Unsupervised Object Representation Learning using Translation and Rotation Group Equivariant VAE

Alireza Nasiri
Simons Machine Learning Center
New York Structural Biology Center
anasiri@nysbc.org

Tristan Bepler
Simons Machine Learning Center
New York Structural Biology Center
tbepler@nysbc.org

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 representation, 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. When trained on images highly corrupted by rotation and translation, the semantic representations learned by TARGET-VAE are similar to those learned on consistently posed objects, dramatically improving clustering in the semantic latent space. Furthermore, TARGET-VAE is able to perform remarkably accurate unsupervised pose and location inference. We expect methods like TARGET-VAE will underpin future approaches for unsupervised object generation, pose prediction, and object detection. Our code is available at https://github.com/SMLC-NYSBC/TARGET-VAE.

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 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 locations and poses.

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 (VAE) [3], usually use encoder-decoder frameworks in which an encoder network is learned to map an input image to an unstructured latent variable

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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, 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 the 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, 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 significant progress in machine learning methods for unsupervised semantic analysis of images using VAEs [6, 10, 11, 12], flow-based methods [13], Generative Adversarial Networks (GAN) [14, 15, 16], and capsule networks [17]. These methods generally seek to learn a low-dimensional representation of each image in a dataset, or a distribution over this latent, by learning to reconstruct the image from the latent variable. The latent variable, then, must capture variation between images in the dataset in order for them to be accurately reconstructed. These latents can then be used as features for downstream analysis, as they capture image content. However, these representations must capture all information needed to reconstruct an image including common transformations that are not semantically meaningful. This often results in latent representations that group objects primarily by location, pose, or other nuisance variables, rather than type. One simple approach to address this is data augmentation. In this approach, additional images are created by applying a known set of nuisance transformations to the original training dataset. A network is then trained to be invariant to these transformations by penalizing the difference between low-dimensional representations of the same image with different nuisance transforms applied [18, 19]. However, this approach is data inefficient and, more importantly, does not guarantee invariance to the underlying nuisance transformations. This general approach has improved with recent contrastive learning methods [20, 21, 22], but these ultimately suffer from the same underlying problems.

Other methods approach this problem by applying constraints on the unstructured latent space [4, 5, 15]. In this general class of methods, the latent space is not explicitly structured to disentangle semantic latent variables from latents representing other transformations. Instead, the models need to be inspected post-hoc to determine if a meaningfully separated concepts have been encoded in each latent variable and what those concepts are. A few works have addressed explicitly disentangling the latent space into structured and unstructured variables. Sun et al. [17] propose a 3D point cloud representation of objects decomposed into capsules. Using a self-supervised learning scheme, their model learns transformation-invariant descriptors along with transformation-equivariant poses for each capsule, improving reconstruction and clustering. Bepler et al. [6] propose spatial-VAE, a VAE framework that divides the latent variables into unstructured, rotation, and translation components. However, only the generative part of spatial-VAE is equivariant to rotation and translation and the inference network struggles to perform meaningful inference on these transformations.

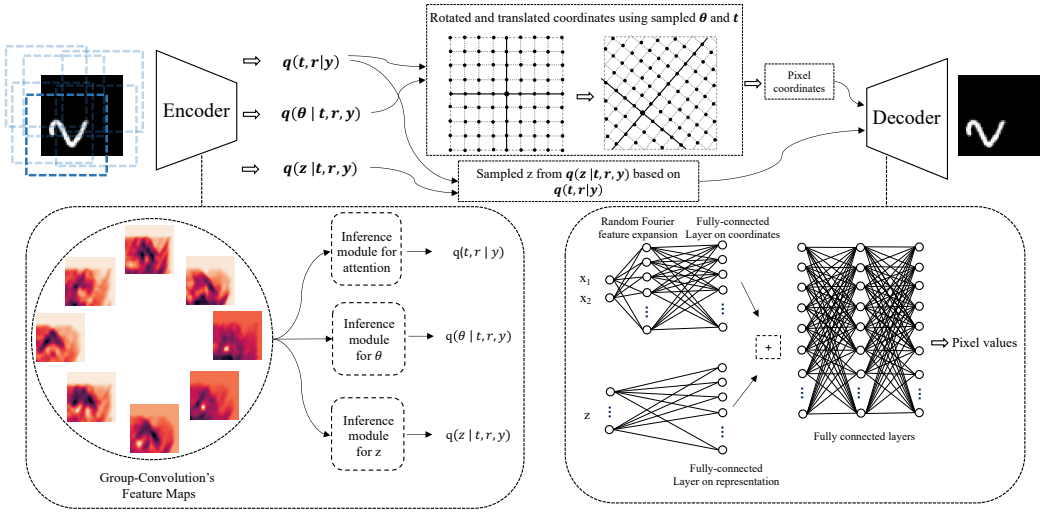

Figure 1: The TARGET-VAE framework. The encoder uses group-equivariant convolutional layers to output mixture distributions over semantic representations, rotation, and translation. The transformation-equivariant generator reconstructs the image based on the representation value z, and the transformed coordinates of the pixels.

To disentangle object semantic representations from spatial transformations, the inference network should also be equivariant to the spatial transformations. Translation-equivariance can be achieved by using convolutional layers in the inference model. However, convolutional layers are not rotation-equivariant. There has been many studies in designing transformation-equivariant and specifically rotation-equivariant neural networks [9, 23, 24, 25]. In Group-equivariant Convolutional Neural Networks (G-CNNs) [9], 2d rotation space is discretized and filters are applied over this rotation dimension in addition to the spatial dimensions of the image. This creates a network that is structurally equivariant to the discrete rotation group at the cost of additional compute. G-CNNs allow us to create a rotation-equivariant inference network.

A number of studies have proposed spatial transformation equivariant generative models, by modeling an image as a function that maps 2D or 3D spatial coordinates to the values of the pixels at those coordinates [7, 8, 26, 27, 6]. Due this mapping, any transformation of the spatial coordinates produces exactly the same transformation of the generated image. However, other than Bepler et al. [6], none of these studies perform inference on spatial transformations. To the best of our knowledge, TARGET-VAE is the first method to have rotation and translation equivariance in both the inference and the generative networks to achieve disentangling.

