Unsupervised Object Representation Learning using Translation and Rotation Group Equivariant VAE

Anonymous Author(s)
Affiliation
Address
email

Abstract

In many imaging modalities, objects of interest can occur in a variety of locations and poses (i.e. are subject to translations and rotations in 2d or 3d), but the location and pose of an object does not change its semantics (i.e. the object's essence). That is, the specific location and rotation of an airplane in satellite imagery, or the 3d rotation of a chair in a natural image, or the rotation of a particle in a cryo-electron micrograph, do not change the intrinsic nature of those objects. Here, we consider the problem of learning semantic representations of objects that are invariant to pose and location in a fully unsupervised manner. We address shortcomings in previous approaches to this problem by introducing TARGET-VAE, a translation and rotation group-equivariant variational autoencoder framework. TARGET-VAE combines three core innovations: 1) a rotation and translation group-equivariant encoder architecture, 2) a structurally disentangled distribution over latent rotation, translation, and a rotation-translation-invariant semantic object vector, which are jointly inferred by the approximate inference network, and 3) a spatially equivariant generator network. In comprehensive experiments, we show that TARGET-VAE learns disentangled representations without supervision that significantly improve upon and avoid the pathologies of previous methods. Semantic representations learned by TARGET-VAE on images highly corrupted by rotation and translation approach those learned on consistently posed objects, dramatically improving clustering and pose inference on multiple datasets.

1 Introduction

In many imaging modalities, objects of interest are arbitrarily located and oriented within the image frame. Examples include airplanes in satellite images, galaxies in astronomy images [1], and even particles in single-particle cryo-electron microscopy (cryo-EM) micrographs [2]. However, neither the location nor the rotation of an object within an image frame changes the nature (i.e. the semantics) of the object itself. An airplane is an airplane regardless of where it is in the image, and different rotations of a particle in a cryo-EM micrograph are still projections of the same protein. Hence, there is great interest in learning semantic representations of objects that are invariant to their location and pose.

In general, unsupervised representation learning methods do not recover representations that disentangle the semantics of an object from its location or its pose. Popular unsupervised deep learning methods for images, such as variational autoencoders [3], usually use encoder-decoder frameworks in which an encoder network is learned to map an input image to an unstructured latent variable or distribution over latent variables which is then decoded back to the input image by the decoder network. However, the unstructured nature of the latent variables means that they do not separate into any specific and interpretable sources of variation. To achieve disentanglement, unstructured

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

methods have been proposed that encourage independence between latent variables [4, 5]. However, these methods make no prior or structural assumptions about what these latent variables should encode, even when some sources of variation e.g. an object's location or pose are prevalent and well-known. Recently, several methods have proposed structured models that explicitly model rotation or translation within their generative networks [6, 7, 8] by formulating the image generative model as a function mapping coordinates in space to pixel values. Although promising, only the generative portion of these methods is equivariant to rotation and translation and inference networks have lackluster performance due to poor inductive bias for these structured latents.

To address this, we propose TARGET-VAE, a Translation and Rotation Group Equivariant Variational Auto-Encoder. TARGET-VAE is able to learn semantic object representations that are invariant to pose and location from images corrupted by these transformations by structurally decomposing the image generative factors into semantic, rotation, and translation components. We perform approximate inference with a group equivariant convolutional neural network [9] and specially formulated approximate posterior distribution that allows us to disentangle the latent variables into a rotationally equivariant latent rotation, translationally equivariant latent translation, and rotation and translation invariant semantic latent variables. By combining this with a spatial generator network [6, 8, 7], our framework is completely invariant to rotation and translation, unsupervised, and fully differentiable; the model is trained end-to-end using only observed images. In comprehensive experiments, we show that TARGET-VAE accurately infers the rotation and translation of objects without supervision and learns high quality object representations even when training images are heavily confounded by rotation and translation. We then show that this framework can be used to map continuous variation in 2d images of proteins collected with cryo-EM.

2 Related Work

In the recent years, there has been a great progress in developing models for unsupervised semantic analysis of images using VAEs [10, 11, 6, 12], flow-based methods [13], and Generative Adversarial Networks (GAN) [14, 15, 16]. In these methods, usually the goal is to build a low-dimensional representation of the images, which can be sampled to reconstruct similar images with some variations, or to use the representations to identify clusters in the data. The unsupervised learning methods used for clustering objects, need to address the variations caused by the common transformations in the objects. One simple approach to building invariant representations is data augmentation. In this approach, training dataset plus samples generated by the applying multiple transformations on them, are used to learn the transformation-invariant representations in a lower-dimensional latent space [17, 18]. This training approach is not efficient and when it comes to severe transformations such as rotations with high angles, the model might not be able to map the original data samples and the rotated ones to the same cluster in the latent space. Another group of methods that use data augmentation for learning transformation-invariant features, are using contrastive learning [19, 20, 21]. These models learn to output similar encoding for the different transformed versions of the same data sample. Even though this training scheme has made great progress in the recent years, it is considered to be pre-training method which needs to be combined with some supervised training and is not completely unsupervised.

