Disentangling Factors of Variation with Cycle-Consistent Variational Auto-Encoders

Ananya Harsh Jha[1], Saket Anand[1], Maneesh Singh[2], and VSR Veeravasarapu[2]

[1] IIIT-Delhi
{ananyaharsh12018, anands}@iiitd.ac.in
[2] Verisk Analytics
maneesh.singh@verisk.com, vsr.veera@gmail.com

Abstract. Generative models that learn disentangled representations for different factors of variation in an image can be very useful for targeted data augmentation. By sampling from the disentangled latent subspace of interest, we can efficiently generate new data necessary for a particular task. Learning disentangled representations is a challenging problem, especially when certain factors of variation are difficult to label. In this paper, we introduce a novel architecture that disentangles the latent space into two complementary subspaces by using only weak supervision in form of pairwise similarity labels. Inspired by the recent success of cycle-consistent adversarial architectures, we use cycle-consistency in a variational auto-encoder framework. Our non-adversarial approach is in contrast with the recent works that combine adversarial training with auto-encoders to disentangle representations. We show compelling results of disentangled latent subspaces on three datasets and compare with recent works that leverage adversarial training.

Keywords: Disentangling Factors of Variation, Cycle-Consistent Architecture, Variational Auto-encoders

1 Introduction

Natural images can be thought of as samples from an unknown distribution conditioned on different factors of variation. The appearance of objects in an image is influenced by these factors that may correspond to shape, geometric attributes, illumination, texture and pose. Based on the task at hand, like image classification, many of these factors serve as a distraction for the prediction model and are often referred to as nuisance variables. One way to mitigate the confusion caused by uninformative factors of variation is to design representations that ignore all nuisance variables [1, 2]. This approach, however, is limited by the quantity and quality of training data available. Another way is to train a classifier to learn representations, invariant to uninformative factors of variation, by providing sufficient diversity via data augmentation [3].

Code for the paper: github.com/ananyahjha93/cycle-consistent-vae

Generative models that are driven by a disentangled latent space can be an efficient way of controlled data augmentation. Although Generative Adversarial Networks (GANs) [4, 5] have proven to be excellent at generating new data samples, vanilla GAN architecture does not support inference over latent variables. This prevents control over different factors of variation during data generation. DNA-GANs [6] introduce a fully supervised architecture to disentangle factors of variation, however, acquiring labels for each factor, even when possible, is cumbersome and time consuming.

Recent works [7, 8] combine auto-encoders with adversarial training to disentangle informative and uninformative factors of variation and map them onto separate sets of latent variables. The informative factors, typically specified by the task of interest, are associated with the available source of supervision, e.g. class identity or pose, and are referred to as the *specified* factors of variation. The remaining uninformative factors are grouped together as *unspecified* factors of variation. Learning such a model has two benefits: first, the encoder learns to factor out nuisance variables for the task under consideration, and second, the decoder can be used as a generative model that can generate novel samples with controlled specified and randomized unspecified factors of variation.

In context of disentangled latent representations, Mathieu et al. [7] define *degenerate solution* as a failure case, where the specified latent variables are entirely ignored by the decoder and all information (including image identity) is taken from the unspecified latent variables during image generation (Fig. 1 (c) and (d)). This degeneracy is expected in auto-encoders unless the latent space is somehow constrained to preserve information about the specified and unspecified factors in the corresponding subspaces. Both [7] and [8] circumvent this issue by using an adversarial loss that trains their auto-encoder to produce images whose identity is defined by the specified latent variables instead of the unspecified latent variables. While this strategy produces good quality novel images, it may train the decoder to *ignore any leakage of information* across the specified and unspecified latent spaces, rather than training the encoder to restrict this leakage.

Szabó et al. [8] have also explored a non-adversarial approach to disentangle factors of variation. They demonstrate that severely restricting the dimensionality of the unspecified latent space discourages the encoder from encoding information related to the specified factors of variation in it. However, the results of this architecture are extremely sensitive to the dimensionality of the unspecified space. As shown in Fig. 1 (e), even slightly plausible results require careful selection of dimensionality.
Based on these observations, we make the following contributions in this work:

- We introduce *cycle-consistent variational auto-encoders*, a weakly supervised generative model, that disentangles specified and unspecified factors of variation using only pairwise similarity labels
- We empirically show that our proposed architecture avoids *degeneracy* and is robust to the choices of dimensionality of both the specified and unspecified latent subspaces
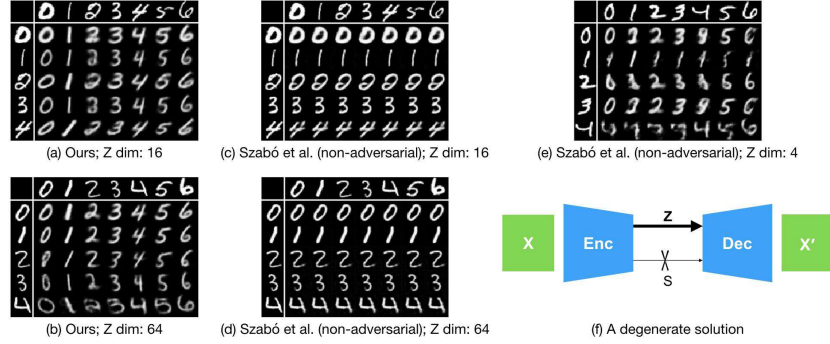
(a) Ours; Z dim: 16   (c) Szabó et al. (non-adversarial); Z dim: 16   (e) Szabó et al. (non-adversarial); Z dim: 4