3 Method

TARGET-VAE disentangles image latent variables into an unstructured semantic vector, and rotation and translation latents (Figure 1). The approximate posterior distribution is factorized into translation, rotation, and content components where the joint distribution over translation and discrete rotation groups is defined by an attention map output by the equivariant encoder. The fine-grained rotation and content vectors are drawn from a mixture model where each of the joint translation and discrete rotation combinations are components of the mixture and are used to select mean and standard deviations of the continuous latents output by the encoder for each possible rotation and translation. Sampled translation, rotation, and content vectors are fed through the spatial decoder to generate images conditioned on the latent variables.

3.1 Image generation process

Images can be considered as the combination of discreetly identifiable pixels, and the spatial transformation of an image is equivalent to transforming the spatial coordinates of its pixels. To have an image generation process adherent to the transformations identified in the latent space, we define our

generator as a function which maps the spatial coordinates of the pixels to their values. The generator outputs a probability distribution as $p(\hat{y}_i|z, x_i)$, where z, x_i, and $\hat{y}_i$ are the latent content vector, the spatial coordinate of the ith pixel, and the value of that pixel, 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:

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

To define spatial coordinates over the pixels of the image, we use Cartesian coordinates with the origin at the center of the image. Translation and rotation of the image is achieved by shifting the origin and rotating coordinates around it, respectively. Depending on the value of the content vector, z, the generator acts as a function over the pixel coordinates to produce an image. In this setup, the generation process is equivariant to transformations of 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 is

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

In this study, we focus specifically on rotation and translation of the spatial coordinates, but the image generative process extends to any transformation of the coordinate space.

3.2 Approximate inference on latent variables

We implement a VAE framework to perform approximate inference on content, rotation, and translation latent variables. Since rotation optimization is non-convex, we implement a mixture distribution over the 2D rotation space that allows the model to choose from r discrete components. These components are used to approximate the posterior distributions over the rotation angle θ. We represent the overall approximate posterior as $q(z, \theta, t, r|y)$, where z is the latent content vector, θ is the rotation angle, t and r refer to the translation and discretized rotation components, and y is the input image. By making the simplifying assumption that $q(z|t, r, y)$ and $q(\theta|t, r, y)$ are independent, the approximate posterior distribution factorizes as

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

where $q(t, r|y)$ is the joint distribution over discrete translations and rotations and $q(\theta|t, r, y)$ and $q(z|t, r, y)$ are the distributions over the real-valued rotation and the latent content vector conditioned on t, r, and the input y.

The Kullback-Leibler (KL) divergence between the approximate posterior and the prior over the latent variables is then

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

To simplify this equation, we use the factorization from Equation 3 to reduce the KL-divergence to the following (see Appendix A1 for the full derivation), with the joint prior factorized into independent priors over the latent variables,

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

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

Multiplication of Kl_θ and KL_z with $q(t, r)$ in Equation 5, weights the KL-divergence on θ and z by the joint posterior distribution over t and r. We choose to make the priors, $p(t)$ and $p(r)$, independent giving $p(t, r) = p(t)p(r)$, but this is straightforward to relax if a non-independent prior is desired.

We define the prior on θ to be independent of the translation and z to be independent of the rotation and translation. We model θ as being drawn from a mixture distribution where the mixture components are the discrete rotations, $p(\theta) = \sum_r p(\theta|r)p(r)$. Given r, θ is drawn from a Gaussian distribution with mean given by the angle offset defined by r and a fixed standard deviation which depends on the number of discrete rotation groups and is set to $\frac{\pi}{r}$. We tighten each of the θ mixture distributions by shrinking the standard deviation as r grows to reduce overlap between these distributions when the number of discrete rotations is large. For the prior on r, we usually use a uniform distribution, but a discretized normal distribution can also be used to bias the model towards certain rotation angles if a non-uniform rotation distribution is known a priori. Then, $p(r)$ is calculated based on θ_{offset} for each discrete rotation. For example, when $r = 4$, then $\theta_{offset} \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, which are the rotation angles of the discrete rotation group, P_4. The prior over translation, t, is 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 more heavily penalize large predicted translation values.

The full variational lower-bound for our model is

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

where the KL-divergence term is defined in Equation 5.

3.3 Neural network architecture

3.3.1 Group convolutional encoder

Convolutional layers are translation equivariant, meaning that a spatial shift in the image causes the same shift in the output feature map, but are not rotation-equivariant. To incorporate rotation-equivariance in the inference model, we use group convolutional layers [9]. A symmetry group G is defined as a set X with an operation ., which is associative on the set, and has inverse and identity elements. Furthermore, combination of two symmetry transformations on a set, also creates a symmetry group.

We define the P_r group where $r \in \mathbb{N}$, and the operations consist of all translations, and r discrete 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 is then convolved with the input to produce an output map with r values corresponding to the rotations of the kernels.

Following the group convolutional layer, we apply three 1x1 group convolutional layers (Figure 1), which are efficiently implemented as 3D convolutions. The final layer outputs the parameters of the approximate posterior distributions, $q(t = (i, j), r = r'|y) = \text{softmax}(a(y))_{i,j,r'}$ where $a(y)$ is an rxNxM activation map output by the network given input y, and $q(z|t = (i, j), r = r', y) = \mathcal{N}(\mu_z(y)_{i,j,r'}, \sigma_z(y)^2_{i,j,r'})$ and $q(\theta|t = (i, j), r = r', y) = \mathcal{N}(\mu_\theta(y)_{i,j,r'} + \frac{r'2\pi}{r}, \sigma_\theta(y)^2_{i,j,r'})$ where $\mu_z(y)_{i,j,r'}, \sigma_z(y)^2_{i,j,r'}, \mu_\theta(y)_{i,j,r'}$, and $\sigma_\theta(y)^2_{i,j,r'}$ are output by the inference network for each spatial location and discrete rotation group. $\frac{r'2\pi}{r}$ is the angle of the r'th discrete rotation.

During training, we sample from the joint posterior distribution of $q(t, r|y)$ using Gumbel-Softmax [28] in order to differentiably sample from the approximate posterior to estimate the expected reconstruction error given q. We use the soft one-hot sample to calculate the weighted sum of the parameters of the $q(z|t, r, y)$ and $q(\theta|t, r, y)$ distributions and then sample from those distributions to get a sample from the full joint approximate posterior. We define a coordinate system where x is in the [-1,1] range, and is centered on the middle of the image. The output of the Gumbel-Softmax directly describes a sample from the approximate posterior over the translation, which we get by calculating the weighted sum over the translation grid points.