In some methods, the model is trained to learn an invariant representation by constraining the latent space in the different variations of beta-VAE [15, 4, 5]. In these methods, the latent space is not explicitly structured to separate the representation variable from the transformation ones. A more explicit constraint is imposed on the latent space in Spatial-VAE [6], where the latent space is divided to variables encoding the representation and transformations. The latent space that can capture each targeted transformation in a structured space, can theoretically disentangle them from the content of the image.

To be able to disentangle the representation, from the transformations, the inference model should be equivariant to the transformations. Convolutional layers are one of the most popular feature extraction methods in image processing, due their translation-equivariant features. But they are not rotation-equivariant. There has been many studies in designing transformation-equivariant and specially rotation-equivariant neural networks [9, 22, 23]. In Group equivariant Convolutional Neural Networks (G-CNNs) [9], the 2D rotation space is discretized and filters are also applied on this rotation dimension. In this method, without adding anymore parameters and by small computational overhead, the regular convolutional layers can reflect the rotation of the input images. None of the

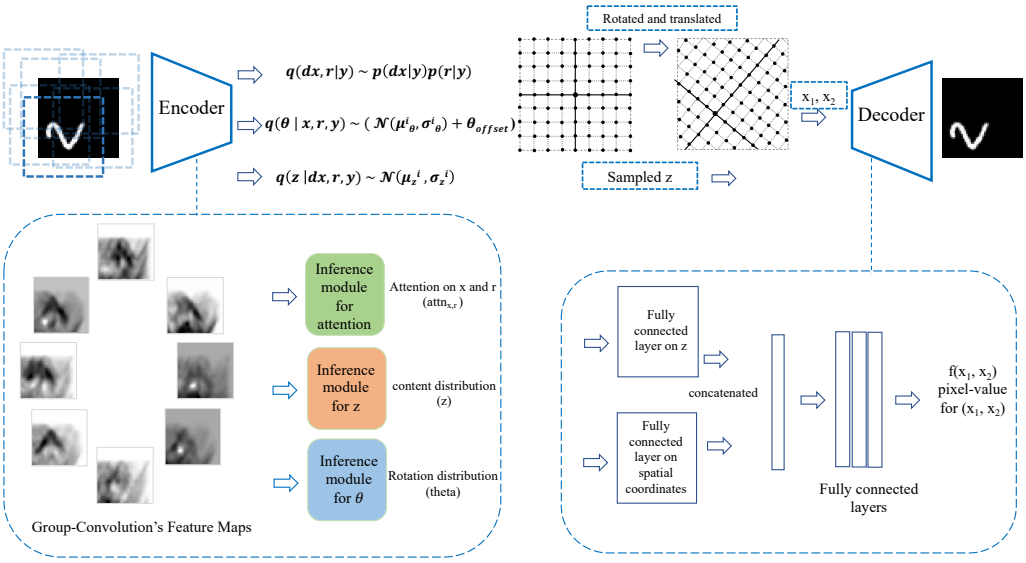

Figure 1: Overall framework where the encoder uses group-equivariant layers to output mixture distributions over representation,rotation and translation, and the transformation-equivariant generator reconstructs the image based on the representation vale z, and the transformed coordinates of the pixels.

previous methods in unsupervised object detection have included equivariance in their inference model.

Some image processing methods have proposed to train their model as a function that maps spatial coordinates of a pixel to their values [24, 25, 8, 7]. In these models, the generator learns to map the 2D or 3D coordinates of the image to its pixel values. However, in these studies, no direct inference on the rotation is performed. To the best of our knowledge, we are the first method to establish rotation and translation equivariance in both the inference and generator models of our proposed VAE framework, along with disentangling the mentioned transformations as structured variables in the latent space from the continuous representation variable of the objects.

3 Method

In this section, we present in details our proposed VAE framework that enables an unsupervised disentanglement of the rotation and translation of the objects from their representations. Figure 1 displays our proposed VAE-based framework.

