two images. (e) The velocity field of the transformation. (f) Illustration of Riemannian concepts: $\mathcal{V}$ is the tangent space of the manifold M at the identity transformation; the exponential map take points from $\mathcal{V}$ onto M.

It can be shown that the restriction of the Lie group exponential map to $\mathcal{V}$ coincides with the mapping of $\boldsymbol{v}^{\boldsymbol{\theta}}$ to $T^{\boldsymbol{\theta}}$ via the integral equation (3). We thus write $T^{\boldsymbol{\theta}} = \mathrm{Exp}_{T^0}(\boldsymbol{v}^{\boldsymbol{\theta}})$, where $\mathrm{Exp}_{T^0} : \mathcal{V} \to M$ denotes the exponential mapping. Figure 2f illustrates the notation. It is worth noting that due to the special structure of CPA velocity fields, the exponential map has an efficient numerical implementation, which makes inference tractable [18]. Particularly, this enables a fast GPU implementation that parallelizes the evaluation of $(\boldsymbol{\xi}, \boldsymbol{\theta}) \mapsto T^{\boldsymbol{\theta}}(\boldsymbol{\xi})$ over multiple $\boldsymbol{\xi}$'s. We end this section with two remarks. First, as noted in [18], $\boldsymbol{\xi} \mapsto \phi^{\boldsymbol{\theta}}(\boldsymbol{\xi}, t = 1)$ should not be confused with (the non-diffeomorphism, parametric optical-flow-like representation) $\boldsymbol{\xi} \mapsto \boldsymbol{\xi} + \boldsymbol{v}^{\boldsymbol{\theta}}(\boldsymbol{\xi})$, the latter being only a *Taylor approximation* of the former. Second, while here we used the transformations from [18] due the reasons mentioned above, one may also consider other finite-dimensional spaces of diffeomorphisms [5, 50].

3.2 Estimating Transformations by Aligning Images

To estimate the transformations that naturally appear within a class, we align image pairs from said class. To avoid aligning all image pairs, we consider only the image pairs where one image is among the $K = 5$ nearest neighbors of the other. Let $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$ denote two such images. We then seek $T^{\boldsymbol{\theta}_{mn}} \in M$ such that $\boldsymbol{x}_m \circ T^{\boldsymbol{\theta}_{mn}} \approx \boldsymbol{x}_n$.

We adopt a Bayesian approach where our prior encodes smoothness. The posterior is given by

$$p(T^{\boldsymbol{\theta}}|\boldsymbol{x}_m, \boldsymbol{x}_n) \propto p(\boldsymbol{x}_m, \boldsymbol{x}_n|T^{\boldsymbol{\theta}}) \, p(T^{\boldsymbol{\theta}}). \qquad (4)$$

For the likelihood we take a simple i.i.d. Gaussian model of the intensity differences between the destination image and warped one:

$$\log p(\boldsymbol{x}_m, \boldsymbol{x}_n|T^{\boldsymbol{\theta}}) = -\frac{\|\boldsymbol{x}_m \circ T^{\boldsymbol{\theta}} - \boldsymbol{x}_n\|^2}{2\sigma^2} + \mathrm{const}, \quad (5)$$

where σ is user-specified. Other similarity measures, such as mutual information [47], can be used as well.

Following [18], we construct the prior over velocity fields $\boldsymbol{v}^{\boldsymbol{\theta}} \in \mathcal{V}$, which induces a prior over transformations via the exponential map, $T^{\boldsymbol{\theta}} = \mathrm{Exp}_{T^0}(\boldsymbol{v}^{\boldsymbol{\theta}})$. Let $\mathcal{V}_{\mathrm{PA}}$ denote the linear space of (possibly-discontinuous, and without boundary constraints) PA $\Omega \to \mathbb{R}^2$ velocity fields. The affine map associated with each triangle is given by a 2×3 matrix so $\dim(\mathcal{V}_{\mathrm{PA}}) \triangleq D = 6N_c$, where N_c is the number of triangles in $\mathcal{P}$. We use a standard *squared-exponential* (aka. *Gaussian*) kernel to define a zero-mean Gaussian distribution over $\mathcal{V}_{\mathrm{PA}}$ such that the correlations between the affine maps associated with each pair of triangles decay with the distance between triangle centroids [18]. Let $\boldsymbol{\Sigma}_{\mathrm{PA}}$ denote the covariance of that Gaussian. Note that $\mathcal{V}$ is a linear subspace of $\mathcal{V}_{\mathrm{PA}}$ with $d = \dim(\mathcal{V}) < D$. Let $\boldsymbol{B}$ be the $D \times d$ orthogonal basis matrix associated with the weights $\boldsymbol{\theta}$. The prior over $\boldsymbol{v}^{\boldsymbol{\theta}}$ is then defined as the projection of the prior over $\mathcal{V}_{\mathrm{PA}}$,

