# A Cyclically-Trained Adversarial Network for Invariant Representation Learning

Jiawei Chen
Boston University
garychen@bu.edu

Janusz Konrad
Boston University
jkonrad@bu.edu

Prakash Ishwar
Boston University
pi@bu.edu

## Abstract

*Recent studies show that deep neural networks are vulnerable to adversarial examples which can be generated via certain types of transformations. Being robust to a desired family of adversarial attacks is then equivalent to being invariant to a family of transformations. Learning invariant representations then naturally emerges as an important goal to achieve which we explore in this paper within specific application contexts. Specifically, we propose a cyclically-trained adversarial network to learn a mapping from image space to latent representation space and back such that the latent representation is invariant to a specified factor of variation (e.g., identity). The learned mapping assures that the synthesized image is not only realistic, but has the same values for unspecified factors (e.g., pose and illumination) as the original image and a desired value of the specified factor. Unlike disentangled representation learning, which requires two latent spaces, one for specified and another for unspecified factors, invariant representation learning needs only one such space. We encourage invariance to a specified factor by applying adversarial training using a variational autoencoder in the image space as opposed to the latent space. We strengthen this invariance by introducing a cyclic training process (forward and backward cycle). We also propose a new method to evaluate conditional generative networks. It compares how well different factors of variation can be predicted from the synthesized, as opposed to real, images. In quantitative terms, our approach attains state-of-the-art performance in experiments spanning three datasets with factors such as identity, pose, illumination or style. Our method produces sharp, high-quality synthetic images with little visible artefacts compared to previous approaches.*

## 1. Introduction

The performance of machine learning algorithms is usually related to the quality of internal data representation (features). Thus, representation learning has been extensively studied in the fields of machine learning and artificial intelligence (AI). Among the criteria for a "good" data representation, invariance to one or more specified factors of variation while simultaneously preserving other factors is an important property since it could benefit the end-task (e.g., classification) by factoring out irrelevant information [5]. Immunity against imperceptible perturbations, which could mislead trained classifiers [19, 2, 33], can be attained by training a network to generate a representation that is invariant to such transformations [25]. Invariance to factors such as gender or race is also useful in applications where it is desirable that decisions not be biased towards or against a particular group. In such cases, the data representation must not contain group-identifying information, but must preserve other attributes.

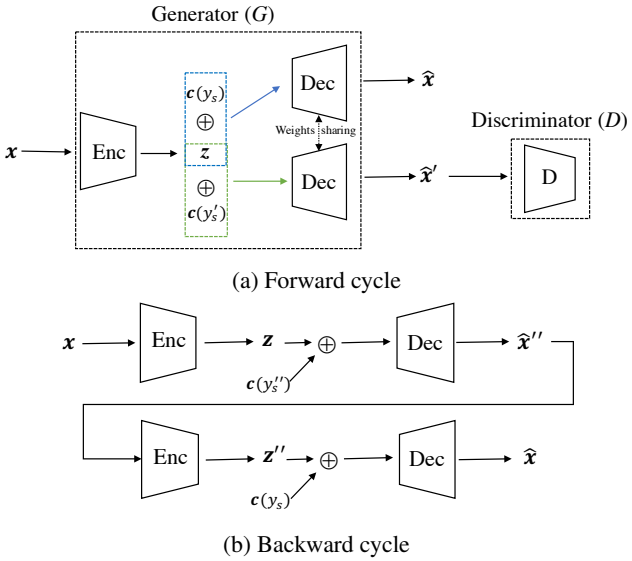



Figure 1: Diagram of the proposed model ($\oplus$ represents concatenation): (a) forward cycle in which training alternates between optimizing $G$ and $D$; (b) backward cycle that only optimizes $G$. Note: the label for image synthesis is denoted by $y_s'$ in the forward cycle and $y_s''$ in the backward cycle.

Invariant representation learning has been studied in dif-

ferent domains, such as fair decision making [15, 6, 41], privacy-preserving visual analytics [9, 8, 7], and domain adaptation [31, 23, 16]. It is also closely related to disentangled representation learning, as both attempt to separate factors of variation in the data. The major difference between them is that invariant representations eliminate unwanted factors in order to reduce sensitivity in the direction of invariance, while disentangled representations try to preserve as much information about the data as possible [5]. In this work, we are interested in the problem of learning image representations that are invariant to certain *specified* factors of variation, e.g., identity, while preserving other *unspecified* factors, e.g., pose, expression, etc., using adversarial training. Our approach also enables us to explicitly set the value of the specified factor within a synthesized image.

Generative models that are driven by an interpretable latent space, i.e., one where the data representations can be used to control certain characteristics of the outputs (e.g., create images from a particular class), are often preferable. Such models are useful in a variety of applications, e.g, automatic image editing [7, 26, 28]. Previous studies [7, 28, 39] combined the generative power of the Variational Auto-Encoder (VAE) [27] and the Generative Adversarial Network (GAN) [18] to learn an invariant latent space which enables controlling a specified factor of variation in the synthesized samples. However, they either require training labels for both specified and unspecified factors of interest [7, 39] or are limited to binary attributes [28].