3.1 Image generation process

Images can be considered as a combination of values for discreetly identifiable pixels. Transformation of an image, can also be considered as transforming the spatial coordinates of the pixels in the image. To have a generation process adherent to the transformations identified in the latent space, we define our generator as a function, which is mapping the spatial coordinates of the pixels to their values. The generator outputs a probability distribution as $p(\hat{y}_i|x_i, z)$, where z, x_i, and $\hat{y}_i$ are the latent representation variable, the spatial coordinates of a point, and the value for that point, respectively. Similar to [3], for an image with n pixels, we can define the probability of the image generated in this manner as sum over probability of its pixel values:

$$log\ p(\hat{y}|z) = \sum_{i=1}^{n} log\ p(\hat{y}_i|z, x_i) \tag{1}$$

We use a Cartesian coordinates with the origin at the center of the image. The translation and rotation of the coordinates can be done by shifting the origin, and rotating coordinates around it, respectively.

Depending on the value of the representation variable z, the generator acts as a function over the pixel coordinates to reconstruct the input image. In this setup the generation process is equivariant with regard to the transformations on the coordinate system. Assuming $R(\theta)$ as the rotation matrix for angle θ and translation value of t, the probability of the generated image from Formula 1, can be rewritten as

$$log\ p(\hat{y}|z,\theta,t) = \sum_{i=1}^{n} log\ p(\hat{y}_i|z, R(\theta)x_i + t) \tag{2}$$

In this study, we focus on disentangling the rotation and translation from the representation of data, but the image generation probability from Formula 2, can be extended the other transformations too.

3.2 Approximate inference on latent variables

We implement a VAE framework to perform approximate inference on representation, rotation, and translation latent variables. Since rotation optimization is non-convex, we have discretized 2D rotation space to r values, each related to a specific $k\frac{2\pi}{r}$ angle, where $k \in \{0, 1, ..., r-1\}$. As a result, our posterior distributions are conditioned on an additional factor r. We represent the approximate posterior as $q(z, \theta, t, r|y)$, where z is the representation value, θ is the rotation angle, and t and r refer to the translation and discretized rotation values, and y is the input image. This posterior distribution is intractable, and making the simplifying assumption that q(zlt,r,y) and q(θlt,r,y) are independent, we can factorize the posterior distribution as

$$q(z,\theta,t,r|y) = q(z|t,r,y)q(\theta|t,r,y)q(t,r|y) \tag{3}$$

, where $q(t, r|y)$ identifies a 3D distribution over t and r. The other two posterior distributions, $q(\theta|t, r, y)$ and $q(z|t, r, y)$, are mixture distributions conditioned on r, t, and the input y.

We use $\mathcal{N}(0, I)$ for the prior distribution over z. The prior over r and θ can be uniform or Gaussian depending on our knowledge about the distribution of rotation values in the dataset. To calculate the prior over r, we calculate the probability of the discrete rotation angles related to each r dimension, which we call θ_{offset}. For example, when $r = 4$, we have rotation values of $\theta_{offset} = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. The prior over translation values x is set to be Gaussian with mean 0, which denotes the center of the 2D coordinate system. The standard deviation of this prior depends on the number of pixels in our fixed-size coordinate system and whether we want the model to be flexible with large translation values or not. Lower standard deviation for the prior over translations, will cause the model to pose higher loss towards larger predicted translation values. Using these priors, we calculate the Kullback-Leibler (KL) divergence for the latent variables:

$$KL(q(z,\theta,t,r|y)||p(z,\theta,t,r|y)) = \sum_{z,\theta,t,r} q(z,\theta,t,r|y) log \frac{q(z,\theta,t,r|y)}{p(z,\theta,t,r)} \tag{4}$$

To simplify this equation, we us the factorization from Equation 3:

$$KL(q(z,\theta,t,r|y)||p(z,\theta,t,r|y)) = KL_{t,r} + \sum_{t,r} q\left(t,r\right)\left(KL_\theta + KL_z\right) \tag{5}$$

, where $KL_{t,r} = \sum_{t,r} q(t,r|y) log \frac{q(t,r|y)}{p(t,r)}$, $KL_\theta = KL\left(q\left(\theta|t,r,y\right)||p\left(\theta|t,r,y\right)\right)$, and $KL_z = KL\left(q\left(z|t,r,y\right)||p\left(z|t,r,y\right)\right)$.