(b) Ours; Z dim: 64   (d) Szabó et al. (non-adversarial); Z dim: 64   (f) A degenerate solution

Fig. 1. s: **specified factors space (class identity),** z: **unspecified factors space.** In each of the image grids: (a), (b), (c), (d) and (e), the digits in the top row and the first column are taken from the test set. Digits within each grid are generated by taking s from the top row and z from the first column. (a) and (b): results of disentangling factors of variation using our method. (c) and (d): results of the non-adversarial architecture from [8]. (e): dimensionality of z required to produce even a few plausible digits using the non-adversarial approach in [8]. (f): visualization of a degenerate solution in case of auto-encoders.

– We claim and empirically verify that cycle-consistent VAEs produce highly disentangled latent representations by explicitly training the encoder to reduce leakage of specified factors of variation into the unspecified subspace

To our knowledge, cycle-consistency has neither been applied to the problem of disentangling factors of variation nor has been used in combination with variational auto-encoders. The remaining paper is organized as follows: Sec. 2 discusses the previous works relevant in context of this paper, Sec. 3 provides the details of our proposed architecture, Sec. 4 empirically verifies each of our claims using quantitative and qualitative experiments, and Sec. 5 concludes this paper by summarizing our work and providing a scope for further development of the ideas presented.

2   Related Work

Variational Auto-Encoders. Kingma et al. [9] present a variational inference approach for an auto-encoder based latent factor model. Let $X = \{x_i\}_{i=1}^{N}$ be a dataset containing N i.i.d samples, each associated with a continuous latent variable z_i drawn from some prior $p(z)$, usually having a simple parametric form. The approximate posterior $q_\phi(z|x)$ is parameterized using the encoder, while the likelihood term $p_\theta(x|z)$ is parameterized by the decoder. The architecture, popularly known as Variational Auto-Encoders (VAEs), optimizes the following variational lower-bound:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x)\|p(z)) \tag{1}$$

The first term in the RHS is the expected value of the data likelihood, while the second term, the KL divergence, acts as a regularizer for the encoder to align the approximate posterior with the prior distribution of the latent variables. By employing a clever linear transformation based reparameterization, the authors enable end-to-end training of the VAE using back-propagation. At test time, VAEs can be used as a generative model by sampling from the prior $p(z)$ followed by a forward pass through the decoder. Our architecture uses the VAE framework to model the unspecified latent subspace.

Generative Adversarial Networks. GANs [4] have been shown to model complex, high dimensional data distributions and generate novel samples from it. They comprise of two neural networks, a generator and a discriminator, that are trained together in a min-max game setting, by optimizing the loss in Eq. (2). The discriminator outputs the probability that a given sample belongs to true data distribution as opposed to being a sample from the generator. The generator tries to map random samples from a simple parametric prior distribution in the latent space to samples from the true distribution. The generator is said to be successfully trained when the output of the discriminator is $\frac{1}{2}$ for all generated samples. DCGANs [5] use CNNs to replicate complex image distributions and are an excellent example of the success of adversarial training.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (2)$$

Despite their ability to generate high quality samples when successfully trained, GANs require carefully designed tricks to stabilize training and avoid issues like mode collapse. We do not use adversarial training in our proposed approach, however, recent works of Mathieu et al. [7] and Szabó et al. [8] have shown interesting application of adversarial training for disentangling latent factors.

Cycle-Consistency. Cycle-consistency has been used to enable a Neural Machine Translation system to learn from unlabeled data by following a closed loop of machine translation [10]. Zhou et al. [11] use cycle-consistency to establish cross-instance correspondences between pairs of images depicting objects of the same category. Cycle-consistent architectures further find applications in depth estimation [12], unpaired image-to-image translation [13] and unsupervised domain adaptation [14]. We leverage the idea of cycle-consistency in the unspecified latent space and explicitly train the encoder to reduce leakage of information associated with specified factors of variation.

Disentangling Factors of Variation. Initial works like [15] utilize the E-M framework to discover independent factors of variation which describe the observed data. Tenenbaum et al. [16] learn bilinear maps from style and content parameters to images. More recently, [17–19] use Restricted Boltzmann Machines to separately map factors of variation in images. Kulkarni et al. [20] model *vision as inverse graphics* problem by proposing a network that disentangles transformation and lighting variations. In [1] and [2], invariant representations are learnt by factoring out the nuisance variables for a given task at hand.

Tran et al. [21] utilize identity and pose labels to disentangle facial identity from pose by using a modified GAN architecture. SD-GANs [22] introduce a

siamese network architecture over DC-GANs [5] and BE-GANs [23], that simultaneously generates pairs of images with a common identity but different unspecified factors of variation. However, like vanilla GANs they lack any method for inference over the latent variables. Reed et al. [24] develop a novel architecture for visual analogy making, which transforms a query image according to the relationship between the images of an example pair.