$$\boldsymbol{v}^{\boldsymbol{\theta}} \cong \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{B}^{\top}\boldsymbol{\Sigma}_{\mathrm{PA}}\boldsymbol{B}) \qquad (6)$$

(the symbol $\cong$ indicates that $\boldsymbol{v}^{\boldsymbol{\theta}}$ is isomorphic to $\boldsymbol{\theta}$).

We remark that other priors are easily implemented, e.g. a Gaussian Markov Random Field (GMRF) could provide an (improper) prior that prefers smoothness without penalizing large offsets.

For inference, we adopt a sampling approach using a Markov-Chain Monte-Carlo (MCMC) method [40]. Particularly, our implementation uses the time-honored Metropolis algorithm [35] with a (localized) Gaussian proposal. This is made tractable by the efficient numerical implementation of the exponential map [18] and the moderate dimensionality of $\mathcal{V}$. Using this sampling scheme, we then set $T^{\boldsymbol{\theta}_{mn}} = T^{\boldsymbol{\theta}} \sim p(T^{\boldsymbol{\theta}}|\boldsymbol{x}_m, \boldsymbol{x}_n)$.

Figure 3 shows examples of image alignments performed on the MNIST data set. It is evident that the estimated transformations are highly non-rigid, yet reasonably smooth. While the transformations generally provide a good fit, they are unable to capture topological differences. The rightmost column of Fig. 3 shows two such examples: one of the top 2-digits has

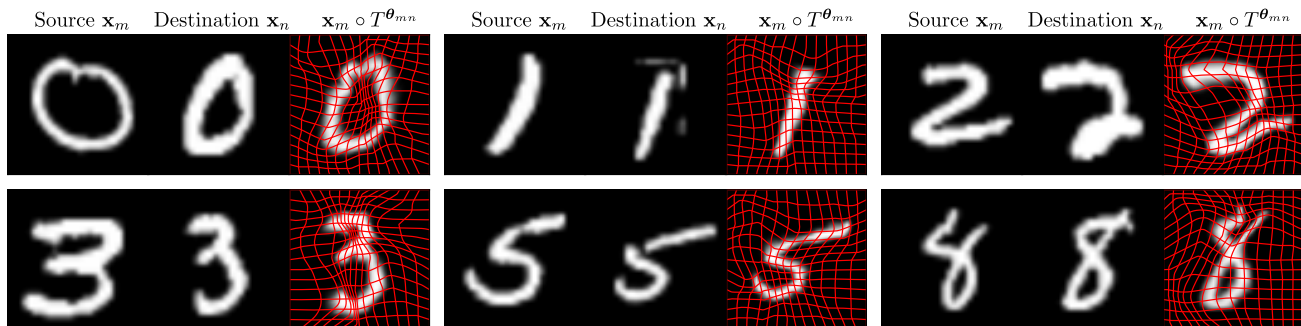| Source $\mathbf{x}_m$ | Destination $\mathbf{x}_n$ | $\mathbf{x}_m \circ T^{\boldsymbol{\theta}_{mn}}$ | Source $\mathbf{x}_m$ | Destination $\mathbf{x}_n$ | $\mathbf{x}_m \circ T^{\boldsymbol{\theta}_{mn}}$ | Source $\mathbf{x}_m$ | Destination $\mathbf{x}_n$ | $\mathbf{x}_m \circ T^{\boldsymbol{\theta}_{mn}}$ |

Figure 3: Example alignments.

a self-intersection, while the other does not; and one of the bottom 8-digits is open-ended at the top, while the other is not. In both cases, the transformation preserves the topology of the input image.