We give a detailed calculation of the KL-divergence in Appendix A1. Multiplication of Kl_θ and KL_z with $q(t, r)$ weights the KL-divergence on θ and z based on the joint posterior distributions over the discretized rotation and translation values. We can calculate $p(t, r|y)$ by multiplying $p(t|y)$ and $p(r|y)$, and $p(z|t, r, y)$ is the same as as $p(z|y)$ which is $\mathcal{N}(0, I)$. The $p(\theta|t, r, y)$ cannot be simplified to $p(\theta|y)$ since the value of θ is dependent on the θ_{offset} of r. However, $p(\theta|t, r, y)$ can be written as $p(\theta|r, y)$ since it is independent of t translation values. We define this prior as $\mathcal{N}(\theta_{offset}, \frac{\pi}{r})$, which means that the Gaussian prior for each r dimension has a mean equal to the rotation of filters in that dimension. Also, we are tightening the prior distributions by lowering their standard deviation as r grows, so that the prior distribution for the r dimension would not overlap.

The full variational lower-bound for our model in this VAE framework is calculated based on the reconstruction loss and the Kl-divergence:

$$\mathop{\mathbb{E}}_{(z,\theta,t)\sim\, q(z,\theta,t|y)}[log\ p(\hat{y}|z,\theta,t)] \quad - \quad KL(q(z,\theta,t,r|y)||p(z,\theta,t,r|y)) \tag{6}$$

, where we calculate the KL-divergence value in closed form using Equation 5 during the training.

3.3 Neural network architecture

Convolutional layers are translation equivariant meaning that the spatial shift in the image causes the same shift in their output feature maps. To incorporate rotation-equivariance in the inference model, we choose to use group convolutional layers from [9]. A set X with an operation ., which is associative on the set, and has inverse and identity elements, specifies a symmetry group G. Furthermore, combination of two symmetry transformations on a set, also creates a symmetry group. We can consider the regular convolutional layers as a function $f : \mathbb{Z}^2 \rightarrow \mathbb{R}^k$, that maps every pixel to a vector of k values where k is the number of kernels in the convolutional layer. The convolutional layer does this by sliding the kernels on the input image. Replacing this shifting operation with transformations from group G, changes the function to a function over group G [26].

We define p_r group where $r \in \mathbb{N}$, and the operation consists of all translations, and r rotations, about any center of rotation in a 2D space. In the group convolutional layer, each kernel is rotated by $k\frac{2\pi}{r}$ angles, where $k \in \{0, 1, ..., r-1\}$ and then we calculate its dot product with the input image or feature map. Rotating kernels happens in addition to the translation of them across the input. As a result, the output feature map of the p_n group convolutions has one extra dimension which corresponds to the dot-product of the kernels with different rotations.

As shown in Figure 1, we define three parallel 1x1 3D convolutional layers on the translation and rotation equivariant feature maps of the group convolutional layers. These layers output three log posterior distributions of $log\ q(t, r|y)$, $log\ q(z|t, r, y)$, and $log\ q(\theta|t, r, y)$. To build the association between the θ values of each r dimension and the rotation of the kernel in that r dimension, we add the mean values from the $q(\theta|t, r, y)$ with the respective θ_{offset} values of their r dimension.

During the training, we sample from the joint posterior distribution of $log\ q(t, r)$ using Gumbel-softmax [27], and use the sample to obtain z and θ values from the other two posterior distributions. We define a coordinate system where x is in [-1,1] range. Then we translate and rotate this coordinate system using the sampled t and θ values. The generator receives the sampled z value along with the coordinates a of point at every pass and outputs the value from that point in the image. We use random Fourier feature expansion [28] on the coordinates of the pixel before feeding it to the fully-connected layers. As shown in Figure 1, there are two sets of parallel fully-connected layers each with 512 hidden units, where one receives the coordinates of the pixel (or its Fourier feature expanded version), and the other one takes the z as input. The output of these layers are concatenated before being input to a final set of fully-connected layers to output the value for the pixel.

4 Results

4.1 Experiment setup

We run our experiments with p_4, p_8, and p_{16} group convolutional layers in the inference model. The input images are normalized to the range of [0,1]. We use 128 rectangular filters where size of the kernels in the first layer needs to be set to a value larger than the size of the targeted objects. We use leaky-ReLU activation functions. The batch size for all the experiments is set to 100. We use ADAM optimizer [29], with learning rate of 2e-4, and the learning rate is reduced by factor of 0.5 after no improvements in the loss for 10 epochs. We run the training for a maximum of 500 epochs with early-stopping in case of no improvements in the loss for 20 epochs. Most of our experiments converge in less than 200 epochs due to the early-stopping condition. We run all our experiments in a single NVIDIA A100 gpu, with 80 GB memory.