Our proposed framework also builds upon VAEs and GANs. We combine the two by structuring the generator ($G$) in a conventional GAN as an encoder-decoder pair (see Fig. 1a). In order to improve the invariance of data representations and quality of synthesized images, we introduce a forward-backward cyclic training process. During a forward cycle, the generator is given an input image $\mathbf{x}$ with a specified factor label $y_s$. The encoder maps $\mathbf{x}$ to a latent representation $\mathbf{z}$, and the decoder is trained to reconstruct $\mathbf{x}$ based on $(\mathbf{z}, y_s)$ as well as synthesize a realistic image based on $(\mathbf{z}, y'_s)$, that can fool the discriminator into classifying it to class $y'_s$, where $y'_s$ is a generated class code. This encourages the encoder to reduce information about the specified factor in the latent representation as the specified factor of a generated sample is determined by the class code. Meanwhile, the encoder is also encouraged to pass information about the unspecified factors to the latent space to allow an accurate reconstruction. However, a forward cycle alone does not prevent a *degenerate* solution wherein information about the specified factor still leaks into the latent space, but the decoder learns to ignore that information. Therefore, in the backward cycle, we impose a further constraint in the latent space by explicitly minimizing the distance between two latent representations one for a real training image $\mathbf{x}$ associated with label $y_s$, and another for a synthesized sample generated based on $(\mathbf{x}, y''_s)$, where $y_s \neq y''_s$. This forces the encoder to only encode unspecified factors within the latent space. The generator is additionally trained to reconstruct the real image from its synthetic version with the appropriate class code, which could benefit the image synthesis task.

Once trained, our model becomes a conditional image generator that can synthesize novel images with the ability to change the specified factor value by tuning the class code (using the forward cycle only). In order to measure the quality of our model, we follow previous studies [20, 21] and conduct a subjective visual evaluation. We also propose a *quantitative* method to evaluate conditional generative models by measuring how well different factors of variation can be predicted in the synthesized images *via* pretrained attribute estimators.

This paper makes the following contributions:

1. We develop a *conditional Variational Generative Adversarial Network* for learning a representation that is invariant to a specified factor, while preserving other unspecified factors of variation.

2. We empirically verify the effectiveness of the proposed model in learning invariant representations *via* a forward-backward cyclic training process on a number of datasets and tasks.

3. We propose a new quantitative method to evaluate the quality of conditional generative models and show that our model consistently produces better quality images compared to two state-of-the-art methods.

## 2. Related Work

**Invariant Representation Learning:** It has been extensively studied in various contexts and the related literature is vast. For instance, transformation-invariant feature learning has deep roots in computer vision; features are often designed for a specific case, e.g., rotation or scale invariance. Early examples include hand-crafted features such as HOG [14] and SIFT [34]. More recently, deep Convolutional Neural Networks (CNNs) have been very effective in learning transformation-invariant representations [11, 13, 37]

Another line of research aims at building fair, bias-free classifiers that also attempt to learn representations invariant to "nuisance variables", which could potentially induce bias or unfairness [32, 41, 15, 40]. One study proposed to obtain fairness by imposing $l_1$ regularization between representation distributions for data with different nuisance factors of variation [32]. The Variational Fair Auto-Encoder [41] tackles the same task using a VAE with maximum mean discrepancy regularization. Particularly relevant to our work are the methods proposed in [15, 40], that also incorporate adversarial training in an auto-encoder framework. Our

method differs in that we apply adversarial training in the image space instead of the latent space. This creates better quality images. Additionally, we improve the quality of invariance through cyclic training.

Our work is also related to two recent studies [7, 39]. Both studies develop a fully-supervised method with adversarial training in the image space to learn invariant representations by factoring out nuisance variables for a specific task, e.g., identity-invariant expression recognition. However, both methods can only retain certain factors of variation (with labels available for training) in the representations. Another drawback is that in those models a discriminator is trained for each of the factors of variation so the number of model parameters grows linearly with the number of factors. In contrast, our model uses a single discriminator which only requires labels of the specified factor of variation. Moreover, our model is designed to automatically capture all unspecified factors of the data into the representation with no need for corresponding labels.

**Disentangled Representation Learning:** Invariant representation learning has a natural connection to disentangled representation learning, where the goal is to factorize different influencing factors of the data into different parts of its representation. In an early study, a bilinear model was proposed to separate content and style for face and text images [38]. Another method used an E-M algorithm to discover the independent factors of variation of the underlying data distribution [17]. Later, unsupervised approaches to learn disentangled image representations were proposed [10, 24]. A purely-generative model was developed in [10] but, unlike our model, it has no capacity to create an invariant representation for a given image. A method proposed in [24] can learn an image representation that consists of a pre-defined number of disentangled factors of variation, but it has no control over which factors to learn. Recent methods [35, 20] proposed to combine auto-encoder with adversarial training to disentangle specified and unspecified factors of variation and map them onto separate latent spaces. Indeed, the resulting unspecified representation is equivalent to an invariant representation that is disentangled from the specified factor. However, methods with sole pixel-wise reconstruction objective in the image space tend to produce blurry images. Another recent work [21] proposed a cycle-consistent VAE to disentangle the latent space into two complementary subspaces by using weak supervision (pairwise similarity labels). It is related to our work in the sense that both methods constrain the latent space by adding a pair-wise distance between two latent representations (which are supposed to be close) into the cost function. The difference is that our method leverages adversarial training based on a single source of supervision, enabling training with a single image in each iteration instead of a pair of images. Moreover, we impose cycle-consistency loss in the image space as opposed to the latent space in [21]. Such modification, along with the additional adversarial loss in the image space, promotes generation of better quality invariant representations and images.

