Auto-Encoding Transformations in Reparameterized Lie Groups for Unsupervised Learning

Feng Lin,[1] **Haohang Xu,**[2] **Houqiang Li,**[1,4] **Hongkai Xiong,**[2] **Guo-Jun Qi**[3,*]

[1] CAS Key Laboratory of GIPAS, EEIS Department, University of Science and Technology of China
[2] Department of Electronic Engineering, Shanghai Jiao Tong University
[3] Laboratory for MAPLE, Futurewei Technologies
[4] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

Abstract

Unsupervised training of deep representations has demonstrated remarkable potentials in mitigating the prohibitive expenses on annotating labeled data recently. Among them is predicting transformations as a pretext task to self-train representations, which has shown great potentials for unsupervised learning. However, existing approaches in this category learn representations by either treating a discrete set of transformations as separate classes, or using the Euclidean distance as the metric to minimize the errors between transformations. None of them has been dedicated to revealing the vital role of the geometry of transformation groups in learning representations. Indeed, an image must continuously transform along the curved manifold of a transformation group rather than through a straight line in the forbidden ambient Euclidean space. This suggests the use of geodesic distance to minimize the errors between the estimated and groundtruth transformations. Particularly, we focus on homographies, a general group of planar transformations containing the Euclidean, similarity and affine transformations as its special cases. To avoid an explicit computing of intractable Riemannian logarithm, we project homographies onto an alternative group of rotation transformations $\mathbf{SR}(3)$ with a tractable form of geodesic distance. Experiments demonstrate the proposed approach to Auto-Encoding Transformations exhibits superior performances on a variety of recognition problems.

Introduction

Representation learning plays a vital role in many machine learning problems. Among them is supervised pretraining from large-scale labeled datasets such as ImageNet and MS COCO for various recognition tasks ranging from image classification, object detection, and segmentation. Previous results have shown that representations learned through supervised training on labeled data is critical to competitive performances before being finetuned on the downstream tasks. However, it becomes prohibitively expensive to collect labeled data for the purpose of pre-training representation networks, which prevents the application of deep networks in many practical applications. This inspires many works to explore the possibility of pretraining representations in an unsupervised fashion by removing the reliance on labeled data.

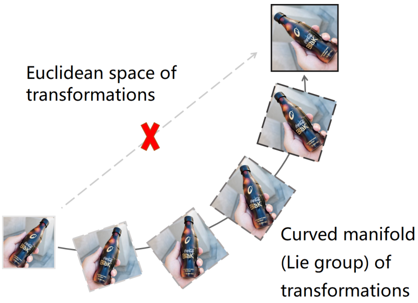

Euclidean space of transformations
Curved manifold (Lie group) of transformations

Figure 1: The deviation between two transformations should be measured along the curved manifold (Lie group) of transformations rather than through the forbidden Euclidean space of transformations.

Specifically, self-supervised representation learning from transformed images has shown great potential. The idea can trace back to the data augmentation in supervised learning of convolutional neural networks in AlexNet, where labeled data are augmented by various copies of their transformed copies. The fine structure of transformations have also been explored in training deep network representations. For example, the celebrated convolutional neural networks are trained to transform equivariantly against the translations to capture the translated visual structures. This idea has been extended to train deep convolutional networks that are equivariant to to a family of transformations beyond translations. However, these models still rely on supervised pretraining of representations on labeled data.

Recently, unsupervised pretraining of representations was proposed in a self-supervised fashion. The deep networks are trained by decoding the pretext signals without labeled data (Kolesnikov, Zhai, and Beyer 2019). The state-of-the-art performances have been achieved by the representations learned through various transformations as the self-supervisory signals. Among them is a representative paradigm of Auto-Encoding Transformations (AET) (Qi 2019) that extends the conventional autoencoders from reconstructing data to transformations. The AET seeks to ex-

* corresponding author

† The work of Houqiang Li was supported by NSFC under Contract 61836011 and 62021001.

plore the variations of data under various transformations, and capture the intrinsic patterns of data variations by pretraining representations without need to annotate data.

It relies on a simple criterion: if a representation network could successfully model the change of visual structures before and after a transformation, we should be able to decode the transformation by comparing the representations of original and transformed images. In other words, transformations are used as a proxy to reveal the pattern of visual structures that are supposed to equivary against the applied transformations. Both deterministic and probabilistic AET models have been presented in literature, which minimizes the Mean-Square Errors (MSE) between the matrix representation of parametric transformations (Zhang et al. 2019) and maximizes the mutual information between the learned representation and transformations (Qi 2019), respectively.

Albeit the promising performances on both AET models, they are subject to a questionable assumption — they use the Euclidean distance as the metric to minimize the errors between the estimated and the groundtruth transformations. Obviously, this assumption is problematic. Indeed, transformations have a Lie group structure restricted on a curved manifold of matrix representations rather than being filled in the ambient Euclidean space. Moreover, a group of transformations can be decomposed into various components of atomic transformations. For example, a homography as one of the most general planar transformations can be decomposed into affine, similarity and Euclidean transformations hierarchically. A direct summation of the Euclidean distances over these components could yield incomparable results with some components overweighing the others due to various scales.

From the perspective of transformation geometry in Lie groups, a more natural way to measure the different between transformations is the shortest curved path (*i.e.*, geodesic) connecting two transformations along the manifold, instead of a straight segment in the ambient Euclidean space. As illustrated in Figure 1, given a pair of original and transformed images, one cannot find a valid path of transformations in the Euclidean space to continuously transform the original one to its transformed counterpart. On the contrary, such a path of transformations can only reside on the curved manifold of a transformation group. Therefore, one should use the geodesic along the manifold as the metric to measure the errors between transformations.