The fully-connected layers in the generator, each has 512 hidden units, and the output can be one probability value in the case of binary valued pixels, or mean and standard deviation if the pixels are real values. For the RGB images, the dimension of output for each pixel can be easily extended to three.

Table 1: Translation and rotation correlation on MNIST(N) and MNIST(U) with z_dim=2

	MNIST(N)		MNIST(U)	
Group Convolution	Translation	Rotation	Translation	Rotation
Spatial-VAE [6]	**0.994, 0.993**	0.564	**0.982, 0.983**	0.005
p_4	0.97, 0.972	0.814	0.975, 0.976	0.80
p_8	0.981, 0.981	0.89	0.972, 0.971	0.859
p_{16}	0.969, 0.972	**0.898**	0.974, 0.971	**0.93**

4.2 TARGET-VAE predicts the translation and rotation values of objects with high correlation

We evaluate how accurately, our proposed model can identify the translation and rotation values of the objects in the images. For this purpose, we create two datasets using hand-written digits in MNIST. In the first one, the digits are randomly rotated by sampling from $\mathcal{N}(0, \frac{\pi^2}{16})$ and randomly translated by sampling from $\mathcal{N}(0, 5^2)$ pixels. We call this dataset MNIST(N), because of the normal distribution over the rotation angles. In the second dataset, the digits are translated similar to the MNIST(N), but the rotations are randomly sampled from $U(0, 2\pi)$. We call this dataset MNIST(U). In both datasets, the training and test sets have 60000, and 10000 images where each image is 50x50 pixels. Appendix A2 shows some examples of the data in these datasets.

We train TARGET-VAE with p_4, p_8, and p_{16} on these generated datasets. For the prior over θ, we use $\mathcal{N}(0, \frac{\pi}{4})$ for the MNIST(N) dataset, and $U(0, 2\pi)$ for the MNIST(U). For $p(\theta|r)$ we calculate $p(\theta_{offset(r)}) \sim p(\theta)$, where $\theta_{offset(r)}$ is the rotation angle of the kernels in each dimension of r. In the training for both of the datasets, we set the prior over t to $\mathcal{N}(0, 5^2)$.

After the training, to evaluate our model's performance on the prediction of the translation and rotation of the objects, we use the samples from the test set as the input to the inference model to obtain $q(t, r|y)$ and $q(\theta|t, r, y)$. To identify the most-likely (t, r), we apply the softmax function on $q(t, r|y)$. By marginalize the $q(t, r)$ over r, we identify the translation value t, and by using the result of this softmax to sample from $q(\theta|t, r, y)$, we identify the rotation value θ. For the translation values, we calculate Pearson correlation, but because of the circular nature of the rotations values, we calculate the circular correlation [30] for them. Table 1, shows the correlation values on the test set for MNIST(N) and MNIST(U), respectively. The two values in the translation column, show the translation correlation for x_1 and x_2 coordinates. The values in these tables are related to the model trained with dimension of the representation space (z_dim) equal to 2. Our experiments show that changing dimension of the representation space, does not have a significant effect on the translation and rotation correlation values, but increasing the level of the discretization over the rotation dimension usually improves the rotation correlation. To the best of our knowledge, the only other method that performs direct inference on the rotation value to identify the invariant representation for the objects is spatial-VAE [6]. To compare our results, we run spatial-VAE, with the same number of parameters and z_dim=2. As shown in Table 1, in both MNIST(N) and MNIST(U) datasets, our proposed method, TARGET-VAE significantly improves the rotation correlation value.

Some of the digits in MNIST datasets, have natural ambiguity, with regard to their rotations. Investigating the correlation between the predicted rotation values, and the ground-truth ones, shows a difference about π between the some of the predicted angles and their ground-truth ones. Figure 2 shows this difference. By drawing digitwise rotation correlation figures in appendix A.3, we show that this π difference mostly belongs to the three digits of 0, 1, and 8. This shows how our model can acknowledge the rotation symmetry in these digits.