It is also worthwhile to mention that our work is related to several previous works on image generation [29, 4] in terms of using auto-encoder and GAN. However, our goals are very different. The previous works mainly focus on developing image generation models, whereas our model is explicitly optimized to create invariant image representations. Once trained, our model becomes a conditional image generator.

## 3. Methodology

Let $\mathbf{X}$ denote the image domain and $\mathbf{Y} = \{y_1, ..., y_K\}$ a set of $K$ possible factors of variation associated with data samples in $\mathbf{X}$. Given an image $\mathbf{x} \in \mathbf{X}$ and one specified factor $y_s$, where $y_s \in \{1, ..., N_s\}$ and $N_s$ is the number of possible classes, our proposed approach has two objectives: 1) to learn a latent representation $\mathbf{z}$ which is invariant to the specified factor but preserves the other unspecified factors of variation, and 2) to synthesize a realistic sample $\hat{\mathbf{x}}'$ which has the same unspecified factors as $\mathbf{x}$ and a desired specified factor value which is determined by an input class code $\mathbf{c}(y'_s)$, where $y'_s \in \{1, ..., N_s\}$ is generated from a distribution $p(y'_s)$ and $\mathbf{c}(\cdot)$ is a one-hot encoding function. For simplicity, we consider here the case where $y_s$ is categorical, but our approach can be extended to continuous $y_s$.

**Generator:** We structure the generator in the proposed model as an encoder-decoder pair (Fig. 1). The encoder ($Enc$) aims to create a low-dimensional data representation $\mathbf{z} = Enc(\mathbf{x})$ *via* a randomized mapping $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x})$ parameterized by the weights of the encoder's neural network $\theta_{enc}$. On the other hand, the decoder ($Dec$) is a neural network with weights $\theta_{dec}$. It is responsible for learning a mapping function $\hat{\mathbf{x}}' \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y'_s))$ that can map the latent representation $\mathbf{z}$ in combination with class code $\mathbf{c}(y'_s)$ back to the image space. The latent space is regularized by imposing a prior distribution, in our experiments a normal distribution $r(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

**Discriminator:** Different from the discriminators in conventional GANs, the discriminator $D$ in our model is a multi-class classifier represented by a neural network with weights $\theta_{dis}$. The outputs of the discriminator $D(\mathbf{x}) \in \mathbb{R}^{N_s+1}$ are the predicted probabilities of each class corresponding to $N_s$ different values of the specified factor and an additional "fake" class for synthesized images.

**Forward cycle:** First, we sample an image $\mathbf{x}$ from the training set and pass it through the encoder to generate a latent representation $\mathbf{z}$. The decoder is trained to produce a reconstruction of the input $\hat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y_s))$ and also to synthesize a new data sample $\hat{\mathbf{x}}' \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y'_s))$ that can fool the discriminator $D$ into classifying it as the specified class

$y'_s$. Specifically, the weights of the generator network are adjusted to *minimize* the following cost function:

$$\begin{aligned}\mathcal{L}_G^{fw}(G,D) = \\ -\lambda_1^G E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}\left[\log D_{y'_s}\left(G(\mathbf{x},\mathbf{c}(y'_s))\right)\right]+ \\ \lambda_2^G E_{(\mathbf{x},y_s)\sim p(\mathbf{x},y_s)}\left[||\mathbf{x}-G(\mathbf{x},\mathbf{c}(y_s))||_2^2\right]+ \\ \lambda_3^G KL(p(\mathbf{z}|\mathbf{x})||r(\mathbf{z}))\end{aligned} \quad (1)$$

where $p(\mathbf{x}, y_s)$ denotes the joint distribution of the real image and the specified factor in the training data, $p(\mathbf{x})$ is the corresponding marginal distribution of the real image, $p(y'_s)$ is a distribution of the specified factor used to synthesize a "fake" image, $D_i$ is the predicted probability of the $i$-th class, and $\lambda_1^G, \lambda_2^G, \lambda_3^G$ are weighting factors.

The discriminator aims to correctly classify a real training sample $\mathbf{x}$ to its ground-truth class value $y_s$ of the specified attribute but, when given a synthetic sample $\hat{\mathbf{x}}'$ from the generator, it attempts to classify it as fake. This is accomplished by adjusting the weights of the discriminator by *maximizing* the following cost function:

$$\begin{aligned}\mathcal{L}_D^{fw}(G,D) = \\ \lambda_1^D E_{(\mathbf{x},y_s)\sim p(\mathbf{x},y_s)}[\log D_{y_s}(\mathbf{x})]+ \\ \lambda_2^D E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}[\log D_{N_s+1}(G(\mathbf{x},\mathbf{c}(y'_s)))]\end{aligned} \quad (2)$$

where $\lambda_1^D$ and $\lambda_2^D$ are tuning parameters.