DNA-GANs [6] present a fully supervised approach to learn disentangled representations. Adversarial auto-encoders [25] use a semi-supervised approach to disentangle style and class representations, however, unlike the methods of [7], [8] and ours, they cannot generalize to unseen object identities. Hu et al. [26] present an interesting approach that combines auto-encoders with adversarial training to disentangle factors of variation in a fully unsupervised manner. However, the quality of disentanglement still falls short in comparison to [7, 8].

Our work builds upon the network architectures introduced by Mathieu et al. [7] and Szabó et al. [8]. Both of them combine auto-encoders with adversarial training to disentangle specified and unspecified factors of variation based on a single source of supervision, like class labels. Our work differs from these two by introducing a non-adversarial approach to disentangle factors of variation under a weaker source of supervision which uses only pairwise similarity labels. Recently, [27] also proposed another non-adversarial approach for disentangling representations based on group-level supervision. However, their architecture does not explicitly train the encoder, thus making it susceptible to degeneracy when randomly selecting latent dimensions.

3   Cycle-Consistent Variational Auto-Encoders

In this section, we describe our model architecture, explain all its components and develop its training strategy.

3.1   Cycle-Consistency

The intuition behind a cycle-consistent framework is simple – the forward and reverse transformations composited together in any order should approximate an identity function. For the forward cycle, this translates to a forward transform $F(x_i)$ followed by a reverse transform $G(F(x_i)) = x_i'$, such that $x_i' \simeq x_i$.
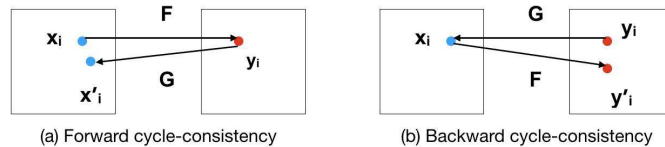

(a) Forward cycle-consistency   (b) Backward cycle-consistency

Fig. 2. (a): Forward cycle in a cycle-consistent framework: $x_i \rightarrow F(x_i) \rightarrow G(F(x_i)) \rightarrow x_i'$. (b): Backward cycle in a cycle-consistent framework: $y_i \rightarrow G(y_i) \rightarrow F(G(y_i)) \rightarrow y_i'$.

The reverse cycle should ensure that a reverse transform followed by a forward transform yields $F(G(y_i)) = y'_i \simeq y_i$. The mappings $F(\cdot)$ and $G(\cdot)$ can be implemented using neural networks with training done by minimizing the ℓ_p norm based *cyclic* loss defined in Eq. (3).

Cycle-consistency naturally fits into the (variational) auto-encoder training framework, where the KL divergence regularized reconstruction comprises the $\mathcal{L}_{forward}$. We also use the reverse cycle-consistency loss to train the encoder to disentangle better. As is typical for such loss functions, we train our model by alternating between the forward and reverse losses. We discuss the details in the sections that follow.

$$\mathcal{L}_{cyclic} = \mathcal{L}_{forward} + \mathcal{L}_{reverse}$$
$$\mathcal{L}_{cyclic} = \mathbb{E}_{x \sim p(x)}[||G(F(x)) - x||_p] + \mathbb{E}_{y \sim p(y)}[||F(G(y)) - y||_p] \quad (3)$$

3.2   Model Description

We propose a conditional variational auto-encoder based model, where the latent space is partitioned into two *complementary* subspaces: s, which controls specified factors of variation associated with the available supervision in the dataset, and z, which models the remaining unspecified factors of variation. Similar to Mathieu et al.'s [7] work we keep s as a real valued vector space and z is assumed to have a standard normal prior distribution $p(z) = \mathcal{N}(0, I)$. Such an architecture enables explicit control in the specified subspace, while permitting random sampling from the unspecified subspace. We assume marginal independence between z and s, which implies complete disentanglement between the factors of variation associated with the two latent subspaces.

Encoder. The encoder can be written as a mapping $Enc(x) = (f_z(x), f_s(x))$, where $f_z(x) = (\mu, \sigma) = z$ and $f_s(x) = s$. Function $f_s(x)$ is a standard encoder with real valued vector latent space and $f_z(x)$ is an encoder whose vector outputs parameterize the approximate posterior $q_\phi(z|x)$. Since the same set of features extracted from x be used to create mappings to z and s, we define a single encoder with shared weights for all but the last layer, which branches out to give outputs of the two functions $f_z(x)$ and $f_s(x)$.

Decoder. The decoder, $x' = Dec(z, s)$, in this VAE is represented by the conditional likelihood $p_\theta(x|z, s)$. Maximizing the expectation of this likelihood w.r.t the approximate posterior and s is equivalent to minimizing the squared reconstruction error.