To this end, we will leverage the theory of Lie groups to explore the geometry of transformations, and choose to minimize the revealed geodesic errors between transformations on the underlying manifold to pretrain the unsupervised representations. Specifically, we will consider the Lie group $\mathbf{PG}(2)$ of homographies as to train the AET model, as it contains a rich family of planar transformations that generalize various transformations including as its special cases the Euclidean transformations such as translations and rotations, as well as isotropic-scaling similarity transformations and anisotropic-scaling affine transformations.

There exist two challenges to implement the AET for the group of homographies. First, one needs the Riemannian logarithm to express the geodesic between transformations, which can be intractable for the homography group with no closed form. Instead, we project the transformations onto another group with a tractable expression of geodesic distances. Particularly, we choose the scaled rotation group $\mathbf{S}^+(1) \times \mathbf{SO}(3)$ to approximate $\mathbf{PG}(2)$, in which the geodesic distance has a tractable closed form that can be efficiently computed without taking an explicit group logarithm. This can greatly reduce the computing overhead in self-training the proposed model in the Lie group of homographies. Second, to train the AET, we need a reasonable parameterization of homographies that can be computed from the deep network. Although a 3×3 matrix representation in homogeneous coordinates is available to parameterize homographies directly, it mixes up various components of transformations such as rotations, translations, and scales that have incomparable scales of values. The direct matrix parameterization is thus not optimal which may result in unstable numeric results. Thus, inspired by deep homography estimation (DeTone, Malisiewicz, and Rabinovich 2016; Erlik Nowruzi, Laganiere, and Japkowicz 2017; Nguyen et al. 2018; Le et al. 2020), we choose to a reparameterization approach by making the network output an implicit expression of homographies with the four-point correspondences between original and transformed images. The matrix representation is computed from the output correspondences through a SVD operation. Such reparameterization of homographies is differentiable, thereby enabling us to minimize the geodesic errors between transformations to train the deep network in an end-to-end fashion.

In summary, our major contributions are as follows:

- We leverage the theory of Lie groups to explore the geometry of transformations, using geodesic distance to minimize the errors between the estimated and groundtruth transformations in AET.
- Due to the intractable Riemannian logarithm, we propose an approximation of $\mathbf{PG}(2)$ in a closed form by projecting the transformations onto the scaled rotation group.
- Experiments demonstrate the proposed method exhibits superior performances on various recognition problems.

Background & Methodology

AET with Euclidean Distances

First, let us review the Auto-Encoding Transformations (AET) (Zhang et al. 2019). In the AET, a transformation $\mathbf{t}$ is sampled from a Lie group $\mathcal{G}$, which is then applied to an image $\mathbf{x}$, resulting in a transformed copy $\mathbf{t}(\mathbf{x})$. Usually, an image transformation $\mathbf{t}$ can be represented by the corresponding 3×3 matrix $\mathbf{T}$ in a 3D homogeneous coordinate. Such a Lie group $\mathcal{G}$ of transformations we will consider in this paper is both a group equipped with a composition between transformation matrices, and a manifold of matrices endowed with a Riemannian metric. More preliminaries about the Lie group can be found in Appendix 1.

The goal of the AET is to learn a representation encoder $E_\phi(\mathbf{x})$ for each image $\mathbf{x}$, as well as a transformation decoder D_ψ to estimate the transformation $\mathbf{t}$ from the representations $E_\phi(\mathbf{x})$ and $E_\phi(\mathbf{t}(\mathbf{x}))$ of original and transformed images, where ϕ and ψ are model weights of the encoder and decoder. It is supposed that a good representation E_ϕ ought to capture the intrinsic visual structures of individual images, so that the decoder can infer the applied transformation by comparing the encoded image representations before and after the transformations. Recent work has also revealed the relation between the AET model and the transformation-equivariant representations from an information-theoretic point of view (Qi et al. 2019).

Formally, to learn the encoder E_ϕ and the decoder D_ψ, one can choose to minimize the following Mean-Squared Error (MSE) over weights ϕ and ψ to train the AET model

$$\min_{\phi,\psi} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{t}} \frac{1}{2}\|\hat{T}_{\phi,\psi} - T\|_{\mathrm{F}}^2 \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of matrix, $\hat{\mathbf{T}}$ is an estimate of the matrix representation $\mathbf{T}$ of the applied transformation, which is a function of the model parameters (ϕ, ψ), and the mean-squared error is taken over the sampled images $\mathbf{x}$ and transformations $\mathbf{t}$. For the notational simplicity, we will drop the subscript (ϕ, ψ) in $\hat{\mathbf{T}}_{\phi,\psi}$ whenever it is clear from the context. The model that minimizes the MSE between transformation matrices is named the AETv1 in this paper. Although the results (Zhang et al. 2019) showed its impressive performances, the MSE objective may not exactly characterize the intrinsic distance between a sampled and an estimated transformations, as it simply uses the Euclidean distances between them.

Indeed, a Lie group $\mathcal{G}$ of transformations is embedded into the ambient matrix space, often forming a curved manifold. Obviously, a more accurate distance between transformations should be characterized by the length of the geodesic connecting them along the manifold, which characterizes how a transformation can continuously change to another transformation along the manifold. Minimizing such a geodesic distance on the manifold can yield a more exact estimate of a sampled transformation along the Lie group of transformations.

AET in Lie Groups

In the AET, we seek to train the model by minimizing the mean-squared error between the estimated and sampled transformations $\hat{\mathbf{T}}$ and $\mathbf{T}$. However, a Lie group $\mathcal{G}$ of transformations has a curved manifold structure embedded in an ambient matrix space. The mean-squared error characterizes the Euclidean distance between two transformation matrices, which reflects their external distance in the ambient Euclidean space. It cannot reflect the intrinsic distance between transformations along the curved manifold of the Lie group.