The weights of the networks in $G$ and $D$ are updated in an alternating order. Over successive training steps, $G$ learns to fit the true data distribution and reconstruct the input image as well as synthesize realistic images that can fool $D$. The generator objective (second term in Eq. (1)) encourages the encoder to pass as much information about the unspecified factors as possible to the latent representation. Since the class code $\mathbf{c}$ determines the specified factor value of $\hat{\mathbf{x}}'$, the encoder is also encouraged to eliminate information about the specified factor of $\mathbf{x}$ in the latent representation. The encoder may, however, fail to disentangle the specified and unspecified factors of variation and the decoder may still learn to synthesize images according to the class code $\mathbf{c}$ by ignoring any *residual* information about the specified factor that is contained within the representation. To avoid such a *degenerate* solution, we use a backward cycle to further constrain the latent space.

**Backward cycle:** This cycle requires a synthesized image $\hat{\mathbf{x}}''$ of class $y''_s$ generated from a real image $\mathbf{x}$ of class $y_s$. We intentionally choose $y''_s \neq y_s$ so that $\hat{\mathbf{x}}''$ and $\mathbf{x}$ carry different specified factor values. Two latent representations $\mathbf{z} = Enc(\mathbf{x})$ and $\mathbf{z}'' = Enc(\hat{\mathbf{x}}'')$ can be computed by passing, respectively, $\mathbf{x}$ and $\hat{\mathbf{x}}''$ through the encoder. If the encoder fails to transmit information about unspecified factors from the input to its latent representation, or if it retains considerable information about the specified factor in the latent space, then $\mathbf{z}$ and $\mathbf{z}''$ are expected to have a large pairwise distance. In other words, if the latent space only maintains information about the unspecified factors, $\mathbf{z}$ should be equivalent to $\mathbf{z}''$. In addition, we would like to encourage the generator to reconstruct the input $\mathbf{x}$ from its synthetic version $\hat{\mathbf{x}}''$ in combination with a class code $\mathbf{c}(y_s)$ that encodes the ground-truth label of the specified factor of $\mathbf{x}$. These considerations motivate optimizing the generator in the backward cycle by minimizing the following cost function:

$$\begin{aligned}\mathcal{L}_G^{bw} = E_{(\mathbf{x},y_s)\sim p(\mathbf{x},y_s),y''_s\sim p(y''_s)}\left[\lambda_1^{bw}||\mathbf{z}-\mathbf{z}''||_1+\right. \\ \left.\lambda_2^{bw}||\mathbf{x}-G(\hat{\mathbf{x}}'',\mathbf{c}(y_s))||_2^2\right]\end{aligned} \quad (3)$$

where $\lambda_1^{bw}$ and $\lambda_2^{bw}$ are two weighting factors. The first term in Eq. (3) penalizes the generator if $\mathbf{z}$ is not close to $\mathbf{z}''$. The second term encourages the synthesized $\hat{\mathbf{x}}$ to resemble $\mathbf{x}$.

Essentially, the forward cycle translates $\mathbf{x}$ to a synthetic image $\hat{\mathbf{x}}'' = G(\mathbf{x}, \mathbf{c}(y''_s))$ followed by a backward transform $\hat{\mathbf{x}} = G(\hat{\mathbf{x}}'', \mathbf{c}(y_s))$, such that $\hat{\mathbf{x}} \simeq \mathbf{x}$. This cyclic training process assists the model in generating good quality images and further encourages invariance to the specified factor in the latent space

## 4. Experimental Evaluation

We evaluate the performance of the proposed model on three image datasets: 3D Chairs [3], YaleFace [30] and UPNA Synthetic [1]. We first conduct a quantitative evaluation of the degree of invariance in the latent space by training dedicated neural networks (one per factor) to predict the values of the specified and certain unspecified factors (that have ground-truth labels) from the latent representation. The factor prediction accuracies quantify how much information about each factor has been preserved in the latent representation. If the model succeeds in eliminating all information about the specified factor and preserving all information about unspecified factors, we should expect the prediction accuracy for the specified factor to be close to pure chance and the prediction accuracies for the unspecified factors to be nearly perfect. We also evaluate the quality of the image generation process. Unlike previous works [20, 21], which only provide a qualitative evaluation through visual inspection of the synthesized images, we propose a new method to quantitatively assess the ability of a conditional generative model to synthesize realistic images while preserving unspecified factors. In Section 4.3, we present details of the proposed evaluation method and associated experimental results.

We compare our model with two state-of-the-art methods [20, 21] that learn to produce, for a given input image, two latent vectors (as opposed to just one in our method). One of the latent vectors captures information related to the

unspecified factors of variation and is, in an ideal scenario, devoid of any information related to the specified factor of variation. This latent vector is the counterpart of the latent invariant representation in our method. For synthesizing an image with a desired value for the specified factor, the methods in [20, 21] require an additional surrogate image which has the desired value for the specified factor. They would then *substitute* the latent vector of the specified factor in the original image with that of the surrogate image and then decode the result. Our approach, in contrast, uses a class code (as opposed to a surrogate image) to explicitly set the value of the specified factor in the synthesized image. In our experiments, we compare the latent vectors for the unspecified factors from the competing methods and the latent representation from our method in terms of their ability to predict the specified and unspecified factors which indicates the quality of invariance. We used the publicly-available source code to implement both benchmarks, but slightly modified their network architectures to ensure that all three competing models have similar numbers of parameters. We also did parameter tuning for each method for each of the three datasets.