3.3.2 Spatial decoder

Given t and θ, we translate and rotate the image spatial coordinates, $x' = R(\theta)x + t$, where $R(\theta)$ is the rotation matrix for angle θ. The generator receives the sampled z value along with transformed spatial coordinates, x', and outputs the value for the pixels at those spatial coordinates, forming the image. We use random Fourier feature expansion [29] on the coordinates of the pixel before feeding it through a stack of fully-connected layers (Figure 1). The spatial coordinate and z inputs are processed separately by two parallel sets of fully-connected layers. Those representations are then summed and processed by the remaining shared fully-connected layers which give the final output value.

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 kernels where size of the kernels in the first layer, is set to a value larger than the size of the targeted objects. The reason for this is that we only use one group convolutional layer in our models and we want to assure that its receptive field is at least as large as the objects of interest. We use leaky-ReLU activation functions, and the batch size for all the experiments is set to 100. We use ADAM optimizer [30], with learning rate of 2e-4, and the learning rate is decayed by the 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. We run all our experiments in a single NVIDIA A100 GPU, with 80 GB memory.

Each fully-connected layer in the generator has 512 hidden units and the generator output is a probability value, in the case of the binary valued pixels, or mean and standard deviation for real valued pixels. For RGB images, the generator outputs R, G, and B values for each spatial location.

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

We first examine whether TARGET-VAE can accurately predict rotation and translation of objects despite being trained without supervision.

Datasets We use two variants of the MNIST dataset: 1) (MNIST(N)) MNIST digits randomly rotated by sampling from $\mathcal{N}(0, \frac{\pi^2}{16})$ and randomly translated by sampling from $\mathcal{N}(0, 5^2)$ pixels, and 2) (MNIST(U)) MNIST digits rotated by sampling uniformly from $U(0, 2\pi)$ and translated using the same distribution as in 1) (Appendix Figure 6). In both datasets, the train and test sets have 60,000, and 10,000 images of dimensions 50x50 pixels, respectively.

Training We train TARGET-VAE with P_4, P_8, and P_{16} and z_dim=2 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 $p(\theta_{offset}) \sim p(\theta)$, where θ_{offset} are the rotation angles of the kernels in each dimension of r. For both datasets, we set the prior over t to $\mathcal{N}(0, 5^2)$.

Given the trained model, we perform inference on t and θ on the test set using the inference network. We find the most likely values of t and θ from the approximate posterior distributions given by the network and compare these with the ground truth values for the test set images. We calculate the Pearson correlation between the predicted and true translations and the circular correlation between our predicted and true rotation angles [31]. In a comparison with a spatial-VAE network [6] trained with the same number of parameters as TARGET-VAE and z_dim=2, we find that TARGET-VAE significantly improves over spatial-VAE when predicting rotation. Spatial-VAE fails to give meaningful rotation predictions when digits are uniformly rotated, but TARGET-VAE predicts rotation with remarkable accuracy (Table 1). We also find that increasing the number of discrete rotations in the group convolution improves the accuracy of rotation inference, but does not have a significant impact on translation inference. Interestingly, spatial-VAE performs better than TARGET-VAE on translation inference, though the difference is small and predictions from both correlate strongly

with the ground truth. One possible reason for this is that TARGET-VAE only predicts whole pixel translations, due to the formulation of the approximate posterior, whereas spatial-VAE predicts real valued translations and, therefore, can predict sub-pixel values. In the future, including fine-grained translation inference in TARGET-VAE, similarly to the rotation distribution, could allow TARGET-VAE to resolve sub-pixel translations as well. We also found that increasing the dimension of z did not have a noticeable effect on the ability of the network to predict translation and rotation.

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
TARGET-VAE P_4	0.97, 0.972	0.814	0.975, 0.976	0.80
TARGET-VAE P_8	0.981, 0.981	0.89	0.972, 0.971	0.859
TARGET-VAE P_{16}	0.969, 0.972	**0.898**	0.974, 0.971	**0.93**

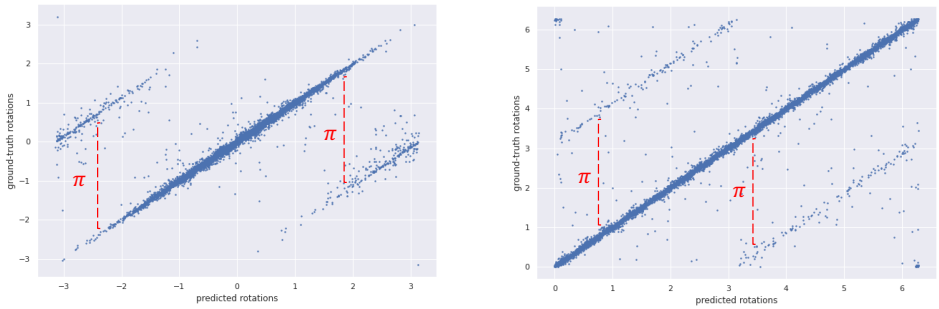

Figure 2: Left: Predicted θ vs. ground-truth data in MNIST(N), Right: Predicted θ vs. ground-truth data in MNIST(N), with TARGET-VAE P_8. There is a π difference between the predicted angles and the ground-truth values of some of the digits (digits 0, 1, and 8), which is related to the rotation symmetry of them.

Some of the digits in MNIST have natural ambiguity in their rotations. Investigating the correlation between the predicted rotation values and the ground truth ones shows a difference about π between some of the predicted angles and the ground truth (Figure 2). This is expected, because several digits have approximate symmetry. For example, 0, 1, and 8 are roughly 2-fold symmetric (Appendix A.3) creating inherent ambiguity in the rotation that is reflected in larger error in the rotation predictions (Table 3). Six and 9 are also similar when rotated by π, although, interestingly, TARGET-VAE is able to distinguish these digits and predict their rotations accurately. Perhaps this is because 9 is typical drawn with a straight spine while 6 is drawn with a curved one. This also suggests that the posterior predictions given by TARGET-VAE could be helpful for identifying the symmetry group of objects.