Instead, in this paper, we will consider the geodesic that is the short curve connecting two transformations along the curved manifold between them. It can be shown that the geodesic between $\hat{\mathbf{T}}$ and $\mathbf{T}$ can be represented as

$$\gamma(\mathrm{t}) = \mathrm{Exp}_{\mathbf{T}}(\mathrm{t}\,\mathrm{Log}_{\mathbf{T}}(\hat{\mathbf{T}})) = \mathbf{T}\mathrm{Exp}_{\mathbf{I}}(\mathrm{t}\,\mathrm{Log}_{\mathbf{I}}(\mathbf{T}^{-1}\hat{\mathbf{T}})) \tag{2}$$

where $\gamma(0) = \mathbf{T}$ and $\gamma(1) = \hat{\mathbf{T}}$, and $\mathrm{Exp}_{\mathbf{T}}$ and $\mathrm{Log}_{\mathbf{T}}$ are the Riemannian exponential and logarithm that map from the tangent space at a transformation $\mathbf{T}$ to the manifold of a Lie group and conversely. The readers who are interested in more preliminaries about the Lie group can refer to (Zacur, Bossa, and Olmos 2014a) and Appendix.

The geodesic distance between $\mathbf{T}$ and $\hat{\mathbf{T}}$ is defined as

$$\int_0^1 \|\gamma'(\mathrm{t})\|\mathrm{dt} = \|\gamma'(0)\|$$

where $\|\cdot\|$ is the left-invariant metric. This equality follows from that $\gamma'(\mathrm{t})$ is parallel along $\gamma(\mathrm{t})$ by the definition of geodesics, and thus $\|\gamma'(\mathrm{t})\|$ is a constant of t along the geodesic.

The derivative of the geodesic $\gamma(\mathrm{t})$ at $\mathrm{t} = 0$ is

$$\gamma'(0) = \mathrm{Log}_{\mathbf{T}}(\hat{\mathbf{T}}) = \mathbf{T}\,\mathrm{Log}_{\mathbf{I}}(\mathbf{T}^{-1}\hat{\mathbf{T}}) \in \mathcal{T}_{\mathbf{T}}\mathcal{G}$$

which lies in the tangent space $\mathcal{T}_T\mathcal{G}$ of the transformation group $\mathcal{G}$ at $\mathbf{T}$. Thus, the (squared) Geodesic Distance between Transformations (GDT) can be rewritten as

$$\ell(\hat{\mathbf{T}}, \mathbf{T}) \triangleq \frac{1}{2}\|\gamma'(0)\|^2 = \frac{1}{2}\|\mathrm{Log}_{\mathbf{I}}(\mathbf{T}^{-1}\hat{\mathbf{T}})\|_{\mathrm{F}}^2 \tag{3}$$

where we use the left-invariant property of $\|\cdot\|$, and left-translate $\gamma'(0)$ by $\mathbf{T}$ to the tangent space at the identity. Then, one can minimize the GDT between the sampled and estimated transformations to train the AET.

Let us denote $\mathrm{Log}_{\mathbf{I}}(\mathbf{T}^{-1}\hat{\mathbf{T}})$ by $\mathbf{R}$. Following the chain rule, we show in Appendix 2 that the derivative of the loss ℓ *w.r.t.* the decoded transformation $\hat{\mathbf{T}}$

$$\nabla_{\hat{\mathbf{T}}}\ell(\hat{\mathbf{T}}, \mathbf{T}) = \overline{\mathbf{R}}^{\intercal}\left(\nabla_{\mathbf{R}}\mathrm{Exp}_{\mathbf{I}}(\mathbf{R})\right)^{-1}\mathbf{I} \otimes \mathbf{T}^{-1} \tag{4}$$

where the overline $\overline{\mathbf{R}}$ denotes the stacking of the matrix columns, and $\otimes$ is the Kronecker product between matrices. Then, the training errors can be back-propagated through this derivative to update the network weights iteratively. However, it is usually intractable to directly minimize the Riemannian logarithm $\mathbf{R}$ to train the AET. We will show a concrete example to implement AET in a general Lie group with a alternative projection approach in the next section.

AET with Homographies

In this section, we will discuss in details how to implement the AET in the homograhy group, a Lie group that covers the entire family of projective transformations. First, we will review the Lie group of homographies in the next subsection and a direct computing of geodesic distances in it.

Homography Transformations The AETv1 have shown the group $\mathbf{PG}(2)$ of homographies (aka projective transformations) have gained impressive performances with the AET model (Zhang et al. 2019). It contains a rich family of spatial transformations that can reveal the visual structures of images. Thus, we discuss the details about how to implement the AET with $\mathbf{PG}(2)$ considering its Lie group structure, namely AETv2 in this paper.

The 2D homography transformations $\mathbf{PG}(2)$ can be defined as the transformations in the augmented 3D homogeneous space of image coordinates (Beutelspacher, Albrecht, and Rosenbaum 1998). Its matrix representation contains all matrices with a unit determinant, and the corresponding Lie algebra (*i.e.*, the tangent space at the identity) $\mathfrak{g}$ consists of all matrices with zero trace. With a left-invariant metric, the Riemannian exponential can be written in terms of matrix exponential (Zacur, Bossa, and Olmos 2014a),

$$\mathrm{Exp}_{\mathbf{I}}(\mathbf{R}) = \exp(\mathbf{R}^{\intercal})\exp(\mathbf{R} - \mathbf{R}^{\intercal}) \tag{5}$$

for all $\mathbf{R} \in \mathfrak{g}$. Unfortunately, its inverse, the Riemannian logarithm, has no closed form that can be expressed in terms of matrix logarithm. This makes it hard to directly minimize the geodesic distance between homographies, we will discuss an alternative approximation method to this problem.