### 4.1. Datasets

**3D Chairs:** This dataset includes 1,393 3D chair styles rendered on a white background from 62 different viewpoints that are indexed by two values of angle $\theta$ and 31 values of angle $\phi$. Each image is annotated with the chair identity indicating its style as well as viewpoint $(\theta, \phi)$. For each chair style, we randomly picked 50 images (out of 62) to populate the training set, and used the remaining 12 images in the testing phase. This gives, in total, 69,650 images in the training set, and 16,716 images in the test set.

**YaleFace:** This dataset consists of gray-scale frontal face images of 38 subjects under 64 illumination conditions. In our experiments, we randomly chose 54 images (out of 64) from each subject for training, and used the rest as the test set for performance evaluation.

**UPNA Synthetic:** This is a synthetic human head-pose database. It consists of 12 videos for each of 10 subjects; 120 videos in total with 38,800 frames. Ground-truth *continuous* head pose angles (yaw, pitch, roll) are provided for each frame. We randomly selected $85\%$ of the frames from each video for each subject for the training and used the remaining $15\%$ for testing.

For computational efficiency, in our experiments, we resized each RGB image to $64 \times 64$-pixel resolution for all three datasets. Table 1 summarizes the specified and unspecified factors of variation that we investigate across the three datasets.

### 4.2. Quality of invariance

We follow previous methodology [21, 20] and train dedicated neural network estimators to predict the specified and unspecified factors of variation based on the learned latent representations generated by each competing model. We use correct classification rate (CCR) and mean absolute error (MAE) to measure the performance of classification tasks and regression tasks, respectively. Table 2 summarizes the performance of each model on the three image datasets.

Table 1: Specified and unspecified factor(s) of variation investigated in the three datasets.

| Dataset | Specified factor | Unspecified factor(s) |
|---|---|---|
| 3D Chairs | Chair style | View orientation $(\theta, \phi)$ |
| YaleFace | Identity | Illumination Cond. |
| UPNA Synthetic | Identity | Head pose |

In the 3D Chairs dataset, we regard chair style as the specified factor and the viewing orientation angles as the unspecified factors. Since both orientation angles are discrete, we treat viewing orientation estimation as a classification problem. As shown in Table 2, all three competing models manage to reduce the style information contained within the latent representation to a large extent (very low style prediction CCR values). However the proposed model (with backward cycle) outperforms the benchmark models, in terms of the ability to predict the viewing orientation angles, by a large margin (about 11–28$\%$ CCR improvement for $\phi$ and 9–13$\%$ CCR improvement for $\theta$). We also note that the backward cycle significantly improves invariance, e.g., style prediction CCR decreases from 3.21$\%$ to 0.79$\%$.

For the YaleFace dataset, subject identity is considered as the specified factor and illumination condition as the unspecified factor of variation. We first observe that the identification performance of the three models is comparable and close to a random guess, which suggests the competing models perform equally well in creating representations that are invariant to identity. For the recognition of illumination condition, the classification CCR for our model is $85.50\%$, which again surpasses the two benchmark CCRs by about $8\%$ and $53\%$ in accuracy. Such large performance gaps suggest that the invariant representation learned by our model is better, than the competing alternatives, in preserving information about unspecified factors of variation. Furthermore, we observe that the backward cycle helps reduce the identification CCR by about $5\%$, thus confirming its usefulness.

In the case of UPNA Synthetic dataset, the specified and unspecified factors of variation used in evaluation are subject identity and head pose, respectively. Head pose is defined as a three-dimensional angular value (yaw, pitch, roll) in continuous space. Thus, we train neural-network based regressors to estimate head pose and report the mean and standard deviation of the absolute errors for yaw, pitch and roll angles separately. In terms of identification accuracy, the performance of the three methods is similar (no more than $3\%$ difference in CCR or about 2-3 times that of a random guess). We also notice that the backward cycle greatly promotes invariance as it helps to reduce identification CCR

Table 2: Evaluation of the quality of invariance of representations generated by the competing models on 3D Chairs, YaleFace and UPNA Synthetic datasets. Classification performance is measured using CCR. Regression performance is measured using MAE and standard deviation. ↑ means higher is better. ↓ means lower is better.

| **Datasets** | **Factors of variation** | **Methods** | | | | |
|---|---|---|---|---|---|---|
| | | Random guess/ Median | [20] | [21] | Ours | Ours w/o b.w. cycle |
| 3D Chairs | Chair Style ↓ | 0.07% | 0.77% | 0.70% | 0.79% | 3.21% |
| | $\theta$ ↑ | 50% | 68.92% | 64.22% | 78.17% | 74.37% |
| | $\phi$ ↑ | 3.22% | 50.23% | 43.75% | 71.90% | 69.45% |
| YaleFace | Identity ↓ | 2.63% | 4.68% | 5.50% | 6.97% | 12.36% |
| | Illumination Cond. ↑ | 1.56% | 77.80% | 32.36% | 85.50% | 85.40% |
| UPNA Synthetic | Identity ↓ | 10% | 15.80% | 18.83% | 18.05% | 33.40% |
| | Yaw ↓ | 5.10°±6.70° | 2.77°±2.00° | 2.42°±2.52° | 2.12°±2.12° | 2.10°±2.08° |
| | Pitch ↓ | 4.98°±5.02° | 2.43°±2.10° | 2.88°±2.71° | 2.23°±2.10° | 2.20°±2.06° |
| | Roll ↓ | 4.68°±6.88° | 1.19°±1.43° | 1.65°±2.35° | 1.16°±1.24° | 1.29°±1.43° |

from 33.40% to 18.05%. As for head-pose estimation, we use "Median" estimate as a baseline, i.e., the median value of ground truth across the entire training set. We note that our model slightly, but consistently, outperforms the benchmarks, and significantly outperforms the median estimate. This once again confirms the effectiveness of our model in preserving information pertaining to the unspecified factors in the latent representation while discarding information related to the specified factor.