4.3 TARGET-VAE improves clustering accuracy by learning translation and rotation invariant features

By disentangling rotation and translation from the representation values, TARGET-VAE can identify the different clusters of the objects. Similar to 4.2, we use $softmax(q(t, r))$ to identify the most-probable representation from $q(z|t, r, y)$. We train TARGET-VAE with p_4, p_8, and p_{16} and evaluate their clustering performance using K-means algorithm [31] on MNIST(N) and MNIST(U). Table

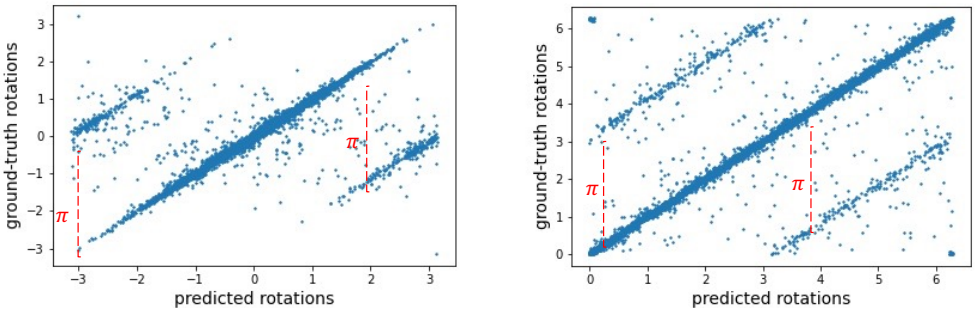

Figure 2: Left: Predicted θ vs. ground-truth data in MNIST(N), Right: Predicted θ vs. ground-truth data in MNIST(N), with TARGET-VAE and p_8. There is a π difference between the predicted angles and the ground-truth values of some of the digits, which is related to the rotation symmetry of them.

Table 2: Clustering accuracy (%) on MNIST(N) and MNIST(U)

Model	MNIST(N)	MNIST(U)
VAE (z_dim=4) [3]	15.3	12.8
Beta-VAE (z_dim=4) [4]	15.1	18.0
Spatial-VAE (z_dim=2) [6]	37.1	28.2
TARGET-VAE p_4 (z_dim=2)	56.4	56.6
TARGET-VAE p_8 (z_dim=2)	60.1	57.1
TARGET-VAE p_{16} (z_dim=2)	60.1	63.4
TARGET-VAE p_4 (z_dim=32)	65.1	64.3
TARGET-VAE p_8 (z_dim=32)	**77.7**	69.1
TARGET-VAE p_{16} (z_dim=32)	75.2	**71.2**

2 shows the clustering accuracies on these two datasets, where increasing the dimension of the representation space, helps the model with the separation of different digits and achieving higher clustering accuracy. Based on the accuracies in this table and rotation correlations in Table 1, higher discretization of the rotation space, allows the TARGET-VAE to have a better estimate of the rotation of the objects, and this is turn results in more rotation-invariant representation encoding and higher clustering accuracy. To compare the clustering performance of TARGET-VAE with other similar methods, we train a standard VAE [3], beta-VAE [4], and spatial-VAE [6] with about same number of parameters as TARGET-VAE, on MNIST(N) and MNIST(U) datasets. We set the z_dim=2 for both spatial-VAE and TARGET-VAE. Since the standard VAE and the beta-VAE do not have dedicated latent variable for identifying translation and rotation, we set the dimension of latent space to 4, so that it would have equal dimensions with the total size of the latent spaces in our model. Also, for the beta-VAE, we set β value to 4 based on the proposed settings in [4]. As shown in the Table 2, all variations of the TARGET-VAE outperform the other methods. We believe that the direct inference on the rotation and dedicating a structured latent variable to encode the rotation of the objects in TARGET-VAE, plus the transformation-equivariant generator, are the main reason for the dramatic increase in the clustering accuracy with our proposed method. In MNIST(N), since there are limited range for the rotation values, increasing the size of the group convolution, does not necessarily increase the rotation correlation or the clustering accuracy, where in MNIST(U) due to the larger variations in the rotation values, increasing the size of the group convolution leads to a better estimate of rotations and improved clustering accuracy.

Figure 3 shows the confusion matrix and t-sne plot for TARGET-VAE with p_8 group-convolution, on MNIST(N). Based on the confusion matrix, most of the digits are clustered correctly, and some misclassification is related to the confusion between rotated digits 6 and 9.

In appendix A.4, we perform experiments using images with multiple digits to show that TARGET-VAE can detect multiple objects in the evaluation stage. Furthermore, in appendix A.5 and A.6, we train and evaluate TARGET-VAE on the dSprites [32], and galaxy zoo [1]. We also perform an ablation study on the important components of our proposed model in Appendix A.7.