Two Challenges In this section, we will address two challenges of implementing the AET in the homography group. First, directly solving the derivative in Eq. (4) needs to calculate the Riemannian logarithm $\mathbf{R}$ of $\mathbf{T^{-1}\hat{T}}$. However, the lack of the closed-form solution to this Riemannian logarithm requires an iterative algorithm to solve it, which is computationally prohibitive and hard to implement. Instead, we will choose to project the homographies into a subgroup with tractable geodesic distances in Subsection *A*. Second, the output of the transformation decoder needs to be suitably parameterized to estimate the homography matrix. A direct parameterization of homography matrix is problematic as it contains differents components of transformations that cannot be estimated in a balanced fashion, such as rotations, translations and shears. Thus, we will discuss an alterative parameterization in Subsection *B*.

A. **Projection onto $\mathbf{SR}(3)$ Subgroup**: here we present an alternative method by projecting the target transformation $\mathbf{T^{-1}\hat{T}}$ onto a subgroup of transformations, where there exists a tractable Frobenius norm of Riemannian logarithm (Zacur, Bossa, and Olmos 2014b). Let us consider a scaled rotation group $\mathbf{SR}(3) \triangleq \mathbf{S}^{+}(1) \times \mathbf{SO}(3)$, which is the direct product between the group $\mathbf{S}^{+}(1)$ of positive isotropic scalings and the 3D rotation group $\mathbf{SO}(3)$. Its matrix representation consists of 3×3 matrices $s\mathbf{T}$ such that $s \in \mathbb{R}^{+}$, and $\mathbf{T} \in \mathbf{SO}(3)$ subject to $\mathbf{T}^{\intercal}\mathbf{T} = \mathbf{I}$ and $\det(\mathbf{T}) = 1$. The corresponding Lie algebra has a form of $\epsilon\mathbf{I} + \mathbf{L}$ for all skew-symmetric matrices (*i.e.*, $\mathbf{L}^{\intercal} = -\mathbf{L}$) with traces equal to zero (Zacur, Bossa, and Olmos 2014a). The Riemannian logarithm for such a scaled rotation group is the matrix logarithm since the matrices commute in its Lie algebra.

Formally, one can estimate $\mathbf{R} \triangleq \mathrm{Log}_{\mathbf{I}}(\mathbf{T^{-1}\hat{T}})$ in (3) by projecting it onto $\mathbf{SR}(3)$, where the matrix logarithm can be used in place of the Riemannian logarithm $\mathrm{Log}_{\mathbf{I}}$,

$$\hat{\mathbf{R}} = \log\left[\underset{\mathbf{S}^{+}(1)\times\mathbf{SO}(3)}{\Pi}(\mathbf{T^{-1}\hat{T}})\right]$$

where $\log$ is the conventional matrix logarithm, and $\Pi(\cdot)$ is the projection onto $\mathbf{S}^{+}(1) \times \mathbf{SO}(3)$. With the singular value decomposition of $\mathbf{T^{-1}\hat{T}} = \mathbf{U\Sigma V}^{\intercal}$, we prove in Appendix 3 that the projection onto $\mathbf{SR}(3)$ can be written in a closed form as

$$\underset{\mathbf{S}^{+}(1)\times\mathbf{SO}(3)}{\Pi}(\mathbf{T^{-1}\hat{T}}) = \alpha\mathbf{P} \tag{6}$$

with $\alpha = \dfrac{1}{3}\mathrm{tr}(\mathbf{\Sigma D})$, $\mathbf{P} \triangleq \mathbf{UDV}^{\intercal} \in \mathbf{SO}(3)$, where $\mathbf{D} \triangleq \mathrm{diag}\{1, 1, \det(\mathbf{UV}^{\intercal})\}$.

Then, the Frobenius norm of $\hat{\mathbf{R}}$ can be written as

$$\begin{aligned}\|\hat{\mathbf{R}}\|_{\mathrm{F}}^2 &\triangleq \|\log\alpha\mathbf{P}\|_{\mathrm{F}}^2 = \|\log\alpha\mathbf{I} + \log\mathbf{P}\|_{\mathrm{F}}^2\\ &= \|\log\alpha\mathbf{I}\|_{\mathrm{F}}^2 + 2\log\alpha\,\mathrm{tr}(\log\mathbf{P}) + \|\log\mathbf{P}\|_{\mathrm{F}}^2\end{aligned}$$

where the first equality follows from the fact that $\alpha\mathbf{I}$ and $\mathbf{P}$ are both positive definite and commute with each other. Since $\mathbf{P} \in \mathbf{SO}(3)$, its logarithm $\log\mathbf{P}$, which maps to the Lie algebra of $\mathbf{SO}(3)$, is a skew-symmetric matrix and thus we have $\mathrm{tr}(\log\mathbf{P}) = 0$. Moreover, due to Rodrigues' rotation formula (Engø 2001), the Frobenius norm of $\mathbf{P}$ can be written as

$$\|\log\mathbf{P}\|_{\mathrm{F}}^2 = 2\theta^2 \tag{7}$$

where $\theta = \arccos\left[\dfrac{\mathrm{tr}(\mathbf{P}) - 1}{2}\right] \in [0, \pi]$ is the rotation angle around a unit 3D axis given by $\theta^{-1}\log\mathbf{P}$ whenever $\theta \neq 0$. Therefore, the Frobenius norm of $\hat{\mathbf{R}}$ eventually becomes

$$\|\hat{\mathbf{R}}\|_{\mathrm{F}}^2 = \|\log\alpha\mathbf{I}\|_{\mathrm{F}}^2 + \|\log\mathbf{P}\|_{\mathrm{F}}^2 = 3(\log\alpha)^2 + 2\theta^2 \tag{8}$$