### 4.3. Quality of image generation

Many studies have proposed measures to evaluate generative models for image synthesis. Some of them attempt to quantitatively evaluate models while some others emphasize qualitative approaches, such as user studies (e.g., visual examination). However, such subjective assessment may be inconsistent and not robust as human operators may fail to distinguish subtle differences in color, texture, etc. In addition, such a measure may favor models that can merely memorize training samples. In terms of quantitative methods, some studies proposed to use measures from image quality assessment literature such as SSIM, MSE and PSNR. However, they require a corresponding reference real image for each synthesized one. Other widely-adopted reference-free quantitative measures like Inception Score [36] and Fréchet Inception Distance [22] are designed for generic GANs. Thus, they are not suitable for conditional models that aim to generate samples from a particular class. Several quantitative evaluation methods have been proposed for conditional generative models [12, 42]. For example, [42] proposed to feed fake colorized images (of real grayscale images) to a classifier that was trained on real color images. If the classifier performs well, this indicates that the colorization is accurate. In contrast to these works, we evaluate multiple objectives simultaneously: one to evaluate invariance to a target attribute and others to evaluate the preservation of un-specified attributes.

Inspired by the previous studies that use an off-the-shelf classifier to assess the realism of synthesized data, we propose a quantitative method that utilizes a number of attribute estimators to evaluate the quality of conditional generative models. The intuition is that a good generative model for learning an invariant/disentangled representation should have the capability to explicitly and accurately control the specified factor value when it generates a novel image. Furthermore, it should precisely transfer the other unspecified factors of variation from the source image to its synthetic version. Therefore, we can evaluate a model by measuring how well the different factors of variation in the synthesized images can be predicted *via* estimators that are pretrained on the real images.

Specifically, we train a number of attribute estimators $\mathcal{F}^j$, where $j \in \{1, ..., K\}$, on the original training sets of real images. For each (real) test image $\mathbf{x}$ having specified and unspecified factors of variation $y_j$, $j \in \{1, \ldots, K\}$, we synthesize a new version $\hat{\mathbf{x}}' = G(\mathbf{x}, \mathbf{c}(y'_s))$ using the generator, where $y'_s$ is sampled at random, independently of $\mathbf{x}, y_s$, from a distribution $p(y'_s)$. The image $\hat{\mathbf{x}}'$ thus synthesized is passed to the pretrained estimators to obtain a prediction for each attribute (whether specified or unspecified). If a factor of variation $y_j$ is categorical, then $\mathcal{F}^j(\hat{\mathbf{x}}')$ is a probability distribution over the set of all possible values that factor can take. In particular, $\mathcal{F}^j_{y_j}(\hat{\mathbf{x}}') = p(y_j|\hat{\mathbf{x}}')$. If $y_j$ is continuous, then $\hat{y_j} := \mathcal{F}^j(\hat{\mathbf{x}}')$ is a numerical value which should be approximately equal to $y_j$. In order to quantify performance, we introduce the following *Generator Label Score* (*GLS*) for both discrete and continuous factors of variation. For a categorical unspecified factor $y_j$,

$$GLS := E_{(\mathbf{x},y_j)\sim p(\mathbf{x},y_j),y'_s\sim p(y'_s)}\left[\mathcal{F}^j_{y_j}\left(G(\mathbf{x}, \mathbf{c}(y'_s))\right)\right]$$

whereas for a categorical specified factor $y_s$,

$$GLS := E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}\left[\mathcal{F}^j_{y'_s}\left(G(\mathbf{x}, \mathbf{c}(y'_s))\right)\right].$$

For a quantitative unspecified factor $y_j$,

$$GLS := E_{(\mathbf{x},y_j)\sim p(\mathbf{x},y_j),y'_s\sim p(y'_s)}||\mathcal{F}^j\left(G(\mathbf{x}, \mathbf{c}(y'_s))\right) - y_j||^p$$

whereas for a quantitative unspecified factor $y_s$,

$$GLS := E_{\mathbf{x}\sim p(\mathbf{x}), y'_s \sim p(y'_s)} ||\mathcal{F}^j\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big) - y'_s||^p.$$

For a good conditional generative model, the value of *GLS* should be high for every categorical factor of variation (specified or unspecified) and low for every quantitative factor. If the relative importance of each attribute is known, *GLS* values can be converted to a single value. Although quantitative, $GLS$ need not correlate well with the subjective quality of synthesized images as perceived by humans.

In order to compute *GLS*, we use the three competing models to create, separately, synthetic versions of test images for each dataset. For the proposed model, the input image $\mathbf{x}$ and class code $\mathbf{c}$ provide the necessary information about unspecified and specified factors, respectively. Thus, we synthesize a new version for each test image by passing it through the generator in combination with a randomly-generated class code. For the benchmark methods, we follow the procedure described in the respective papers to generate new images. In order to generate a new sample, we combine the unspecified latent representation of a test image and the specified latent representation of another image randomly picked from the same test set.