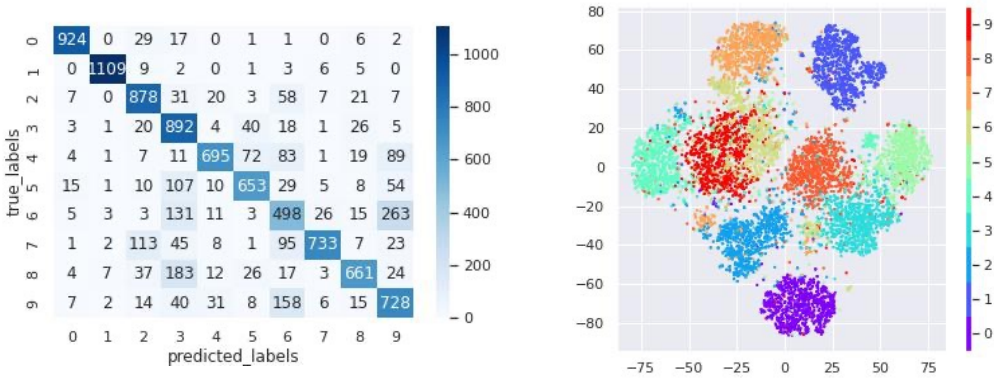

Figure 3: Confusion matrix of trained TARGET-VAE on MNIST(N) with z_dim=32 (left), and the t-sne figure of the 32 dimension latent representation space (right).

4.4 Identifying class-averages and protein conformations in the electron-microscopy images with TARGET-VAE

In the single particle electron microscopy images, biological particles can appear with different rotations and in different locations of the images. To be able to identify different types of particles or different conformations of the protein, the representation vector should be invariant to these transformations. Here, we demonstrate the capacity of TARGET-VAE in capturing different classes and variable conformations of particles in the electron microscopy images.

Identifying different classes of particles in EMPIAR-10025. The dataset EMPIAR-10025 [33] contains T20S Proteasome micrographs with 2.8 Å resolution. We use Topaz [34] package to create 161292 images of 400x400 pixels, where each image contains a particle. After downsampling by factor of 4 and normalizing the images, we train TARGET-VAE with p_8 and z_dim=2 on them. Due to the high-complexity of the structures of the proteins, we use 6 fully-connected layers with 512 units in the generator. In the rotation inference our prior over r is uniform, and the prior over θ, is $\mathcal{N}(0, \pi)$. We randomly select 10% of images as validation set to monitor the training process. As shown in Figure 4, our model identifies three clusters in this dataset. The image on the left side of the Figure 4 contains instances of input images which are extremely noisy.

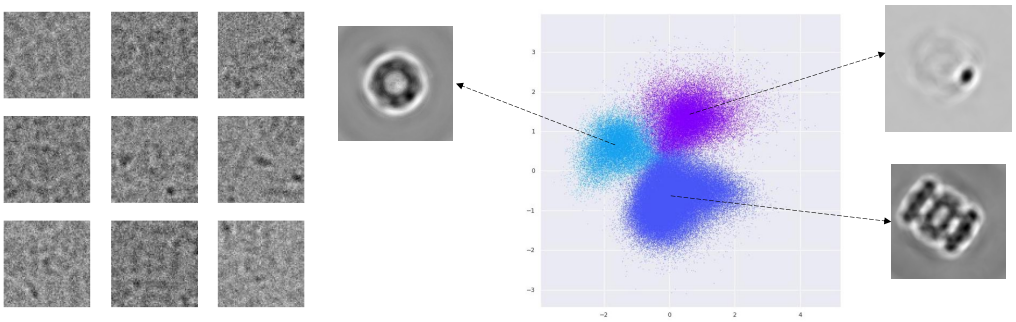

Figure 4: Left: Images of T20S Proteasome particles from EMPIAR-10025 dataset. Right: The 2D latent representation space for EMPIAR-10025 dataset learnt by TARGET-VAE, which identifies three clusters.

Identifying different views of particles in EMPIAR-10029. The EMPIAR-10029 dataset contains simulated EM dataset of GroEL particles. This dataset has 10000 images of shape 200x200, which we downsample to 100x100 and normalize. Similar to EMPIAR-10025, we train TARGET-VAE with the same settings as the EMPIAR-10025. The latent space does not identify separate clusters in this dataset, but we can perform a random walk in the latent space to identify the different projections of

this biological particle. We have also tested our model in identifying the heterogeneity in some other datasets, which we discuss in appendix A.8.