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 is able to recover semantic organization of objects. Using the z_dim=2 models trained above and additional TARGET-VAE models trained with z_dim=32, we extract the most likely value of z for each model from $q(z|t, r, y)$ for each image in the MNIST(N) and MNIST(U) test sets. To evaluate the disentangled semantic value of these representations, we cluster the test set images in z-space using agglomerative clustering [32]. We then calculate the clustering accuracy by finding the highest accuracy assignment between clusters and digits for each set of representations. We find that semantic representations produced by TARGET-VAE much more closely represent the underlying semantics of the MNIST dataset when compared to representations produced by baseline methods (Table 2). TARGET-VAE achieves clustering accuracies approximately 2-5x higher than standard VAE [3], beta-VAE [4], and spatial-VAE [6] models trained with the same number of parameters as TARGET-VAE. We set the z_dim=2 for both spatial-VAE and TARGET-VAE. Since the standard VAE and the

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**

beta-VAE do not have dedicated latent variables for identifying translation and rotation, we set the dimension of the unstructured latent space to 4 for these models. Also, we use $\beta = 4$ for the beta-VAE based on the settings proposed by Higgins et al. [4]. We believe that direct inference on rotation and translation with the equivariant inference network of TARGET-VAE plus the transformation-equivariant generator are the main reason for the dramatic increase in the clustering accuracy (see ablation studies in Appendix A.7).

We also observe that increasing the dimension of z can significantly increase clustering performance and produces clearly distinct clusters for each digit (Figure 3). Increasing the number of discrete rotations also can improve clustering performance, likely because improving rotation inference means that z and θ are better disentangled, but this trend is not universally true for both MNIST(N) and MNIST(U). We note that the P_8 model with z_dim=32 performs best on MNIST(N), perhaps because rotation inference is performed roughly equally well by the P_8 and P_{16} models on this dataset.

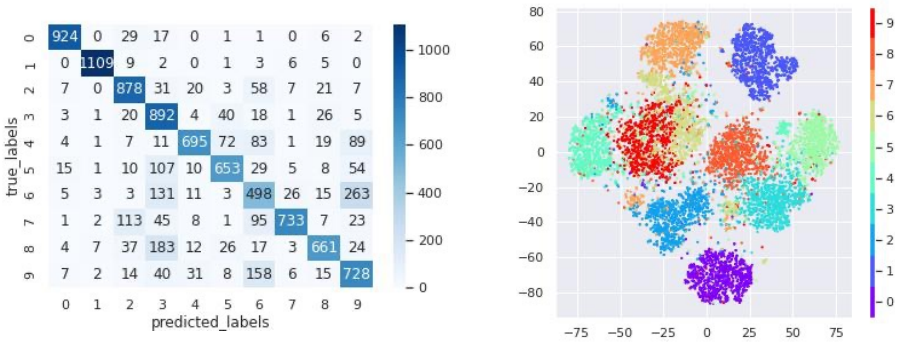

Figure 3: Confusion matrix of trained TARGET-VAE on MNIST(N) with z_dim=32 (left), and the t-sne figure of the 32 dimension latent representation space (right). Based on the confusion matrix, most of the digits are clustered correctly, and the largest misclassification is related to the confusion between rotated digits 6 and 9, which is understandable.

TARGET-VAE is also able to detect multiple objects at inference time (Appendix A.4) and is able to learn meaningfully disentangled semantic representations on other datasets, including dSprites [33] (Appendix A.5) and the galaxy zoo [1] (Appendix A.6).

4.4 Identifying class-averages and protein conformations in the cryo-EM micrographs with TARGET-VAE

In single-particle cryo-EM micrographs, individual biomolecules (particles) are spread throughout images with random orientations and locations. Being 2d projections of 3d structures, particle images have variability due to variation in the orientation of the particles in sample and also have variability due to conformational heterogeneity between structures. To be able to identify the particles and their conformations, semantic representations should be invariant to these transformations. Here,

we demonstrate the capacity of TARGET-VAE to learning different classes and conformations of particles in cryo-EM micrographs.

Identifying different classes of particles in EMPIAR-10025. The dataset EMPIAR-10025 [34] contains T20S proteasome micrographs used to solve the T20S proteasome structure at 2.8 Å resolution. We use the Topaz [35] to create a stack of 161,292 400x400 images of T20S proteasome particles. After downsampling by factor of 4 and normalizing the images, we train a P_8 TARGET-VAE with z_dim=2. Due to the size and variability of these images, we use a larger spatial generator network than the one used on the MNIST datasets. Here, the generator has six fully-connected layers where each layer has 512 hidden units. We use a uniform prior over r. We randomly select 10% of the images as the validation set to monitor the training process. After training, we apply the inference model on the images of the particles to extract the most-probable semantic representations (same as the clustering of the MNIST datasets described in 4.3). We find that these representations produce meaningful semantic clustering of the particles, grouping them by view and the presence of contaminants (gold particles in the dataset), and also can be used to reconstruct the particles with specified rotation and translation (Figure 4).

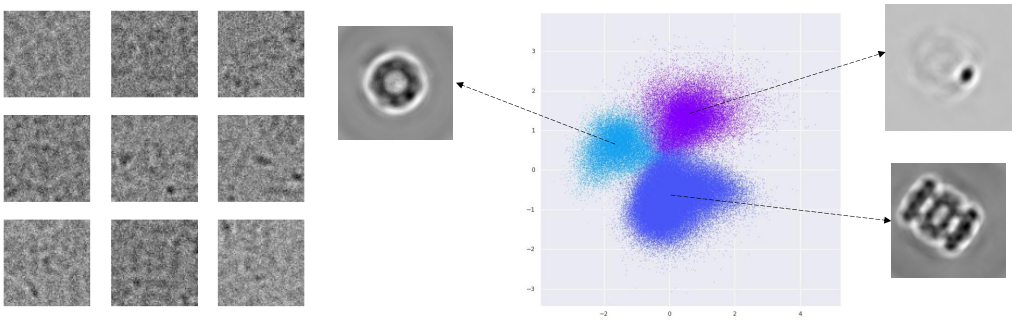

Figure 4: Left: Images of T20S proteasome particles from EMPIAR-10025. These images are extremely noisy. Right: The 2D latent representation space for EMPIAR-10025 learned by TARGET-VAE. TARGET-VAE identifies three clusters from which we show reconstructed examples produced by the spatial generator network.