Table 3: Comparison of *GLS* values for the competing models. ↑ means higher is better. ↓ means lower is better.

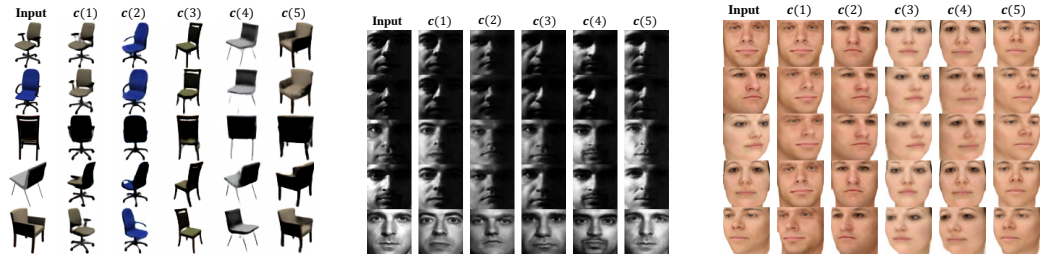
| Datasets | Factors of variation | Methods | | | |
|---|---|---|---|---|---|
| | | [20] | [21] | Ours | Ours w/o b.w. cycle |
| 3D Chairs | Chair Style ↑ | 0.02 | 0.02 | **0.87** | 0.77 |
| | $\theta$ ↑ | 0.56 | 0.61 | **0.66** | 0.66 |
| | $\phi$ ↑ | 0.38 | 0.49 | **0.57** | 0.52 |
| YaleFace | Identity ↑ | 0.24 | 0.07 | **0.98** | 0.97 |
| | Illumination Cond. ↑ | 0.17 | 0.29 | **0.70** | 0.68 |
| UPNA | Identity ↑ | 0. 88 | 0.98 | **1.00** | 0.99 |
| | Yaw ↓ | 3.51 | 2.65 | **2.55** | 2.62 |
| | Pitch ↓ | 4.07 | 2.84 | **2.46** | 2.95 |
| | Roll ↓ | 3.17 | 1.47 | **1.37** | 1.39 |

Table 3 reports the *GLS* for the three datasets. We first observe that the proposed model consistently achieves better scores compared to the benchmark models. We can also see that the backward cycle does indeed improve the quality of the synthesized images. In particular, *GLS* values for the specified factors (chair style and identity) for our model are nearly perfect suggesting that our model manages to accurately alter the specified factor value in the generated images. With respect to unspecified factors of variation, our model yields a high *GLS* value for the illumination condition (0.70) and a low value for head pose (e.g., 1.37 for roll angle). While the achieved scores on viewing orientation ($\theta$, $\phi$) for our model are slightly lower than expected, they are still better than those for the benchmarks. This is likely because our model occasionally fails to precisely construct chairlegs or arms (see Fig. 2a), which provide important cues for recognizing the viewing orientation. It is worth mentioning that the performance differences are less significant on UPNA Synthetic dataset. One possible reason is that it has the maximum number of training samples per class among the three datasets which could benefit the training of the generator.

As can be seen in Figure 2, our model can change a specified factor of variation in an input image, such as face identity or chair style, by adjusting class code $\mathbf{c}$. Meanwhile, the other unspecified factors such as orientation, illumination condition or head pose of the input image are largely preserved in its synthetic version. Overall, images generated by our model are realistic although distortions may occur in image details, e.g., chair legs (see the fifth image in the second row of Fig. 2a). In contrast, the benchmark methods can only combine the specified factors from one source image and the unspecified factors from another source image to generate a new image. Therefore, they have less flexibility to modify a specified factor of variation to a desired value. Images shown in Figs. 2b and 2c are generated by feeding the specified representations from images in the first row, and the unspecified representations from images in the first column to the decoder. The visual quality of corresponding images is inferior to those from our model; blur and distortions are clearly visible. Furthermore, the benchmark methods are less effective in maintaining certain important factors of variation, e.g., color in the synthesized images (see the generated chair images in Figs. 2b and 2c).

The remarkable consistency of the quantitative and qualitative results confirms the effectiveness of the proposed model in creating realistic images with a desired value for the specified factor and the same unspecified traits as the source images.

**Interpolation of synthesis variables:** In order to further evaluate the generative capacity of the proposed model, we conducted additional experiments wherein we linearly interpolate between latent representations and class codes of an initial and a final image in order to obtain a series of new image representations and class codes which are then combined and fed to a trained decoder to synthesize new images. Specifically, let $\mathbf{z}_{\text{initial}}, \mathbf{z}_{\text{final}}$ and $\mathbf{c}_{\text{initial}}, \mathbf{c}_{\text{final}}$ denote, respectively, the learned latent representations and class codes of the initial and a final images and $\mathbf{c}_{\text{interp}} = (1-\alpha_c)\mathbf{c}_{\text{initial}} + \alpha_c \mathbf{c}_{\text{final}}$ and $\mathbf{z}_{\text{interp}} = (1-\alpha_z)\mathbf{z}_{\text{initial}} + \alpha_z \mathbf{z}_{\text{final}}$ their interpolated values, where $\alpha_c, \alpha_z \in [0, 1]$. We synthesize new images by passing $(\mathbf{c}_{\text{interp}}, \mathbf{z}_{\text{interp}})$ into the decoder. Surprisingly, when this is applied to a face dataset, our trained model can generate a sequence of face images that show a seamless transition from one identity into another, i.e., face morphing (3), and also a seamless transition from one value of an unspecified factor (e.g., illumination, pose) into another (columns of Fig. 3). This is despite the fact that the model can only see one-hot codes specifying *discrete* identities during training. In Fig. 3, the class code