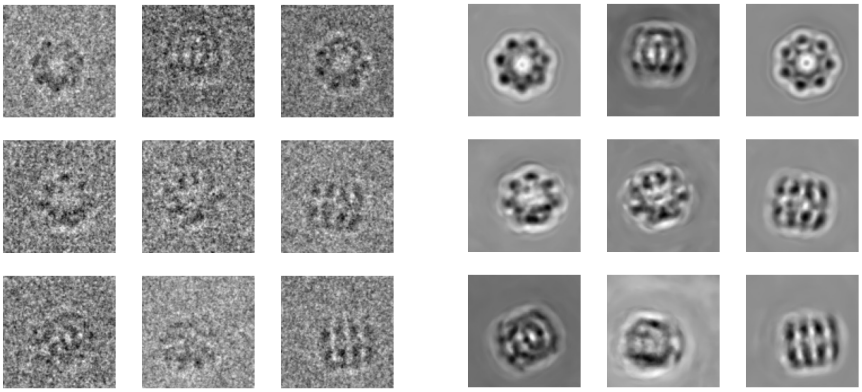

Figure 5: Left: Images of simulated EM dataset of GroEL particles from EMPIAR-10029 dataset. Right: The reconstructed particles with rotation angle set to zero.

5 Conclusion

In this paper, we presented TARGET-VAE, a translation and rotation group equivariant variational autoencoder framework for learning to detect and represent objects in images without supervision. By structuring the encoder to be translation and rotation equivariant, the model can identify the latent rotation and translation variables and disentangle them from the latent variables encoding the content of the image. The spatially equivariant generator samples from the posterior distribution to reconstruct the image. We showed that this approach not only accurately estimates the rotation and translation of objects, but also learns content representations that reflect known semantics (i.e., clusters match known semantic labels) across multiple datasets. Although we only consider 2d images containing single objects with rotation and translation, this framework can be extended to other object transformations by adopting encoder networks with the proper equivariance and adapting the latent distribution. In the future, we expect this framework can also be extended to multi-object detection, object tracking over time, and to 3d environments. TARGET-VAE lays the groundwork for a new generation of fully unsupervised object detection and semantic analysis methods for cryo-EM and other imaging modalities.

References

[1] M Jordan Raddick, Georgia Bracey, Pamela L Gay, Chris J Lintott, Carie Cardamone, Phil Murray, Kevin Schawinski, Alexander S Szalay, and Jan Vandenberg. Galaxy zoo: Motivations of citizen scientists. *arXiv preprint arXiv:1303.6886*, 2013.

[2] Fred J Sigworth. Principles of cryo-em single-particle image processing. *Microscopy*, 65(1):57–67, 2016.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[5] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

[6] Tristan Bepler, Ellen Zhong, Kotaro Kelley, Edward Brignole, and Bonnie Berger. Explicitly disentangling image content from translation and rotation with spatial-VAE. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2):176–185, 2021.

[8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[9] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[10] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in Neural Information Processing Systems*, 29, 2016.

[11] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

[12] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3412–3420, 2019.

[13] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[15] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[16] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021.

[17] Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547. IEEE, 2018.

[18] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4994–5002, 2019.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[21] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[22] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[23] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[24] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[25] Paul Andrei Bricman and Radu Tudor Ionescu. Coconet: A deep neural network for mapping pixel coordinates to color values. In *International Conference on Neural Information Processing*, pages 64–76. Springer, 2018.

[26] Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2019.

[27] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[28] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] S Rao Jammalamadaka and Ambar Sengupta. *Topics in circular statistics*, volume 5. world scientific, 2001.

[31] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[32] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[33] Melody G Campbell, David Veesler, Anchi Cheng, Clinton S Potter, and Bridget Carragher. 2.8 å resolution reconstruction of the thermoplasma acidophilum 20s proteasome using cryo-electron microscopy. *Elife*, 4:e06380, 2015.

[34] Tristan Bepler, Andrew Morin, Micah Rapp, Julia Brasch, Lawrence Shapiro, Alex J Noble, and Bonnie Berger. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nature methods*, 16(11):1153–1160, 2019.

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] Our proposed framework can perform inference on the rotation and translation values of the object and use this inference to learn a rotation and translation invariant representation value for them.
   (b) Did you describe the limitations of your work? [Yes] We mentioned the limitations in the conclusion in the form the future directions.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] To the best of our knowledge, our work does not have any negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and we believe that our paper conforms to that.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In the parts that we made any assumptions, we clearly explained them.
   (b) Did you include complete proofs of all theoretical results? [Yes] We included proof for one of our equations in section 3.2 which needs proof.
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] The code is available at ——————————

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All the details are either explained in the paper or can be found in the code.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] The variation in the results is not large enough to justify using error bars.

(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)? [Yes] We included all the processing information.

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]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All the data that we are using is the public domain.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data that we are using does not contain any personally identifiable information or offensive content.

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]