Then, we can approximate the loss $\ell(\hat{\mathbf{T}}, \mathbf{T})$ in Eq. (3) by combining the resultant geodesic distance and the distance between $\mathbf{T^{-1}\hat{T}}$ to its projection onto $\mathbf{SR}(3)$,

$$\begin{aligned}\hat{\ell}(\hat{\mathbf{T}}, \mathbf{T}) = 3\left[\log\frac{\mathrm{tr}(\mathbf{D\Sigma})}{3}\right]^2 &+ 2\arccos^2\left[\frac{\mathrm{tr}(\mathbf{P}) - 1}{2}\right]\\ &+ \lambda\,\|\mathbf{R_{\Pi}}\|_{\mathrm{F}}^{\mathbf{2}}\end{aligned} \tag{9}$$

with the following projection residual

$$\mathbf{R_{\Pi}} = \mathbf{T^{-1}\hat{T}} - \underset{\mathbf{S}^{+}(1)\times\mathbf{SO}(3)}{\Pi}(\mathbf{T^{-1}\hat{T}})$$

where λ is a positive weight on the projection distance, and it will be fixed to one in experiments. Minimizing the projection residual can minimize the deviation incurred by projecting $\mathbf{T^{-1}\hat{T}}$ onto $\mathbf{SR}(3)$.

B. **Reparameterized Homographies**: the second challenge we need to address is how to parameterize the matrices of homographies so that we can compute their geodesic distances in the underlying Lie group. A straightforward approach is to let the decoder network output the homograhy matrices directly, followed with a determinant normalization. However, a 2D homography matrix is composed of several parts of different transformations

$$\mathbf{T} = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12} & \mathrm{T}_{13}\\ \mathrm{T}_{21} & \mathrm{T}_{22} & \mathrm{T}_{23}\\ \mathrm{T}_{31} & \mathrm{T}_{32} & \mathrm{T}_{33}\end{bmatrix}$$

where the topleft 2×2 submatrix accounts for the rotation and scaling component, and $[\mathrm{T}_{13}, \mathrm{T}_{23}]$ represents the translation component. Since different components of homographies have various scales of values, a direct parameterization of homography matrices could result in unstable performances on decoding them jointly.

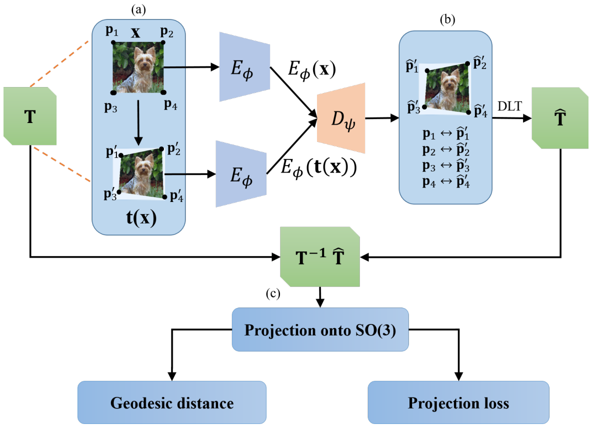
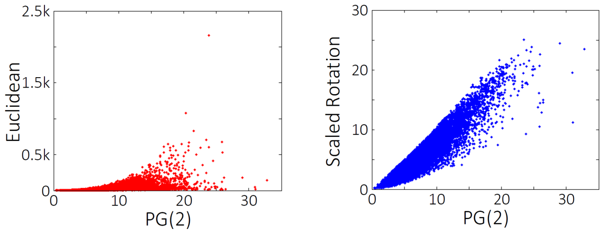


Figure 2: This figure illustrates the end-to-end pipeline.



(a) $\mathbf{PG}(2)$ *vs.* Euclidean (b) $\mathbf{PG}(2)$ *vs.* $\mathbf{SR}(3)$

Figure 3: The scatter plot of $\mathbf{PG}(2)$ *vs.* (a) Euclidean and (b) Scaled Rotation distances.

	KNN	1-FC	2-FC	3-FC	conv
AETv1 (2019)	22.39	16.65	9.41	9.92	7.82
(Ours) AETv2	**21.26**	**15.03**	**9.09**	**9.55**	**7.44**

Table 1: Comparison with different classifiers on top of learned representations for evaluation, where n-FC denotes a n-layer fully connected classifier, and $K = 10$ for KNN.

This motivates us to use an indirect parameterization by using four-point correspondence under a homography. Formally, consider the correspondences $\{\mathbf{p}_i \leftrightarrow \mathbf{p}'_i \mid i = 1,2,3,4\}$ each of which maps a homogeneous coordinate $\mathbf{p}_i = [x_i, y_i, w_i]^\intercal$ from the original image to the corresponding point $\mathbf{p}'_i = [x'_i, y'_i, w'_i]^\intercal$ after the transformation. The associated homography matrix $\mathbf{T}$ can be derived by solving the equation $\mathbf{p}'_i \times \mathbf{T}\mathbf{p}_i = 0, i = 1,2,3,4$ with the cross-product $\times$. This can be further rewritten as $\mathbf{A}_i\overline{\mathbf{T}} = \mathbf{0}$ by staking $\mathbf{T}$ columnwise into a vector $\overline{\mathbf{T}}$ with

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{0}^\intercal & -w_i{}'\mathbf{p}_i^\intercal & y_i{}'\mathbf{p}_i^\intercal \\ w'_i\mathbf{p}_i^\intercal & \mathbf{0}^\intercal & -x'_i\mathbf{p}_i^\intercal \end{bmatrix}$$

After assembling four 2×9 matrices $\{\mathbf{A}_i\}$ into the 8×9 matrix $\mathbf{A}$, $\overline{\mathbf{T}}$ is solved by the Direct Linear Transformation (DLT) as the singular vector of the smallest singular value of $\mathbf{A}$ up to a scalar factor. Furthermore, to ensure the obtained homography matrix invariant to the change of image coordinates under the DLT as well as to make it robust against noises on the estimated correspondences, the coordinates of four correspondence points are normalized in the original and transformed images, respectively.