Identifying different views of particles in EMPIAR-10029. EMPIAR-10029 is a simulated EM dataset of GroEL particles. This dataset has 10,000 200x200 images, which we downsample to 100x100 pixels and normalize. We train the TARGET-VAE with the same settings as for EMPIAR-10025. The latent space does not identify separate clusters in this dataset, likely because a continuum over views is present from simulation, but exploration of the learned semantic latent space identifies GroEL projections independent of their rotation and translation (Figure 5).

TARGET-VAE is also able to discover continuous latent heterogeneity in other cryo-EM datasets (Appendix A.8).

5 Conclusion

We present TARGET-VAE, a translation and rotation group equivariant variational autoencoder framework for learning translation and rotation invariant object representation in images without supervision. By structuring the encoder to be translation and rotation equivariant, the model learns latent rotation and translation variables, disentangling these transformations from object semantics encoded in a separately learned unstructured semantic latent variables, by training it jointly with a spatially equivariant generator network. TARGET-VAE learns to accurately predict object locations and rotations, without any supervision, and 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 equivariances and adapting the approximate posterior 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-

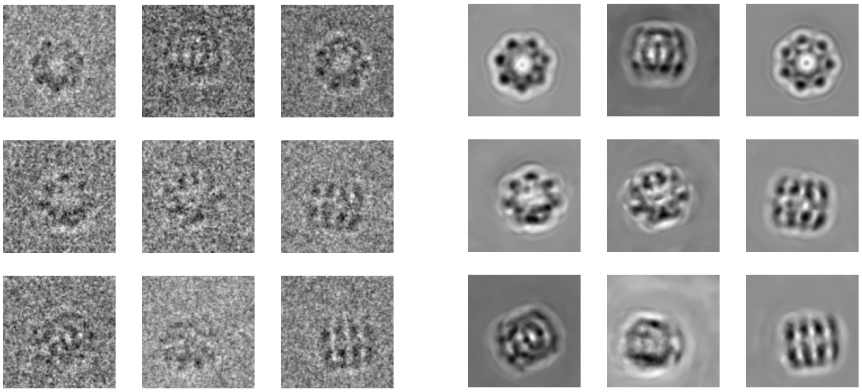

Figure 5: Left: Simulated GroEL particle images from EMPIAR-10029. Right: The reconstructed particles with rotation angle set to zero. TARGET-VAE learns the distribution over particle views controlled for in plane rotation and translation and generates de-noised particle views.

VAE lays the groundwork for a new generation of fully unsupervised object detection and semantic analysis methods for cryo-EM and other imaging modalities.

Acknowledgments and Disclosure of Funding

This work was supported by a grant from the Simons Foundation (SF349247).

References

[1] Chris J Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M Jordan Raddick, Robert C Nichol, Alex Szalay, Dan Andreescu, et al. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, 2008.

[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] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural Information Processing Systems*, 34:24993–25005, 2021.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

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

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

[27] 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.

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

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

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

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

[32] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31(3):274–295, 2014.

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

[34] 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.

[35] 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.

[36] Fu-Yang Lin, Jianghai Zhu, Edward T Eng, Nathan E Hudson, and Timothy A Springer. β-subunit binding is sufficient for ligands to open the integrin αiibβ3 headpiece. *Journal of Biological Chemistry*, 291(9):4537–4546, 2016.

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 of 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 will be available after the double blind review process.

(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 in 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]

A Appendix

A.1 Calculating Kullback-Leibler divergence

Based on the standard definition for the KL-divergence, we have:

$$
\begin{aligned}
KL(q(z,\theta,t,r|y)||p(z,\theta,t,r)) &= \sum_{z,\theta,t,r} q(z,\theta,t,r|y) log \frac{q(z,\theta,t,r|y)}{p(z,\theta,t,r)} \\
&= \sum_{z,\theta,t,r} q(t,r|y)q(\theta|t,r,y)q(z|t,r,y) log \frac{q(t,r|y)q(\theta|t,r,y)q(z|t,r,y)}{p(t,r)p(\theta)p(z)} \\
&= \sum_{t,r} q(t,r|y) log \frac{q(t,r|y)}{p(t,r)} + \\
&\sum_{z,\theta,t,r} q(t,r|y)q(\theta|t,r,y)q(z|t,r,y) log \frac{q(\theta|t,r,y)q(z|t,r,y)}{p(\theta)p(z)}
\end{aligned} \tag{7}
$$

To simplify this equation, we define the first part of the result from Equation 7 as:

$$
KL_{t,r} = \sum_{t,r} q(t,r|y) log \frac{q(t,r|y)}{p(t,r)} \tag{8}
$$

The second part of the result from Equation 7 can be further expanded as:

$$
\begin{aligned}
&\sum_{z,\theta,t,r} q(t,r|y)q(\theta|t,r,y)q(z|t,r,y)log\frac{q(\theta|t,r,y)q(z|t,r,y)}{p(\theta)p(z)} \\
&= \sum_{t,r} q\left(t,r|y\right)\left(\sum_{\theta} q\left(\theta|t,r,y\right) log\frac{q\left(\theta|t,r,y\right)}{p\left(\theta\right)} + \sum_{z} q(z|t,r,y)log\frac{q\left(z|t,r,y\right)}{p\left(z\right)}\right) \\
&= \sum_{t,r} q\left(t,r|y\right)\left(KL\left(q\left(\theta|t,r,y\right)||p\left(\theta\right)\right) + KL\left(q\left(z|t,r,y\right)||p\left(z\right)\right)\right) \qquad (9)
\end{aligned}
$$

Assuming these definitions for KL_θ and KL_z:

$$KL_\theta = KL\left(q\left(\theta|t,r,y\right)||p\left(\theta\right)\right) \qquad (10)$$

$$KL_z = KL\left(q\left(z|t,r,y\right)||p\left(z\right)\right) \qquad (11)$$

, we can rewrite the the Equation 7 using Equations 8, 9, 10, and 11 as:

$$KL(q(z,\theta,t,r|y)||p(z,\theta,t,r)) = KL_{t,r} + \sum_{t,r} q\left(t,r|y\right)\left(KL_\theta + KL_z\right) \qquad (12)$$

A.2 MNIST(N) and MNIST(U) datasets

We generated two datasets of MNIST(N) and MNIST(U), by rotating and translating digits in MNIST. The rotation angles of digits for MNIST(N) are randomly sampled from $\mathcal{N}(0, \frac{\pi^2}{16})$, and for MNIST(U) are randomly sampled from $U(0, 2\pi)$ (Figure 6). Images in both of the datasets are 50x50 pixels.