Forward cycle. We sample a pair of images, x_1 and x_2, from the dataset that have the same class label. We pass both of them through the encoder to generate the corresponding latent representations $Enc(x_1) = (z_1, s_1)$ and $Enc(x_2) = (z_2, s_2)$. The input to the decoder is constructed by swapping the specified latent variables of the two images. This produces the following reconstructions: $x'_1 = Dec(z_1, s_2)$ and $x'_2 = Dec(z_2, s_1)$. Since both these images share class labels, swapping the specified latent variables should have no effect on the reconstruction loss function. We can re-write the conditional likelihood of the decoder as $p_\theta(x|z, s^*)$, where $s^* = f_s(x^*)$ and x^* is any image with the same
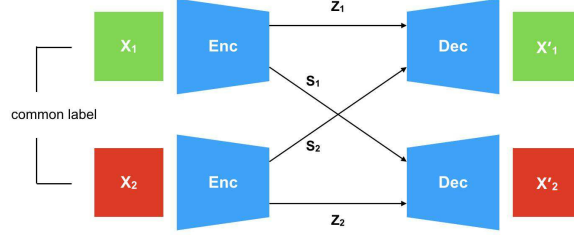
Fig. 3. Image reconstruction using VAEs by swapping the s latent variable between two images from the same class. This process works with pairwise similarity labels, as we do not need to know the actual class label of the sampled image pair.

class label as x. The entire forward cycle minimizes the modified variational upper-bound given in Eq. 4. Fig. 3 shows a diagrammatic representation of the forward cycle.

$$\min_{Enc,\ Dec} \mathcal{L}_{forward} = -\mathbb{E}_{q_\phi(z|x,s^*)}[\log p_\theta(x|z,s^*)] + \mathrm{KL}(q_\phi(z|x,s^*)\|p(z)) \qquad (4)$$

It is worth noting that forward cycle does not demand actual class labels at any given time. This results in the requirement of a weaker form of supervision in which images need to be annotated with pairwise similarity labels. This is in contrast with the previous works of Mathieu et al. [7], which requires actual class labels, and Szabó et al. [8], which requires image triplets.

The forward cycle mentioned above is similar to the auto-encoder reconstruction loss presented in [7] and [8]. As discussed in Sec. 1, the forward cycle alone can produce a *degenerate solution* (Fig. 1 (c) and (d)) as there is no constraint which prevents the decoder from reconstructing images using only the unspecified latent variables. In [7] and [8], an adversarial loss function has been successfully applied to specifically tackle the *degenerate solution*. The resulting generative model works well, however, adversarial training is challenging in general and has limitations in effectively disentangling the latent space. For now, we defer this discussion to Sec. 4.1. In the next section, we introduce our non-adversarial method, based on reverse cycle-consistency, to avoid learning a *degenerate solution* and explicitly train the encoder to prevent information associated with specified factors from leaking into the unspecified subspace.

3.3   Preventing a Degenerate Solution

Reverse cycle. The reverse cycle shown in Fig. 4 is based on the idea of cyclic-consistency in the unspecified latent space. We sample a point z_i from the Gaussian prior $p(z) = \mathcal{N}(0, I)$ over the unspecified latent space and pass it through the decoder in combination with specified latent variables $s_1 = f_s(x_1)$ and $s_2 = f_s(x_2)$ to obtain reconstructions $x_1'' = Dec(z_i, s_1)$ and $x_2'' = Dec(z_i, s_2)$ respectively. Unlike the forward cycle, x_1 and x_2 need not have the same label and can be sampled independently. Since both images x_1'' and x_2'' are generated using
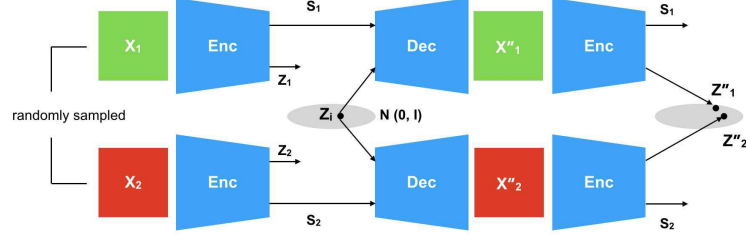
Fig. 4. Reverse cycle of the cycle-consistent VAE architecture. A point sampled from the z latent space, combined with specified factors from two separate sources, forms two different images. However, we should be able to obtain the same sampled point in the z space if we pass the two generated images back through the encoder.

the same z_i, their corresponding unspecified latent embeddings $z_1'' = f_z(x_1'')$ and $z_2'' = f_z(x_2'')$ should be mapped close to each other, regardless of their specified factors. Such a constraint promotes marginal independence of z from s as images generated using different specified factors could potentially be mapped to the same point in the unspecified latent subspace. This step directly drives the encoder to produce disentangled representations by only retaining information related to the unspecified factors in the z latent space.

The variational loss in Eq. (4) enables sampling of the unspecified latent variables and aids the generation of novel images. However, the encoder does not necessarily learn a unique mapping from the image space to the unspecified latent space. In other words, samples with similar unspecified factors are likely to get mapped to significantly different unspecified latent variables. This observation motivates our *pairwise* reverse cycle loss in Eq. (5), which penalizes the encoder if the unspecified latent embeddings z_1'' and z_2'' have a large pairwise distance, but not if they are mapped farther away from the originally sampled point z_i. This modification is in contrast with the typical usage of cycle-consistency in previous works. We found that minimizing the pairwise reverse cycle loss in Eq. (5) was easier than its absolute counterpart $(||z_i - z_1''|| + ||z_i - z_2''||)$, both in terms of the loss value and the extent of disentanglement.

$$\min_{Enc} \mathcal{L}_{reverse} = \mathbb{E}_{x_1,x_2 \sim p(x), z_i \sim \mathcal{N}(0,I)}[||f_z(Dec(z_i, f_s(x_1))) \\ -f_z(Dec(z_i, f_s(x_2)))||_1] \tag{5}$$

4   Experiments

We evaluate the performance of our model on three datasets: MNIST [28], 2D Sprites [24, 29] and LineMod [30, 31]. We divide our experiments into two parts. The first part evaluates the performance of our model in terms of the quality of disentangled representations. The second part evaluates the image generation capabilities of our model. We compare our results with the recent works in [7, 8]. The three dataset we use are described below:

MNIST. The MNIST dataset [28] consists of hand-written digits distributed amongst 10 classes. The specified factors in case of MNIST is the digit identity, while the unspecified factors control digit slant, stroke width etc.