The entire alignment process takes approximately 10 seconds when implemented on GPU hardware.

3.3 Statistical Models of Transformations

Having aligned images pairs as discussed above, we now treat the inferred $\{T^{\boldsymbol{\theta}_{mn}}\}$ as observations. We aim to build a statistical model of the transformations found within a class, i.e. $p(T^{\boldsymbol{\theta}}|y)$. Let $T^{\boldsymbol{\theta}_{nm}}$ denote the inferred CPAB transformation from image $\boldsymbol{x}_n$ to $\boldsymbol{x}_m$, and let $\boldsymbol{\theta}_{nm} \cong \boldsymbol{v}^{\boldsymbol{\theta}_{nm}} = \mathrm{Log}_{T^0}(T^{\boldsymbol{\theta}_{nm}})$ be the representation of the transformation in the tangent space of the identity transformation; we may identify this tangent space with $\mathbb{R}^d$. Here $\mathrm{Log}_T(\cdot)$ is the inverse of $\mathrm{Exp}_T(\cdot)$.

Common strategies for describing distributions over manifold-valued data include kernel density estimators [36], and parametric models over tangent vectors [39]. We have found the latter approach to work well for modeling the estimated transformations. To minimize the distortion due to the linearization of the manifold, it is common to use the tangent space at the intrinsic mean, defined as

$$\mu = \arg\min_{\tilde{\mu}} \sum_{nm} \mathrm{dist}^2(T^{\boldsymbol{\theta}_{nm}}, \tilde{\mu}), \tag{7}$$

where $\mathrm{dist}^2(T^{\boldsymbol{\theta}_{nm}}, \tilde{\mu}) = \|\mathrm{Log}_{\tilde{\mu}}(T^{\boldsymbol{\theta}_{nm}})\|^2$ denotes the squared distance on the manifold. It is straightforward to show that, in our case, the identity transformation is an intrinsic mean:

Lemma 1 *The collection of pair-wise transformations* $\{T^{\boldsymbol{\theta}_{nm}}, T^{\boldsymbol{\theta}_{mn}}\}_{n,m} \forall n, m$ *has an intrinsic mean* (7) $\mu = T^0$.

Proof The gradient of Eq. 7 is $\sum_{nm} \mathrm{Log}_{\tilde{\mu}}(T^{\boldsymbol{\theta}_{nm}})$ [39]. Since $T^{\boldsymbol{\theta}_{nm}} = T^{-\boldsymbol{\theta}_{mn}}$, the gradient is zero at T^0, the identity transformation, implying that this is an intrinsic mean. □

With this in mind, we build a tangential parametric model at the identity. Empirically, we have found $\boldsymbol{v}^{\boldsymbol{\theta}}|y \sim \mathcal{N}(\mathbf{0}, \Sigma_y)$ to provide a good fit to the data. Here $\boldsymbol{\Sigma}_y = \frac{1}{N_y} \sum_{nm} (\boldsymbol{v}^{\boldsymbol{\theta}_{nm}})(\boldsymbol{v}^{\boldsymbol{\theta}_{nm}})^\top$ denotes the Riemannian covariance of the transformations [39], where N_y is the number of sample pairs used for this class. We then have

$$p(T^{\boldsymbol{\theta}}|y) \propto \exp\left(-\frac{1}{2}\mathrm{Log}_{T^0}(T^{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_y^{-1} \mathrm{Log}_{T^0}(T^{\boldsymbol{\theta}})\right) \tag{8}$$

$$= \exp\left(-\frac{1}{2}(\boldsymbol{v}^{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_y^{-1} \boldsymbol{v}^{\boldsymbol{\theta}}\right). \tag{9}$$

3.4 Sampling New Data

We assume that transformations and images are conditionally independent given the class label y. New transformations are generated as $T_i^{\boldsymbol{\theta}} = \mathrm{Exp}_{T^0}(\boldsymbol{v}_i^{\boldsymbol{\theta}})$, where $\boldsymbol{v}_i^{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{0}, \Sigma_y)$. This transformation is then applied to a uniformly-chosen image from the training set, yielding a new image. Note that by sampling not only the transformation but also the "template" image, the generated samples capture the fact that, due to topological changes, some of the inter-class deformations are not diffeomorphic.