Formally, four points $\mathbf{p}_i$ in the original image are fixed to be its four corners in this paper. The decoder D_ψ is designed to output its estimate of each point $\mathbf{p}'_i$ after the transformation. Then, two transformations $\mathbf{S}$ and $\mathbf{S}'$ with only translation and scaling are formed respectively to normalize $\{\mathbf{p}_i\}$ and $\{\mathbf{p}'_i\}$ so that they are centered at the origin and have an average distance of $\sqrt{2}$ to the center. After solving the normalized homography $\tilde{\mathbf{T}}$ from the normalized coordinates, the unnormalized version $\mathbf{T}$ is obtained from $\mathbf{T} = \mathbf{S}'^{-1}\tilde{\mathbf{T}}\mathbf{S}$. Since both SVD and matrix inverse are differentiable, the network outputting such reparameterized homographies can be trained end-to-end by the gradient descent method.

Comparison between EU and $\mathbf{SR}(3)$ To illustrate how well $\mathbf{SR}(3)$ can approximate $\mathbf{PG}(2)$, we provide a comparison between the Euclidean distances and the scaled rotation distances. We randomly draw $10,000$ homgraphies based on (5) by sampling $\mathbf{R}$ with its entries drawn from the unit Gaussian distribution. This allows us to accurately compute the geodesic distances between the drawn homographies and the identity in $\mathbf{PG}(2)$ with known $\mathbf{R}$. Meanwhile, by projecting the obtained homographies onto $\mathbf{SR}(3)$, we can compute the geodesic distances between the projected transformations and the identity in $\mathbf{SR}(3)$ as in (8). We plot the scatter points between the obtained $\mathbf{PG}(2)$ and Euclidean distances in Figure 3(a), as well as that between the $\mathbf{PG}(2)$ and the scaled rotation distances in Figure 3(b). The scaled rotation distance is highly correlated with the $\mathbf{PG}(2)$ distance, since the Pearson correlation coefficient between the $\mathbf{PG}(2)$ and the scaled rotation distances is as high as 0.9303, compared with that of 0.5347 between the $\mathbf{PG}(2)$ and the Euclidean distances. This shows that $\mathbf{SR}(3)$ provides a better approximation to the intractable geodesic distance in the homograhy group, and we can minimize it as a surrogate to train the AET model. From this observation, we believe the AETv2 based on the minimization of the scaled rotation distances ought to outperform the AETv1 that minimizes the Euclidean distances between transformations.

An End-to-end Pipeline In Figure 2, we illustrate how to train AETv2 in an end-to-end pipeline. An input image is first transformed by moving each of four corners $\mathbf{p}_i$ to the corresponding location $\mathbf{p}'_i$ with a randomly sampled homography $\mathbf{T}$. Through Siamese representation encoders E_ϕ and a transformation decoder D_ψ, four correspondence points $\hat{\mathbf{p}}'_i$ after the transformation are predicted, and the estimate $\hat{\mathbf{T}}$ of the input homography is made by the normalized DLT from the SVD of $\mathbf{A}$. This provides the reparameterization of $\mathbf{T}$ as a function of the four-point correspondence. Then, the resultant $\mathbf{T}^{-1}\hat{\mathbf{T}}$ is projected onto a subgroup $\mathbf{SR}(3)$ that admits a tractable geodesic distance. Since both the normalized DLT and the subgroup projection are differentiable, the AETv2 can be trained end-to-end by minimizing (9).

Method	Error rate
Supervised NIN (Lower Bound)	7.20
Random Init. + conv (Upper Bound)	27.50
Roto-Scat + SVM (2015)	17.7
ExamplarCNN (2014)	15.7
DCGAN (2015)	17.2
Scattering (2017)	15.3
RotNet + FC (2018)	10.94
RotNet + conv (2018)	8.84
AETv1 + FC (2019)	9.41
AETv1 + conv (2019)	7.82
(Ours) AETv2 + FC	**9.09**
(Ours) AETv2 + conv	**7.44**

Table 2: Comparison of unsupervised models on CIFAR-10. The fully supervised NIN and the random Init. + conv have the same three NIN blocks, but they are fully supervised and is trained with the first two blocks randomly initialized and staying frozen during training, respectively. "+FC" and "+conv" denote a nonlinear classifier with a hidden FC layer and a NIN convolutional block respectively, followed by a 10-way softmax layer.

Method	Conv4	Conv5
ImageNet Labels (Upper Bound)	59.7	59.7
Random (Lower Bound)	27.1	12.0
Tracking (2015)	38.8	29.8
Context (2015)	45.6	30.4
Colorization (2016)	40.7	35.2
Jigsaw Puzzles (2016)	45.3	34.6
BiGAN (2016)	41.9	32.2
NAT (2017)	-	36.0
DeepCluster (2018)	-	44.0
RotNet (2018)	50.0	43.8
AETv1 (2019)	53.2	47.0
(Ours) AETv2	**54.3**	**47.5**

Table 3: Top-1 accuracy on ImageNet. After unsupervised training of AlexNet, nonlinear classifiers are trained on top of Conv4 and Conv5 layers with labeled data for the evaluation purpose. The fully supervised and random models are also compared that give upper and lower bounded performances, respectively. A single crop is applied with no dropout or local response normalization during the testing.

Experiments

In this section, we present our experiment results by comparing the AETv2 with the AETv1 as well as the other unsupervised models. Following the standard evaluation protocol in literature (Zhang et al. 2019; Qi et al. 2019; Oyallon and Mallat 2015; Dosovitskiy et al. 2014; Radford, Metz, and Chintala 2015; Oyallon, Belilovsky, and Zagoruyko 2017; Gidaris, Singh, and Komodakis 2018), we will adopt downstream classification tasks to evaluate the learned representations on CIFAR10, ImageNet, and Places datasets.