2D Sprites. 2D Sprites consists of game characters (sprites) animated is different poses for use in small scale indie game development. We download the dataset from [29], which consists of 480 unique characters according to variation in gender, hair type, body type, armor type, arm type and greaves type. Each unique character is associated with 298 different poses, 120 of which have weapons and the remaining do not. In total, we have 143040 images in the dataset. The training, validation and the test set contain 320, 80 and 80 unique characters respectively. This implies that character identity in each of the training, validation and test split is mutually exclusive and the dataset presents an opportunity to test our model on completely unseen object identities. The specified factors latent space for 2D Sprites is associated with the character identity, while the pose is associated with the unspecified factors.

Line-MOD. LineMod [30] is an object recognition and 3D pose estimation dataset with 15 unique objects: 'ape', 'benchviseblue', 'bowl', 'cam', 'can', 'cat', 'cup', 'driller', 'duck', 'eggbox', 'glue', 'holepuncher', 'iron', 'lamp' and 'phone', photographed in a highly cluttered environment. We use the synthetic version of the dataset [31], which has the same objects rendered under different viewpoints.

Architecture	z dim	s dim	z train acc.	z test acc.	s train acc.	s test acc.
MNIST						
Szabó et al.	16	16	97.65	96.08	98.89	98.46
Mathieu et al.	16	16	70.85	66.83	99.37	98.52
Ours	16	16	**17.72**	**17.56**	99.72	98.35
Szabó et al.	64	64	99.69	98.14	99.41	98.05
Mathieu et al.	64	64	74.94	72.20	99.94	98.64
Ours	64	64	**26.04**	**26.55**	99.95	98.33
2D Sprites						
Szabó et al.	512	64	99.72	99.63	99.85	99.79
Mathieu et al.	512	64	12.05	11.98	99.18	96.75
Ours	512	64	**11.55**	**11.47**	98.53	97.16
Szabó et al.	1024	512	99.79	99.65	99.87	99.76
Mathieu et al.	1024	512	12.48	12.25	99.22	97.45
Ours	1024	512	**11.27**	**11.61**	98.13	97.22
LineMod						
Szabó et al.	64	256	100.0	100.0	100.0	100.0
Mathieu et al.	64	256	90.14	89.17	100.0	100.0
Ours	64	256	**62.11**	**57.17**	99.99	99.86
Szabó et al.	256	512	100.0	99.97	100.0	100.0
Mathieu et al.	256	512	86.87	86.46	100.0	100.0
Ours	256	512	**60.34**	**57.70**	100.0	100.0

Table 1. Quantitative results for the three datasets. Classification accuracies on the z and s latent spaces are a good indicator of the amount of specified factor information present in them. Since we are aiming for disentangled representations for unspecified and specified factors of variation, *lower is better* for the z latent space and *higher is better* the s latent space.

There are 1541 images per category and we use a split of 1000 images for training, 241 for validation and 300 for test. The specified factors latent space models the object identity in this dataset. The unspecified factors latent space models the remaining factors of variation in the dataset.

During forward cycle, we randomly pick image pairs defined by the same specified factors of variation. During reverse cycle, the selection of images is completely random. All our models were implemented using PyTorch [32]. Specific details about our architectures are included in the supplementary material.

4.1   Quality of Disentangled Representations

We set up the quantitative evaluation experiments similar to [7]. We train a two layer neural network classifier separately on the specified and unspecified latent embeddings generated by each competing model. Since the specified factors of variation are associated with the available labels in each dataset, the classifier accuracy gives a fair measure of the information related to specified factors of variation present in the two latent subspaces. If the factors were completely disentangled, we expect the classification accuracy in the specified latent space to be perfect, while that in the unspecified latent space to be close to chance. In this experiment, we also investigate the effect of change in the dimensionality of the latent spaces. We report the quantitative comparisons in Table 1.

The quantitative results in Table 1 show consistent trends for our proposed Cycle-Consistent VAE architecture across all the three datasets as well as for different dimensionality of the latent spaces. Classification accuracy in the unspecified latent subspace is the smallest for the proposed architecture, while it is comparable with the others in the specified latent subspace. These trends indicate that among the three competing models, the proposed one leaks the least amount of specified factor information into the unspecified latent subspace. This restricted amount of leakage of specified information can be attributed to the reverse cycle-consistency loss that explicitly trains the encoder to disentangle factors more effectively.
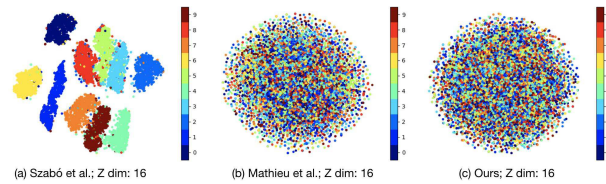

(a) Szabó et al.; Z dim: 16   (b) Mathieu et al.; Z dim: 16   (c) Ours; Z dim: 16

Fig. 5. Comparison between t-SNE plots of the z latent space for MNIST. We can see good cluster formation according to class identities in (a) [8], indicating that adversarial training alone does not promote marginal independence of z from s. Mathieu's work [7] in (b) uses re-parameterization on the encoder output to create confusion regarding the specified factors in the z space while retaining information related to the unspecified factors. Our work (c) combines re-parameterization with reverse cycle loss to create confusion regarding the specified factors.