Figure 1 shows randomly-chosen images from the MNIST dataset and randomly-deformed instances of the same images. It is evident that the randomly-sampled transformations are non-rigid, yet produce realistically-looking images. A more systematic illustration is given in Fig. 4. The shown images are ± 3 times the standard deviation along the first principal component of the transformations found in each class. It is evident that $p(T^{\boldsymbol{\theta}}|y)$ captures non-rigid deformations in the data that would be difficult to model manually, e.g. the first principal deformation of "2" captures whether the stroke of the tail bends upwards or downwards, while the first principal deformation

of "8" captures the position of the self-intersection. The supplementary material contains similar plots for higher-order components as well as an animation of these transformations. This animation provides the best visualization of the principal components.

4   Experiments

We evaluate the learned augmentation scheme on the *MNIST* data set. This allows us to compare with *Infinite MNIST (InfiMNIST)* [32], which is currently the most extensive augmentation scheme available in the literature. The original *MNIST* dataset consists of 60,000 training images and 10,000 test images. We hold out 10,000 of the training images to form a validation set. The augmentations are, thus, based on only 50,000 training images.

We form our augmented datasets as follows. For each of the 10 classes we sample 500 images, and form the $K = 5$ undirected nearest neighbor graph. For each edge we compute the transformation $T^{\boldsymbol{\theta}_{nm}}$ between the corresponding image pair. This gives, on average, 2940 transformations per class. We fit a zero-mean multivariate Gaussian to the tangential representations $\boldsymbol{v}^{\boldsymbol{\theta}_{nm}}$ of the transformations, and generate new data as described in Sec. 3.4. With this approach we generate two new training sets: The *AlignMNIST* set is generated by uniformly sampling images from the 50,000-element training set and applying transformations sampled from $p(T^{\boldsymbol{\theta}}|y)$ to generate 1,000,000 images per class. The *AlignMNIST500* set is generated similarly except we only sample images from the set of 500 images from which transformations were estimated; again we sample 1,000,000 images per class. The *AlignMNIST500* set is, thus, generated from only 500 images per class and allows us to experiment with the effect of learned augmentation in small datasets. For comparative purposes, we also sample 1,000,000 images per class from *InfiMNIST* based on the 50,000-element training set; we further generate *InfiMNIST500* by sampling from *InfiMNIST* using only the same 500 images per class as for *AlignMNIST500*.

As an initial experiment we consider a simple nearest-neighbor classifier on *MNIST* (test error: 3.1%), *InfiMNIST* (test error: 2.6%), and *AlignMNIST* (test error: 1.4%). This gives a hint that the learned augmentation scheme captures invariances that were missed in the laborious manual specification behind *InfiMNIST*.

Next, for each dataset we train a *multilayer perceptron (MLP)* with hyperparameters estimated with cross-validation on the held-out validation set. The best set of hyperparameters is then used to train a network on the entire training set. For both *InfiMNIST*
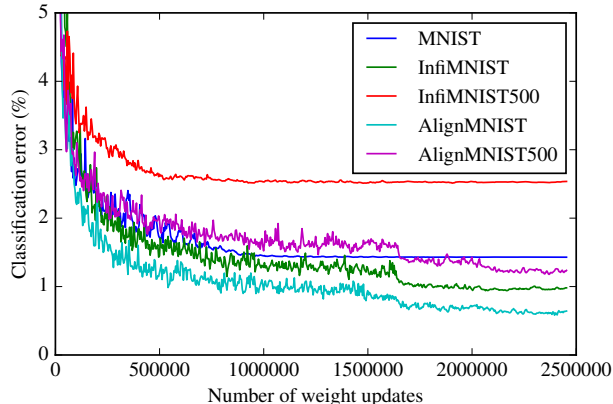

Figure 5: Learning curves for different training sets. The classification error is calculated on the MNIST test set.

and *AlignMNIST*, we experience no overfitting problems due to the variety of the input samples. In fact, for both augmentation schemes, the network training converges even before the network has seen the entire dataset. Therefore, we train the final model without a scheme for early stopping by simply doubling the amount of weight updates used for validation. For both augmentation schemes, we achieve best performance using networks consisting of 3 hidden layers with 2048 units each. To speed up the training, we optimize using a stochastic gradient descent with momentum and employ rectified linear units as activation functions.