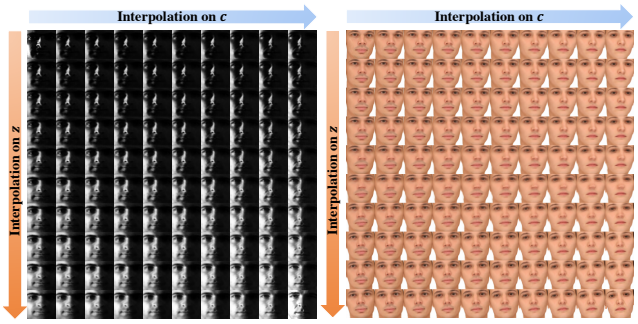

(a) Image synthesis results for the proposed model

(b) Image synthesis results for the model in [20]

(c) Image synthesis results for the model in [21]

Figure 2: Image synthesis by altering the specified factor of variation for 3D Chairs, YaleFace and UPNA Synthetic datasets (from left to right).

is constant within each column while the representation is constant within each rows. We observe that when interpolating $\mathbf{c}$, the unspecified factors such as illumination or head pose are consistent, while the specified factor (identity) changes gradually. In contrast, when interpolating $\mathbf{z}$ the specified factor remains unchanged but the unspecified factors transform continuously.




(a) YaleFace (b) UPNA Synthetic

Figure 3: Linear interpolation results for the proposed model in the latent space ($\mathbf{z}$) and class code space ($\mathbf{c}$). The top-left and bottom-right images are taken from the test set.

## 5. Conclusion

This paper presents a conditional adversarial network for learning an image representation that is invariant to a specified factor of variation, while maintaining unspecified factors. The proposed model does not produce degenerate solutions due to a novel cyclic forward-backward training strategy. Quantitative results from a broad set of experiments show that our model performs better or equally well compared to two state-of-the-art methods in learning invariant image representations. Once trained, our model is also generative as it enables synthesis of a realistic image having a desired value for the specified factor. Both qualitative and quantitative evaluation results confirm that our model can produce better quality images than the competing models.

## 6. Acknowledgement

This work was supported by the NSF under Lighting Enabled Systems and Applications ERC Cooperative Agreement No.EEC-08120256

## References


[1] Mikel Ariz, José J Bengoechea, Arantxa Villanueva, and Rafael Cabeza. A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. *Computer Vision and Image Understanding*, 148:201–210, 2016. 4

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017. 1

[3] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014. 4

[4] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2745–2754, 2017. 3

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1, 2

[6] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010. 2

[7] Jiawei Chen, Janusz Konrad, and Prakash Ishwar. Vgan-based image representation learning for privacy-preserving facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1570–1579, 2018. 2, 3

[8] Jiawei Chen, Jonathan Wu, Janusz Konrad, and Prakash Ishwar. Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 139–147. IEEE, 2017. 2

[9] Jiawei Chen, Jonathan Wu, Kristi Richter, Janusz Konrad, and Prakash Ishwar. Estimating head pose orientation using extremely low resolution images. In *2016 IEEE Southwest symposium on image analysis and interpretation (SSIAI)*, pages 65–68. IEEE, 2016. 2

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 3

[11] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016. 2

[12] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 6

[13] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016. 2

[14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005. 2

[15] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015. 2

[16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[17] Zoubin Ghahramani. Factorial learning and the em algorithm. In *Advances in neural information processing systems*, pages 617–624, 1995. 3

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[20] Naama Hadad, Lior Wolf, , Lior Wolf, and Moni Shahar. A two-step disentanglement method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–780, 2018. 2, 3, 4, 5, 6, 7, 8

[21] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 805–820, 2018. 2, 3, 4, 5, 6, 7, 8

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[23] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013. 2

[24] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3399–3407, 2018. 3

[25] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon, and Nam Ik Cho. Puvae: A variational autoencoder to purify adversarial examples. *IEEE Access*, 7:126582–126593, 2019. 1

[26] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017. 2

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[28] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017. 2

[29] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 3

[30] K.C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005. 4

[31] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[32] Yujia Li, Kevin Swersky, and Richard Zemel. Learning unbiased features. *arXiv preprint arXiv:1412.5244*, 2014. 2

[33] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 1

[34] David G Lowe et al. Object recognition from local scale-invariant features. In *ICCV*, volume 99, pages 1150–1157, 1999. 2

[35] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016. 3

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 6

[37] Stefano Soatto and Alessandro Chiuso. Visual representations: Defining properties and deep approximations. *arXiv preprint arXiv:1411.7676*, 2014. 2

[38] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000. 3

[39] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017. 2, 3

[40] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pages 585–596, 2017. 2

[41] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2015. 2

[42] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 6