We also visualize the unspecified latent space as t-SNE plots [33] to check for the presence of any apparent structure based on the available labels with the MNIST dataset. Fig. 5 shows the t-SNE plots of the unspecified latent space obtained by each of the competing models. The points are color-coded to indicate specified factor labels, which in case of MNIST are the digit identities. We can see clear cluster structures in Fig. 5 (a) indicating strong presence of the specified factor information in the unspecified latent space. This observation is consistent with the quantitative results shown in Table 1. As shown in Fig. 5 (b) and (c), the t-SNE plots for Mathieu et al.'s model [7] and our model appear to have similar levels of confusion with respect to the specified factor information. However, since t-SNE plots are approximations, the quantitative results reported in Table. 1 better capture the performance comparison.

The architectures in [7, 8] utilize adversarial training in combination with a regular and a variational auto-encoder respectively. Despite the significant presence of specified factor information in the unspecified latent embeddings from Szabó et al.'s model [8], it successfully generates novel images by combining the specified and unspecified factors (shown in Sec. 4.2). This apparently conflicting observation suggests that the decoder somehow learns to ignore the specified factor information in the unspecified latent space. We conjecture that since the adversarial loss updates the decoder and the encoder parameters together, and in that order, the encoder remains less likely to disentangle the latent spaces.

A similar argument can be made that Mathieu et al.'s [7] architecture does not explicitly train the encoder to disentangle factors of variation, thus resulting in higher classification accuracy in the unspecified latent space. This
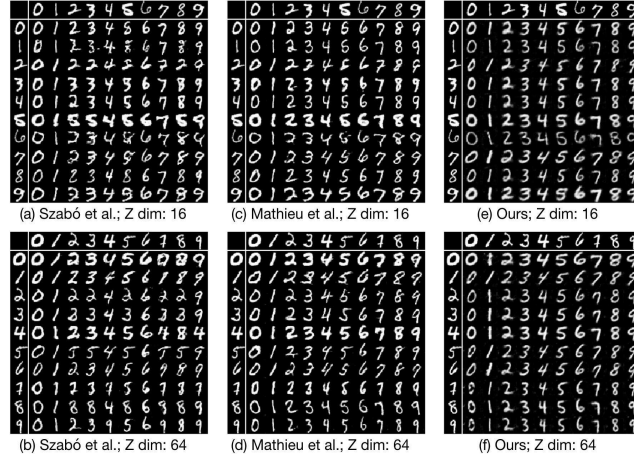

Fig. 6. Image generation results on MNIST by swapping z and s variables. The top row and the first column are randomly selected from the test set. The remaining grid is generated by taking z from the digit in first column and s from the digit in first row. This keeps the unspecified factors constant in rows and the specified factors constant in columns.

behavior, however, is mitigated to a large extent due to the VAE framework, which promotes class confusion in the unspecified latent subspace by performing reparametrization at the time of new image generation. Our approach benefits from the reparametrization as well, however, significantly lower classification accuracies on the unspecified latent space embeddings indicate that the encoder learns to disentangle the factors better by minimizing the reverse cycle-consistency loss.

4.2   Quality of Image Generation

The quality of image generation is evaluated in three different setups. First, we test the capability of our model to combine unspecified and specified factors from different images to generate a new image. The results are shown in the form of a grid of images, where the first row and the first column is taken from the test set. The remaining grid shows images generated by combining the specified factor from images in the first row and the unspecified factors of those in the first column. For this evaluation, we compare our results with prior works [7] and [8]. Unlike the non-adversarial approach proposed by Szabó et al. [8], our model is robust to the choices of dimensionality for both z and s variables. We emphasize that our model avoids *degeneracy* for significantly higher dimensions of latent variables, as compared to the base values, despite being a non-adversarial architecture. Second, we show the effect of linear interpolation in the latent spaces. The images in the top-left and the bottom-right corner are
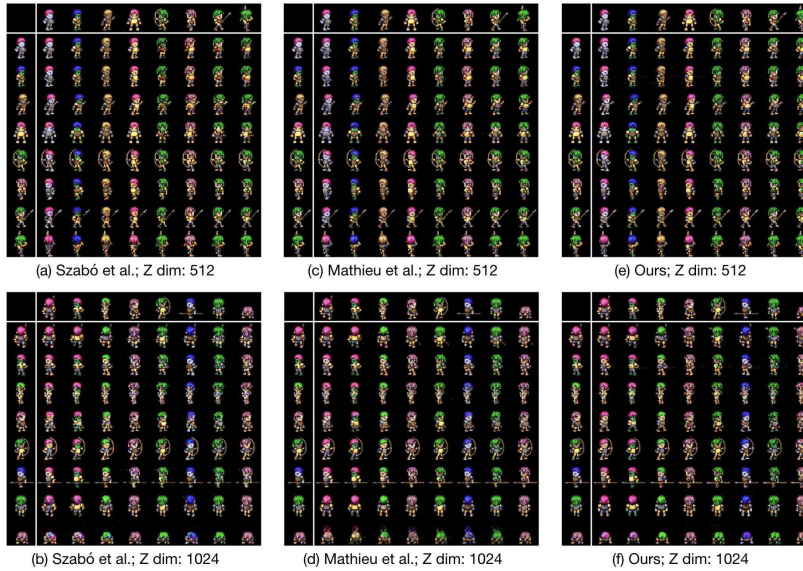