CIFAR-10 Experiments

Network and Experiment Details To make a fair and direct comparison with existing unsupervised models, we adopt the Network-In-Network (NIN) architecture previously used on the CIFAR-10 dataset for the unsupervised learning task (Zhang et al. 2019; Qi et al. 2019; Oyallon and Mallat 2015; Dosovitskiy et al. 2014; Radford, Metz, and Chintala 2015; Oyallon, Belilovsky, and Zagoruyko 2017; Gidaris, Singh, and Komodakis 2018). The NIN consists of four convolutional blocks, each having three convolutional layers. Two Siamese NIN branches are then constructed in the AETv2, each of which takes the original and the transformed images, respectively. The outputs from the last block of two branches are concatenated and average-pooled to form a 384-d feature vector. An output layer follows to predict the eight parameters of the input homography transformation. The model is trained by the Adam solver with a learning rate of 10^{-5}, a value of 0.9 and 0.999 for β_1 and β_2, and a weight decay rate of 5×10^{-4}.

A classifier is then built on top of the second convolutional block to evaluate the quality of the learned representation following the standard protocol in literature (Zhang et al. 2019; Qi et al. 2019; Oyallon and Mallat 2015; Dosovitskiy et al. 2014; Radford, Metz, and Chintala 2015; Oyallon, Belilovsky, and Zagoruyko 2017; Gidaris, Singh, and Komodakis 2018). In particular, the first two blocks are frozen when the classifier atop is trained with labeled data.

Both model-based and model-free classifiers are trained for the evaluation purpose. First, we train a non-linear classifier with various numbers of Fully-Connected (FC) layers. Each hidden layer has 200 neurons followed by a batch-normalization and ReLU activation. We also train a convolutional classifier by adding a third NIN block on top of the unsupervised features, and its output feature map is averaged pooled and connected to a linear softmax layer. Alternatively, we test a model-free KNN classifier based on the averaged-pooled features from the second convolutional block. Without explicitly training a model with labeled data, the KNN classifier can make a direct assessment on the quality of unsupervised feature representations.

Results Table 2 compares the AETv2 with the other models on CIFAR-10. On one hand, it outperforms the AETv1 as well as the other unsupervised models with the same backbone. Furthermore, it narrows the performance gap with the fully supervised convolutional classifier (7.44% vs. 7.20%) that gives the lower bound of error rate when all labels are used to train the model end-to-end. More comparisons with the AETv1 are made in Table 1. We compare the performances by both the model-based and model-free classifiers in the downstream tasks for both versions of the AET models. From the results, we can see that the AETv2 consistently outperforms its AETv1 counterpart with both KNN classifiers and different numbers of fully connected layers.

ImageNet Experiments

Network and Experiment Details We further evaluate the performance on the ImageNet dataset. For a fair comparison with the AETv1, the AlexNet is used as the back-

Method	Conv1	Conv2	Conv3	Conv4	Conv5
Places labels (2014)	22.1	35.1	40.2	43.3	44.6
ImageNet labels	22.7	34.8	38.4	39.4	38.7
Random	15.7	20.3	19.8	19.1	17.5
Random rescaled (2015)	21.4	26.2	27.1	26.1	24.0
Context (2015)	19.7	26.7	31.9	32.7	30.9
Context Encoders (2016)	18.2	23.2	23.4	21.9	18.4
Colorization (2016)	16.0	25.7	29.6	30.3	29.7
Jigsaw Puzzles (2016)	23.0	31.9	35.0	34.2	29.3
BiGAN (2016)	22.0	28.7	31.8	31.3	29.7
Split-Brain (2017)	21.3	30.7	34.0	34.1	32.5
Counting (2017)	**23.3**	**33.9**	36.3	34.7	29.6
RotNet (2018)	21.5	31.0	35.1	34.6	33.7
AETv1 (2019)	22.1	32.9	37.1	36.2	34.7
AETv2	22.8	33.2	**38.1**	**36.8**	**35.3**

Table 4: Top-1 accuracy on the Places dataset. Different layers of feature maps are spatially resized to about $9,000$ elements, and a 205-way linear classifier is trained atop. All unsupervised features are pre-trained on the ImageNet, and are frozen when training the classification layer with Places labels. The fully-supervised networks trained with Places Labels and ImageNet labels are also compared, in addition to random models. The best accuracies are highlighted in bold and the second best values are underlined.

bone to learn the unsupervised features. Two branches with shared parameters are created by taking original and transformed images as inputs to train the unsupervised model. The $4,096$-d output features from the second last fully connected layer in two branches are concatenated and fed into the output layer producing eight projective transformation parameters. We still use the Adam solver to train the network with a batch size of 768 original and transformed images.

Results Table 3 reports the Top-1 accuracies of compared methods on ImageNet with the evaluation protocol used in (Noroozi and Favaro 2016), where Conv4 and Conv5 denote the training of AlexNet with the labeled data, after the bottom convolutional layers up to Conv4 and Conv5 are pretrained in an unsupervised fashion and frozen thereafter. The results show that in both settings, the AETv2 outperforms the other unsupervised models including the AETv1. The performance gap to the fully supervised models that give the upper bounded performance has been further narrowed to 5.4% and 12.2%. To evaluate the quality of unsupervised representations, a weak $1,000$-way fully connected linear classifier is trained on top of different numbers of convolutional layers. The results are shown in Table 5, and the AETv2 again achieves the best Top-1 accuracy among the compared models. This shows that the AETv2 can learn a high-quality unsupervised representation with superior performances even though a weaker classifier is used.

Places Experiments

We conduct experiments to evaluate unsupervised models on the Places dataset. An unsupervised representation is first pretrained on the ImageNet, and a single-layer softmax classifier is trained on top of different layers of the feature maps with Places labels. In this way, we assess how well unsupervised features can generalize across datasets. As shown in Table 4, the AETv2 outperforms the compared unsupervised models again, except for Conv1 and Conv2 in which case Counting performs slightly better.