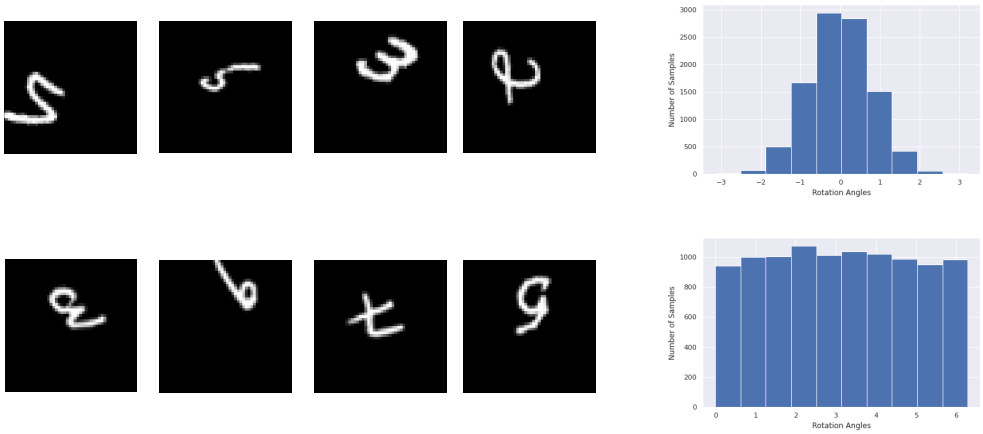

Figure 6: Left: Instances of MNIST(N) (top), and MNIST(U) (bottom) datasets. Right: Distribution of rotation angles in the test set of MNIST(N) (top), and MNIST(U) (bottom) datasets.

A.3 Digit-wise rotation correlation, and RMSE of the predicted rotations

Following our discussion about the difference between the predicted angles and the ground-truth values, we measure the digit-wise rotation correlation for each dataset (Figure 7). We recognized that some predicted rotations are off by about π from their ground-truth angles for digits 0, 1, and 8. We suspect that this is caused by the symmetry of these digits where for example a hand-written digit 8 with rotation π, looks roughly the same as that digit with rotation zero.

To study the accuracy of the predicted rotation angles by TARGET-VAE, we calculate the mean standard deviation of the predicted rotations, introduced in [17]. This metric basically measures the mean square error between the rotation of the object in the input image and the predicted rotation for that object. We rotate each image in the test set of MNIST(U), 160 times using angles uniformly

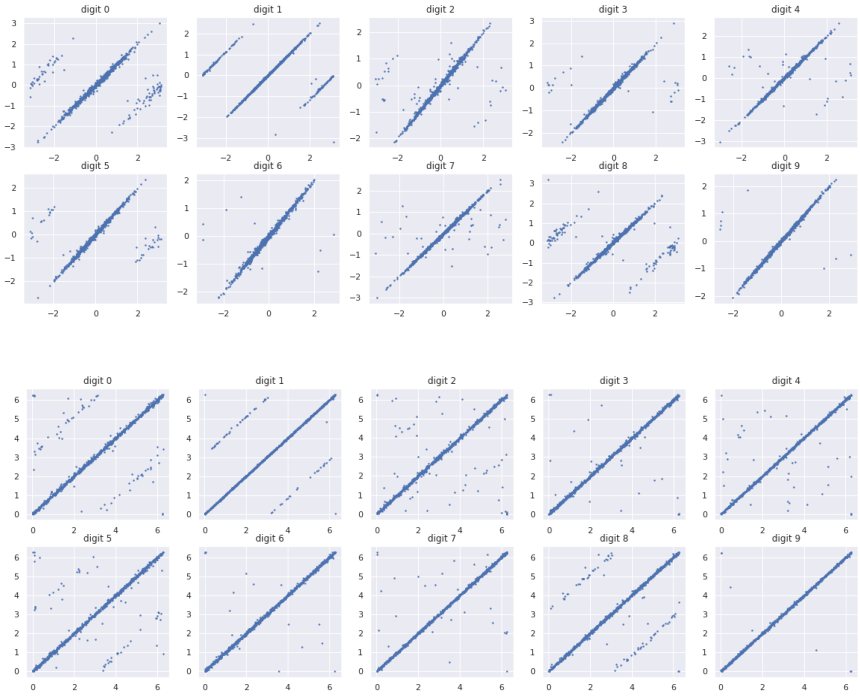

Figure 7: The predicted rotation values (x axis) and the ground-truth rotation angles (y axis) for digits in MNIST(N) (top), and MNIST(U) (bottom) using TARGET-VAE with model P_8 and z_dim=2. Some predicted rotations for digits 0, 1, and 8 are off by π from their ground-truth values.

Table 3: The RMSE of the predicted rotations over MNIST(U) with z-dim=2

Digits	0	1	2	3	4	5	6	7	8.	9	AVG
Spatial-VAE [6]	96.50	98.73	97.83	97.88	98.59	98.06	97.82	98.10	97.62	98.55	97.97
TARGET-VAE P_4	31.57	33.86	32.36	11.66	27.08	20.94	6.26	8.15	28.95	6.68	20.75
TARGET-VAE P_8	32.81	22.04	16.69	6.36	20.71	16.24	13.00	15.84	16.97	7.80	16.85
TARGET-VAE P_{16}	17.39	18.47	14.74	5.79	7.65	12.09	3.40	4.63	17.75	2.74	10.47

sampled from [0, 2π], and then pass them through the trained inference model to get the predicted rotation angle for them. Since assigning predicted rotation of zero to an object is arbitrary in our framework (model might assign predicted rotation angle of zero to the object that is actually rotated 90 degrees), we subtract the predicted rotation for the objects from the predicted rotation value when the input object is not rotated. Table 3, shows the root mean square error (RMSE) calculated in average for each digit's predicted rotation. As expected, the RMSE of rotations is highest when there is no inference done on rotations as in spatial-VAE [6], and using finer discretization in rotation inference results in more accurate prediction of the rotation values. Some digits such as 0, 1, and 8, due to their symmetry, have less accurate rotation predictions compared to the other ones.