(a) Szabó et al.; Z dim: 512   (c) Mathieu et al.; Z dim: 512   (e) Ours; Z dim: 512

(b) Szabó et al.; Z dim: 1024   (d) Mathieu et al.; Z dim: 1024   (f) Ours; Z dim: 1024

Fig. 7. Image generation results on 2D Sprites by swapping z and s variables. Arrangement of the grid is same as Fig. 6.
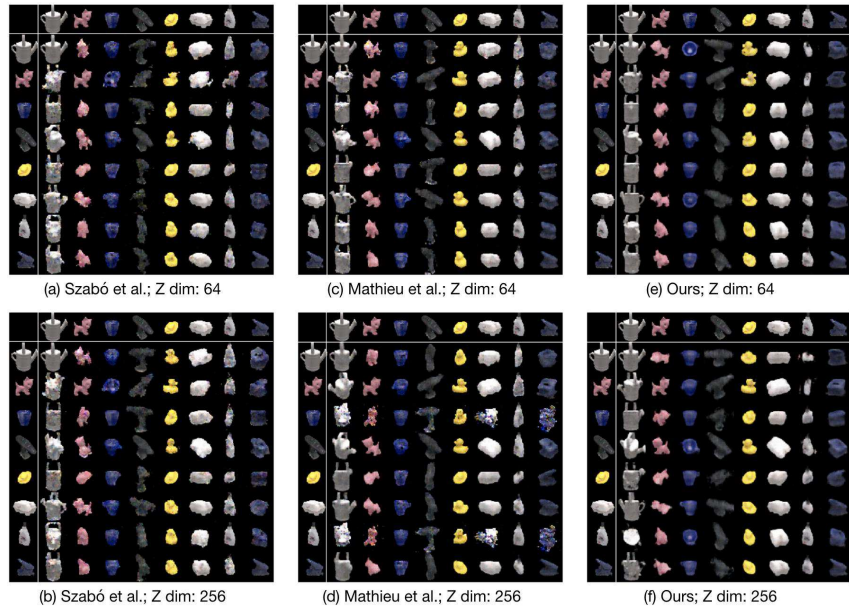
Fig. 8. Image generation results on LineMod by swapping z and s variables. Arrangement of the grid is same as Fig. 6. As explained in Sec. 4.2, we do not observe a direct transfer of viewpoint between the objects.

taken from the test set and similar to the first evaluation, the remaining images are generated by keeping z constant across the rows and s constant across the columns. Finally, we check the conditional image generation capability of our model by conditioning on the s variable and sampling data points directly from the Gaussian prior $p(z)$ for the z variable.

The first evaluation of generating new images by combining z and s from different sources is shown in Figures 6, 7 and 8. LineMod dataset comprises of images with complex geometric structures that do not share a fixed frame of reference. For example, an image of a 'duck' will not be aligned in the same direction as an image of a 'cat' for a common viewpoint. Hence, as is apparent from Fig. 8, interpretation of transfer of unspecified factors as viewpoint transfer does not exactly hold true.

The results for linear interpolation in the latent space and conditional image generation via sampling from $p(z)$ are shown in Figs. 9 and 10 respectively.

5   Conclusion

In this paper we introduced a simple yet effective way to disentangle specified and unspecified factors of variation by leveraging the idea of cycle-consistency. The proposed architecture needs only weak supervision in the form of pairs of data having similar specified factors. The architecture does not produce degenerate
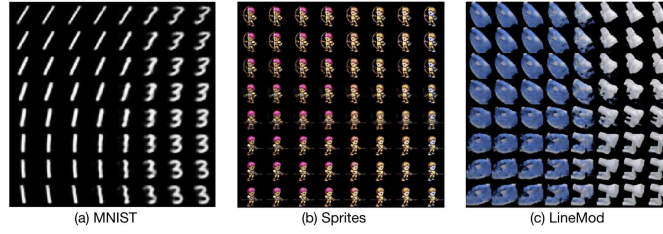
Fig. 9. Linear interpolation results for our model in the z and s latent spaces. The images in the top-left and the bottom-right corner are taken from the test set. Like Fig. 6, z variable is constant in the rows, while s is constant in the columns.
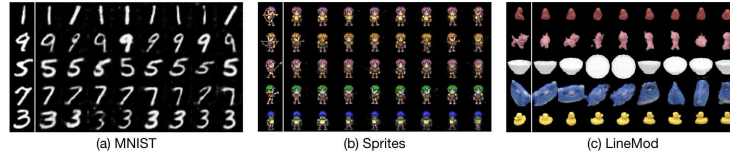

Fig. 10. Image generation by conditioning on s variable, taken from test images, and sampling the z variable from $\mathcal{N}(0, I)$.

solutions and is robust to the choices of dimensionality of the latent space. Through our experimental evaluations, we found that even though adversarial training produces good visual reconstructions, the encoder does not necessarily learn to disentangle the factors of variation effectively. Our model, on the other hand, achieves compelling quantitative results on three different datasets and shows good image generation capabilities as a generative model.