In addition to the MLP we train a convolutional network (ConvNet) in a similar fashion on the datasets. The ConvNet consists of 2 convolutional + pooling layers followed by a fully-connected hidden layer of 512 units. To avoid overfitting, we rely on early stopping as determined on the validation set.

Figure 5 shows the learning curves of the MLP on the different datasets, while Fig. 6 summarizes the results. It is evident that *AlignMNIST* gives rise to the best predictive model, with a significant improvement over *InfiMNIST*. This is clearer when we consider datasets generated from only 500 data points per class. The performance of the MLP trained on *AlignMNIST500* is less than 0.2 percent-points worse than the MLP trained on the entire *InfiMNIST*, and 1.6 percent-points better than the model trained on *InfiMNIST500*. The additional performance gains attained by the ConvNets confirm that the benefits of our augmentation scheme generalize to networks with spatial structure.

Note that we are unable to reproduce the ConvNet results on *InfiMNIST* previously reported by [43].
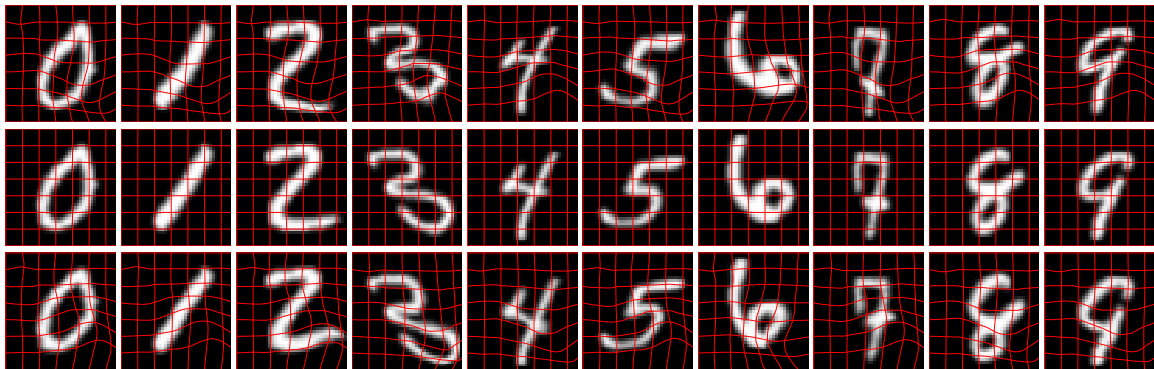
Figure 4: The first principal component of the transformations of each class. *Center row:* the mean (identity) transform. *Top row:* the mean plus 3 standard devidations. *Bottom row:* the mean minus 3 standard deviations. **See also supplementary animation and plots of further components.**

Dataset	MLP Test error	ConvNet Test error
MNIST	$1.42\% \pm 0.055$	$0.65\% \pm 0.08\%$
InfiMNIST	$0.89\% \pm 0.079$	$0.49\% \pm 0.04\%$
AlignMNIST	$0.58\% \pm 0.022$	$0.44\% \pm 0.02\%$
InfiMNIST500	$2.59\% \pm 0.050$	$1.04\% \pm 0.07\%$
AlignMNIST500	$1.06\% \pm 0.047$	$0.84\% \pm 0.05\%$

Figure 6: Final test error on the different datasets.

Nonetheless, we believe that the ConvNet numbers are representative as the same implementation and evaluation scheme are used for all datasets.

The reported results for *InfiMNIST500* and *AlignM-NIST500* should be compared with the current state-of-the-art for MNIST classification based on small training sets. Amit et al. [2] report an error rate of 1.5% when trained on 500 images per class. Both our MLP and our ConvNet significantly improve upon this, which demonstrate that a good augmentation scheme can make deep learning a viable option even for fairly small datasets.