A.4 TARGET-VAE identifies multiple objects without supervision

We created a new dataset using multiple rotated and translated digits from MNIST(U). We call this new dataset MNIST(multi) which we created by randomly sampling rotated and translated digits from MNIST(U) and inserting them in random offsets of a 150x150 pixel image. We use the model trained on MNIST(U) to identify the translation, rotation, and the content latent for the digits in MNIST(multi). We find that the model correctly identifies and reconstructs the objects (Figure 8). Even though the model, can identify non-overlapping objects in this approach, it struggles to identify the overlapping objects. We believe these preliminary results show the potential of our model to be used in multiple objects detection.

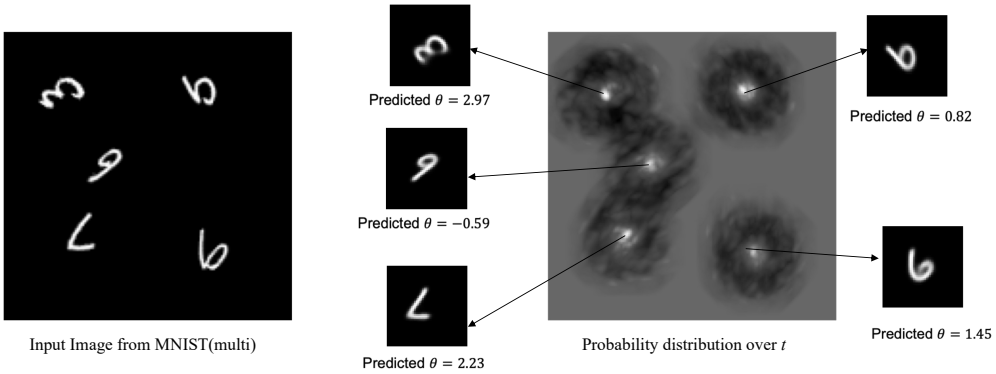

Figure 8: Left: Input image with multiple rotated and translated digits from MNIST(multi); Right: Probability distribution over t ($q(t|y)$), the reconstructed objects and their predicted rotation angles. We marginalized $q(t, r|y)$ over r to obtain $q(t|y)$ for visualization purposes. The high probability values in this attention map show the predicted locations of the objects. We use the peaks in $q(t, r|y)$ to sample from $q(\theta|t, r, y)$ and $q(z|t, r, y)$, the predicted rotation and content values for each object. The sampled θ, t, and z values are used to reconstruct each individual object.

A.5 TARGET-VAE predicts the rotation of the shapes in dSprites and clusters them with high accuracy

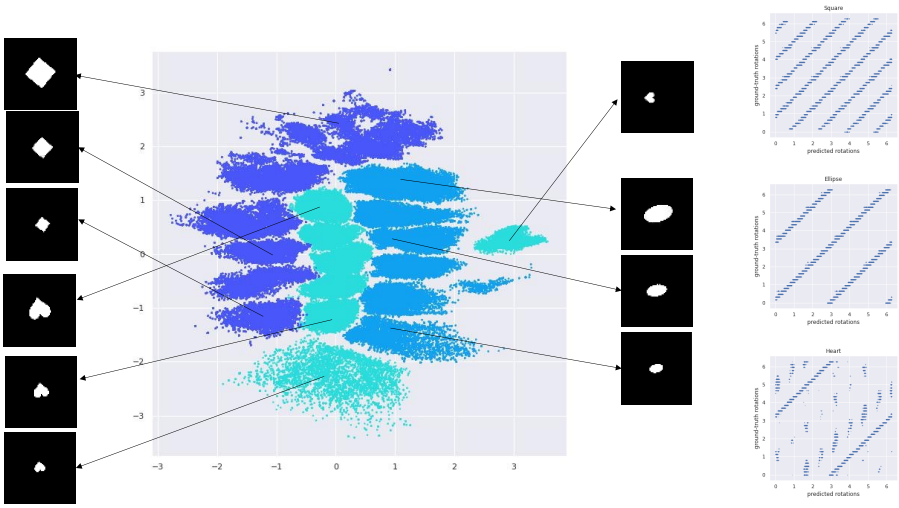

Figure 9: Left: Two-dimension latent space of dSprites using TARGET-VAE with P_8. Right: Correlation between rotations given by TARGET-VAE and ground truth rotations for squares, ellipses, and hearts.

The dSprites dataset was introduced to benchmark unsupervised disentangled representation learning methods [33]. It contains 64x64 images of three shapes: squares, ellipses, and hearts. Each shape is rotated by one of 40 values linearly spaced in [0, 2π], translated across both x and y dimensions, and scaled using one of six linearly spaced values in [0.5, 1]. We train TARGET-VAE with P_8 group convolution and z_dim=2 on dSprites, using a uniform prior over r and $\mathcal{N}(0, \pi)$ prior over θ. TARGET-VAE learns content representations that capture the type and scale of the objects, but is invariant from their location and rotation (Figure 9). Furthermore, TARGET-VAE accurately predicts the rotation of the shapes, and, interestingly, the symmetry groups of the shapes become immediately apparent when examining the correlation between the predicted and ground truth rotations.

A.6 TARGET-VAE learns transformation-invariant representations in the galaxy zoo dataset

Galaxy zoo contains images of galaxies gathered by Sloan Digital Sky Survey [1]. This dataset contains more than 61,000 RGB images, which we cropped and downsampled to 64x64 pixels. The galaxies appear with different rotations and translations in the images. We train TARGET-VAE with P_8 group convolution on the train set. TARGET-VAE learns to accurately predict the rotation and translation of the galaxies and reconstructs centered and aligned galaxy images when generating with t and θ set to zero (Figure 10).