We also note that generative models based on VAEs produce less sharper images compared to GANs and our model is no exception. One way to address this problem could be to train our cycle-consistent VAE as the first step, followed by training the decoder with a combination of adversarial and reverse cycle-consistency loss. This training strategy may improve the sharpness of the generated images while maintaining the disentangling capability of the encoder. Another interesting direction to pursue from here would be to further explore the methods that disentangle factors of variation in an unsupervised manner.

Acknowledgement

References

1. Edwards, H., Storkey, A.J.: Censoring Representations with an Adversary. In: International Conference in Learning Representations. ICLR2016 (2016)
2. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.S.: The Variational Fair Autoencoder. In: International Conference in Learning Representations. ICLR2016 (2016)
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet Classification with Deep Convolutional Neural Networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12 (2012) 1097–1105
4. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative Adversarial Nets. In: NIPS. (2014) 2672–2680
5. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: International Conference in Learning Representations. ICLR2016 (2016)
6. Xiao, T., Hong, J., Ma, J.: DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images. arXiv preprint **arXiv:1711.05415** (2017)
7. Mathieu, M., Zhao, J.J., Sprechmann, P., Ramesh, A., LeCun, Y.: Disentangling Factors of Variation in Deep Representation using Adversarial Training. In: NIPS. (2016) 5041–5049
8. Szabó, A., Hu, Q., Portenier, T., Zwicker, M., Favaro, P.: Challenges in Disentangling Independent Factors of Variation. arXiv preprint **arXiv:1711.02245** (2017)
9. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: International Conference in Learning Representations. ICLR2014 (2014)
10. He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., Ma, W.: Dual Learning for Machine Translation. In: Advances in Neural Information Processing Systems 29. (2016) 820–828
11. Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning Dense Correspondence via 3D-Guided Cycle Consistency. In: CVPR, IEEE Computer Society (2016) 117–126
12. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: CVPR. (2017)
13. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial networks. In: ICCV, IEEE Computer Society (2017) 2242–2251
14. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: CyCADA: Cycle-Consistent Adversarial Domain Adaptation. arXiv preprint **arXiv:1711.03213** (2017)
15. Ghahramani, Z.: Factorial Learning and the EM Algorithm. In: Proceedings of the 7th International Conference on Neural Information Processing Systems. NIPS'94, Cambridge, MA, USA, MIT Press (1994) 617–624
16. Tenenbaum, J.B., Freeman, W.T.: Separating Style and Content with Bilinear Models. Neural Computation **12**(6) (2000) 1247–1283
17. Desjardins, G., Courville, A.C., Bengio, Y.: Disentangling Factors of Variation via Generative Entangling. arXiv preprint **arXiv:1210.5474** (2012)
18. Reed, S.E., Sohn, K., Zhang, Y., Lee, H.: Learning to Disentangle Factors of Variation with Manifold Interaction. In: ICML. Volume 32 of JMLR Workshop and Conference Proceedings., JMLR.org (2014) 1431–1439

19. Tang, Y., Salakhutdinov, R., Hinton, G.E.: Deep Lambertian Networks. In: ICML, icml.cc / Omnipress (2012)
20. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.B.: Deep Convolutional Inverse Graphics Network. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. NIPS'15, Cambridge, MA, USA, MIT Press (2015) 2539–2547
21. Tran, L., Yin, X., Liu, X.: Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
22. Donahue, C., Balsubramani, A., McAuley, J., Lipton, Z.C.: Semantically Decomposing the Latent Spaces of Generative Adversarial Networks. In: International Conference in Learning Representations. ICLR2018 (2018)
23. Berthelot, D., Schumm, T., Metz, L.: BEGAN: boundary equilibrium generative adversarial networks. arXiv preprint **arXiv:1703.10717** (2017)
24. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep Visual Analogy-Making. In: NIPS. (2015) 1252–1260
25. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial autoencoders. In: International Conference on Learning Representations. (2016)
26. Hu, Q., Szabó, A., Portenier, T., Zwicker, M., Favaro, P.: Disentangling Factors of Variation by Mixing Them. arXiv preprint **arXiv:1711.07410** (2017)
27. Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018. (2018)
28. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based Learning Applied to Document Recognition. In: Proceedings of the IEEE. (1998) 2278–2324
29. http://lpc.opengameart.org/: Liberated Pixel Cup Accessed: 2018-02-21.
30. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-less 3D Objects in Heavily Cluttered Scenes. In: Proceedings of the 11th Asian Conference on Computer Vision - Volume Part I. ACCV'12, Berlin, Heidelberg, Springer-Verlag (2013) 548–562
31. Wohlhart, P., Lepetit, V.: Learning Descriptors for Object Recognition and 3D Pose Estimation. In: CVPR, IEEE Computer Society (2015) 3109–3118
32. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic Differentiation in PyTorch. (2017)
33. van der Maaten, L., Hinton, G.: Visualizing High-Dimensional Data using t-SNE. Journal of Machine Learning Research **9: 2579-2605** (Nov 2008)