5 Discussion

Our basic idea is simple: *instead of manually specifying data augmentation schemes, we build a statistical model of the transformations found within a given class, and use this to augment the dataset.* The practical implementation of this idea, however, requires some care. A naive approach would build a statistical model over dense displacement fields. This would, however, be a very high-dimensional model that would allow for too much flexibility. Such an approach would also require aligning substantially more image pairs, which would increase computational demands. Constraining the transformations to lie in a finite-dimensional space

of diffeomorphisms not only drastically lowers dimensionality but also ensures that samples from the probabilistic model are well-behaved transformations. As the first step of the proposed method involves solving multiple problems of inference over latent diffeomorphisms, we crucially depend on the availability of an efficient-yet-highly-expressive representation of diffeomorphisms; fortunately, this is an active area of research with several recent successes [5, 18, 50].

We find that a learned augmentation scheme allows for significantly smaller training sets. In particular, we report state-of-the-art results on small subsamples of MNIST. This observation potentially allows very large models to be trained on fairly small datasets; we observe that the learned augmentation scheme using a fraction of the training data not only outperforms a classifier trained on the entire dataset but also does significantly better than manually-specified augmentations. As there are $\mathcal{O}(N_y^2)$ potential pairs to be aligned in a class of N_y observations, it is generally possible to get access to enough transformations to learn a good augmentation scheme. A further benefit of learned augmentation schemes is that different schemes may be applied within different classes — this is generally impractical to do in manually-designed schemes.

A limitation of our approach is that we must be able to align observations in order to build statistical models of the deformations found within the dataset. We exemplify our idea on the MNIST data where observations often have well-defined alignments. Our approach is, however, not limited to MNIST:

- Image alignment is a routine task in many medical imaging tasks, such as the analysis of *magnetic resonance images (MRI)* [7, 13, 37], *X-ray Computed Tomography (CT)* [34, 41], *Positron Emission Tomography (PET)* [45, 46] and *mam-*

mograms [24, 38]. Our work directly extends to these domains.

- We make similar observations for time-series data such as acoustic signals [12, 23]. Here *dynamic time warping (DTW)* [8] is often used as pre-processing to remove differences in the temporal speed of individual signals. The CPAB representation readily provides a replacement for DTW and our approach can be used to augment the datasets. It is worth noting that DTW itself is poorly-suited for such a task as a generative statistical model of deformations learned from estimated DTWs will not yield new invertible transformations even if all estimated DTWs happen to be invertible. We avoid such issues by relying on the CPAB representation, which ensures diffeomorphic deformations.

- *Mesh alignment* is also standard pre-processing step in the analysis of three-dimensional meshes [1, 3, 9, 25]. As deep models are beginning to appear for three-dimensional data [49] it would be interesting to combine them with learned augmentation schemes.

In summary, the main contribution of this paper is the first approach for learning data augmentation schemes. Our results show that it is beneficial to learn statistical models of transformations over manually specifying them. We do not find it surprising that *learning* beats *hand-crafting*.

Acknowledgements

References

[1] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 612–619. ACM, 2002.

[2] Y. Amit and A. Trouvé. Pop: Patchwork of parts models for object recognition. *International Journal of Computer Vision (IJCV)*, 75(2):267–282, Nov. 2007.

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[4] V. Arnold. *Mathematical methods of classical mechanics.* Springer Science & Business Media, 1989.

[5] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A log-euclidean polyaffine framework for locally rigid or affine registration. In *BIR*. Springer, 2006.

[6] H. S. Baird. Document image defect models. In *SDIA*, pages 546–556. Springer, 1992.

[7] M. Beg, M. Miller, A. Trouv, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, 2005.

[8] R. Bellman and R. Kalaba. On adaptive control processes. *Automatic Control, IRE Transactions on*, 4(2):1–9, Nov 1959.

[9] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3794–3801. IEEE, 2014.

[10] O. Chapelle et al. Semi-supervised learning. 2006.

[11] O. Chapelle and B. Schölkopf. Incorporating invariances in non-linear support vector machines. In *NIPS*, pages 609–616, 2001.

[12] X. Cui, V. Goel, and B. Kingsbury. Data augmentation for deep neural network acoustic modeling. In *ICASSP*, pages 5582–5586, May 2014.

[13] S. Darkner and J. Sporring. Locally orderless registration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1437–1450, June 2013.

[14] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Unsupervised feature learning by augmenting single images. *arXiv preprint arXiv:1312.5242*, 2013.

[15] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1792–1799. IEEE, 2011.

[16] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014.