Method	Conv1	Conv2	Conv3	Conv4	Conv5
ImageNet Labels	19.3	36.3	44.2	48.3	50.5
Random	11.6	17.1	16.9	16.3	14.1
Random rescaled (2015)	17.5	23.0	24.5	23.2	20.6
Context (2015)	16.2	23.3	30.2	31.7	29.6
Context Encoders (2016)	14.1	20.7	21.0	19.8	15.5
Colorization (2016)	12.5	24.5	30.4	31.5	30.3
Jigsaw Puzzles (2016)	18.2	28.8	34.0	33.9	27.1
BiGAN (2016)	17.7	24.5	31.0	29.9	28.0
Split-Brain (2017)	17.7	29.3	35.4	35.2	32.8
Counting (2017)	18.0	30.6	34.3	32.5	25.7
RotNet (2018)	18.8	31.7	38.7	38.2	36.5
AETv1 (2019)	19.2	32.8	40.6	39.7	37.7
(Ours) AETv2	**19.6**	**34.1**	**41.9**	**40.4**	**37.9**
DeepCluster* (2018)	13.4	32.3	41.0	39.6	38.2
AETv1* (2019)	19.3	35.4	44.0	43.6	42.4
(Ours) AETv2*	**21.2**	**36.9**	**45.9**	**44.7**	**43.2**

Table 5: Top-1 accuracy on ImageNet, where a $1,000$-way linear classifier is trained on top of different convolutional layers of feature maps spatially resized to about $9,000$ elements. Fully supervised and random models show the upper and the lower bounds of performances. Only a single crop is used during testing, except that the models marked with "*" apply ten crops.

Conclusions

We present a novel paradigm of Auto-Encoding Transformations (AET) to train unsupervised representations without involving labeled data. From the geometry of transformation, it seeks to minimize the geodesic distance between the input and predicted homographies. Compared with the conventional mean-squared error in the forbidden ambient Euclidean space, like AETv1, our AETv2 provides a natural error metric that enables to characterize how a transformation continuously moves to another one along the manifold of homographies. To this end, we tackle the intractable Riemannian logarithm by projecting the estimated homographies onto a subgroup that admits a tractable form of geodesic distance. Moreover, while a direct parameterization of transformations would mix up different transformation components with incomparable scales of values, we present a reparameterization of homographies instead from four-point correspondence. Since both the subgroup projection and the reparameterization are differentiable, it allows us to train the AET network end-to-end. Finally, we conduct experiments to demonstrate the superior performance of the pre-trained unsupervised representations on various downstream recognition tasks of multiple datasets.

References

Beutelspacher, A.; Albrecht, B.; and Rosenbaum, U. 1998. *Projective geometry: from foundations to applications*. Cambridge University Press.

Bojanowski, P.; and Joulin, A. 2017. Unsupervised learning by predicting noise. *arXiv preprint arXiv:1704.05310* .

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. *arXiv preprint arXiv:1807.05520* .

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2016. Deep image homography estimation. *arXiv preprint arXiv:1606.03798* .

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 1422–1430.

Donahue, J.; Krähenbühl, P.; and Darrell, T. 2016. Adversarial feature learning. *arXiv preprint arXiv:1605.09782* .

Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, 766–774.

Engø, K. 2001. On the BCH-formula in so (3). *BIT Numerical Mathematics* 41(3): 629–632.

Erlik Nowruzi, F.; Laganiere, R.; and Japkowicz, N. 2017. Homography estimation from image pairs with hierarchical convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 913–920.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised Representation Learning by Predicting Image Rotations. *arXiv preprint arXiv:1803.07728* .

Kolesnikov, A.; Zhai, X.; and Beyer, L. 2019. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005* .

Krähenbühl, P.; Doersch, C.; Donahue, J.; and Darrell, T. 2015. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856* .

Le, H.; Liu, F.; Zhang, S.; and Agarwala, A. 2020. Deep Homography Estimation for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7652–7661.

Nguyen, T.; Chen, S. W.; Shivakumar, S. S.; Taylor, C. J.; and Kumar, V. 2018. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters* 3(3): 2346–2353.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 69–84. Springer.

Noroozi, M.; Pirsiavash, H.; and Favaro, P. 2017. Representation learning by learning to count. In *The IEEE International Conference on Computer Vision (ICCV).*

Oyallon, E.; Belilovsky, E.; and Zagoruyko, S. 2017. Scaling the scattering transform: Deep hybrid networks. In *International Conference on Computer Vision (ICCV).*

Oyallon, E.; and Mallat, S. 2015. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2865–2873.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2536–2544.

Qi, G.-J. 2019. Learning Generalized Transformation Equivariant Representations via Autoencoding Transformations. *arXiv preprint arXiv:1906.08628* .

Qi, G.-J.; Zhang, L.; Chen, C. W.; and Tian, Q. 2019. AVT: Unsupervised Learning of Transformation Equivariant Representations by Autoencoding Variational Transformations. *arXiv preprint arXiv:1903.10863* .

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* .

Wang, X.; and Gupta, A. 2015. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2794–2802.

Zacur, E.; Bossa, M.; and Olmos, S. 2014a. Left-invariant riemannian geodesics on spatial transformation groups. *SIAM Journal on Imaging Sciences* 7(3): 1503–1557.

Zacur, E.; Bossa, M.; and Olmos, S. 2014b. Multivariate tensor-based morphometry with a right-invariant Riemannian distance on GL+(n). *Journal of mathematical imaging and vision* 50(1-2): 18–31.

Zhang, L.; Qi, G.-J.; Wang, L.; and Luo, J. 2019. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2547–2555.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *European Conference on Computer Vision*, 649–666. Springer.

Zhang, R.; Isola, P.; and Efros, A. A. 2017. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, 487–495.