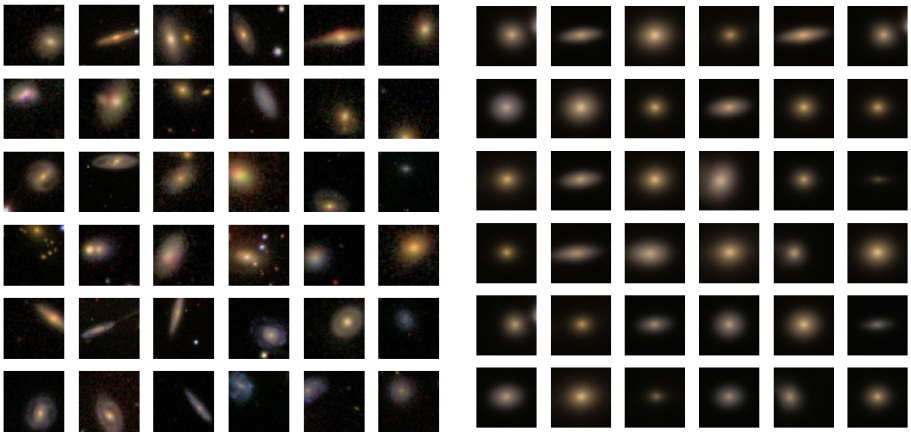

Figure 10: Left: Images of galaxy dataset used for testing. Right: Reconstructed images by TARGET-VAE with P_8 and z_dim=2 and using the transformation-invariant representations, where rotation and translation are set to 0.

A.7 Ablation studies

We conducted extensive ablation study to validate the effectiveness of the main components of our proposed method.

Variant 1 - Inference only on translation We evaluated the importance of the rotation equivariant features, by modifying the inference model to use regular convolutional layers instead of the group convolutional ones. As a result, the posterior distributions will be $q(t|y)$, $q(z|t, y)$, and $q(\theta|t, y)$, where they only depend on the input y and the translation value t. We observed that, as expected, eliminating inference on the discretized rotation dimension has a significant negative effect on identifying transformation-invariant representations and the clustering accuracy on MNIST(U) is only 33.8% (Table 4).

Variant 2 - Inference only on translation + using group convolutional layers In this variant, we use the group convolutional layers in the inference model, but we do not perform any inference on the rotation dimension. We apply a fully-connected layer on the output of the group convolutional layers to map the rotation values at every location to a single value. As a result, the posterior distribution are the same as variant 1 and they do not depend on r. We experiment with this variant to identify how effective is the use of the group convolution layers. It turns out that if we do not perform any inference on the rotation dimension, just using the group convolutional layers for feature extraction, only slightly improves the clustering accuracy.

Variant 3 - Inference on both translation and rotation without adding θ_{offset}: In this variant, we use group convolutional layers and we perform inference on both translation and rotation. The difference between this variant and our proposed framework is that we are not adding θ_{offset} of kernels in each rotation dimension to the $q(\theta|t, r, y)$. Adding θ_{offset} to the posterior on θ, allows us to break down the rotation space among the r discretized rotations, and without it, the model proves not capable of identifying the rotation of the digits in the MNIST(U) dataset.

Table 4: Performace of variants in the ablation study on MNIST(U)

Model	Group Conv	Translation Corr	Rotation Corr	Clustering Accuracy
Variant 1	-	0.966, 0.967	0.005	33.8%
Variant 2	p4	0.967, 0.967	0.005	37.1%
Variant 3	p4	0.968, 0.972	0.008	36.6%
Variant 4	p4	0.975, 0.976	0.80	56.6%
Variant 5	p8	0.972, 0.971	0.859	57.1%
Variant 6	p16	0.974, 0.971	0.93	63.4%

Variants 4 to 6 - Increasing the level of discretization of the rotation space In these variants, we perform inference on both rotation and translation, and we add θ_{offset} of each rotation dimension to the mean of its corresponding q(θ|t,r,y) distribution. In these variants, we show that by increasing the level of discretization of the rotation space, model can have a better estimate of the actual rotation values, and this in turn helps the model with improving the clustering accuracy.

Table 4, shows the translation and rotation correlation, along with the clustering accuracy of the mentioned variants on the MNIST(U) dataset. Performing inference on rotation plus adding θ_{offset} to the posterior on θ, significantly increases the rotation correlation and the clustering accuracy.

A.8 Learning translation-rotation-invariant representations of proteins with TARGET-VAE

In cryo-EM images, rotation and translation are the major transformations that cause variations in particles in the micrograph. Here, we show the result of our experiments with TARGET-VAE to learn the translation-rotation-invariant representations, for two cryo-EM datasets.

Identifying hinge motion of 5HDB We train our model on a dataset of 20,000 simulated projections of integrin α-IIb in complex with integrin β-3 (5HDB) [36]. We aim to identify the translation and rotation invariant representations of the protein to be able to identify the variations in its structure. We train TARGET-VAE with P_8 and uniform prior over θ. Since there is less variation in the data, we set z_dim to 1. After training, we sample from the representation latent space and reconstruct the images with no rotation and translation. Figure 11 shows some examples of the reconstructed particles. We observe that the reconstructed images identify the hinge motion of the particle.

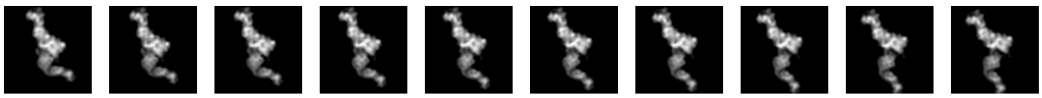

Figure 11: Reconstructed proteins from 5HDB dataset by sampling from 1D representation space, shows the movement in the lower part of the particle.

Learning arm motion of particle in CODH/ACS We have about 14,000 40x40 pixels images of the CODH/ACS protein complex. We train TARGET-VAE with P_8 and $z_dim = 2$, for 100 epochs. The prior over θ is uniform and we set the generator to have 6 fully-connected layers with 512 hidden units in each. After training, we sample from the transformation-invariant representation space and reconstruct the particle to identify the different conformations of the protein. Figure 12 shows the movement on the upper and lower parts of the particle.

Figure 12: Reconstructed proteins from CODH/ACS dataset, where the motion in the upper and lower arms of the particle can be captured.