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[18] O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. F. III. Highly-expressive spaces of well-behaved transformations: Keeping it simple. In *International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015.

[19] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *NIPS*, pages 646–654, 2009.

[20] T. Graepel and R. Herbrich. Invariant pattern recognition by semi-definite programming machines. In S. Thrun, L. Saul, and B. Schölkopf, editors, *NIPS*, pages 33–40. MIT Press, 2004.

[21] U. Grenander. *General pattern theory: A mathematical study of regular structures.* Clarendon Press, 1993.

[22] B. Hariharan, C. L. Zitnick, and P. Dollár. Detecting objects using deformation dictionaries. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1995–2002. IEEE, 2014.

[23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[24] J. H. Hipwell, C. Tanner, W. R. Crum, J. Schnabel, D. J. Hawkes, et al. A new validation method for x-ray mammogram registration algorithms using a projection model of breast x-ray compression. *Medical Imaging, IEEE Transactions on*, 26(9):1190–1200, 2007.

[25] D. A. Hirshberg, M. Loper, E. Rachlin, and M. J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Computer Vision–ECCV 2012*, pages 242–255. Springer, 2012.

[26] B. Hoyle, M. M. Rau, C. Bonnett, S. Seitz, and J. Weller. Data augmentation for machine learning redshifts applied to SDSS galaxies. *Monthly Notices of the Royal Astronomical Society*, 450:305, 2015.

[27] N. Jaitly and G. E. Hinton. Vocal tract length perturbation (VTLP) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013.

[28] A. Kanazawa, A. Sharma, and D. Jacobs. Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*, 2014.

[29] J. J. Kivinen and C. K. Williams. Transformation equivariant boltzmann machines. In *ICANN*, pages 1–9. Springer, 2011.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1097–1105. Curran Associates, Inc., 2012.

[31] Q. Liao, J. Z. Leibo, and T. Poggio. Learning invariant representations and applications to face verification. In *NIPS*, 2013.

[32] G. Loosli, S. Canu, and L. Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, pages 301–320, 2007.

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

[34] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank. Pet-ct image registration in the chest using free-form deformations. *Medical Imaging, IEEE Transactions on*, 22(1):120–128, 2003.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[36] E. G. Miller and C. Chefd'Hotel. Practical non-parametric density estimation on a transformation group for vision. In *CVPR*, volume 2, pages II–114, 2003.

[37] M. I. Miller. Computational anatomy: shape, growth, and atrophy comparison via diffeomorphisms. *NeuroImage*, 23:S19–S33, 2004.

[38] M. Nielsen, P. Johansen, A. Jackson, B. Lautrup, and S. Hauberg. Brownian warps for non-rigid registration. *Journal of Mathematical Imaging and Vision*, 31:221–231, 2008.

[39] X. Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

[40] C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Citeseer, 2004.

[41] D. Rueckert and J. Schnabel. Medical image registration. In T. M. Deserno, editor, *Biomedical Image Processing*, Biological and Medical Physics, Biomedical Engineering, pages 131–154. Springer Berlin Heidelberg, 2011.

[42] P. Simard, B. Victorri, Y. LeCun, and J. S. Denker. Tangent prop – a formalism for specifying selected invariances in an adaptive network. In *NIPS*, pages 895–903, 1992.

[43] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *2013 12th International Conference on Document Analysis and Recognition*, volume 2, pages 958–958. IEEE Computer Society, 2003.

[44] K. Sohn and H. Lee. Learning invariant representations with local transformations. In *ICML*, pages 1311–1318, 2012.

[45] C. Studholme, D. Hill, and D. Hawkes. Automated 3d registration of mr and pet brain images by multi-resolution optimisation of voxel similarity measures. *Med. Phys*, 24(1):25–35, 1997.

[46] C. Studholme, D. L. Hill, and D. J. Hawkes. Multiresolution voxel similarity measures for mr-pet registration. In *Information processing in medical imaging*, volume 3, pages 287–298. Dordrecht, The Netherlands: Kluwer, 1995.

[47] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1997.

[48] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 756–763. IEEE, 2005.

[49] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[50] M. Zhang and P. T. Fletcher. Finite-dimensional Lie algebras for fast diffeomorphic image registration. In *IPMI*, 2015.