Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

# Unpaired Multi-Domain Image Generation via Regularized Conditional GANs

**Xudong Mao** and **Qing Li**
Department of Computer Science, City University of Hong Kong
xudong.xdmao@gmail.com, itqli@cityu.edu.hk

## Abstract

In this paper, we study the problem of multi-domain image generation, the goal of which is to generate pairs of corresponding images from different domains. With the recent development in generative models, image generation has achieved great progress and has been applied to various computer vision tasks. However, multi-domain image generation may not achieve the desired performance due to the difficulty of learning the correspondence of different domain images, especially when the information of paired samples is not given. To tackle this problem, we propose Regularized Conditional GAN (RegCGAN) which is capable of learning to generate corresponding images in the absence of paired training data. RegCGAN is based on the conditional GAN, and we introduce two regularizers to guide the model to learn the corresponding semantics of different domains. We evaluate the proposed model on several tasks for which paired training data is not given, including the generation of edges and photos, the generation of faces with different attributes, etc. The experimental results show that our model can successfully generate corresponding images for all these tasks, while outperforms the baseline methods. We also introduce an approach of applying RegCGAN to unsupervised domain adaptation.

## 1 Introduction

Multi-domain image generation is an important extension of image generation in computer vision. It has many promising applications such as improving the generated image quality [Dosovitskiy *et al.*, 2015; Wang and Gupta, 2016], image-to-image translation [Perarnau *et al.*, 2016; Wang *et al.*, 2017], and unsupervised domain adaptation [Liu and Tuzel, 2016]. As shown in Figures 2 and 3, a successful model for multi-domain image generation should be able to generate pairs of corresponding images which share common semantics but are of different domain-specific semantics. Several early approaches [Dosovitskiy *et al.*, 2015; Wang and Gupta, 2016] have been proposed, but they are all in the supervised setting, which means that they require the information of paired samples to be available. In practice, however, building paired training datasets can be very expensive and may not always be feasible.

Recently, CoGAN [Liu and Tuzel, 2016] has been proposed and achieved great success in multi-domain image generation. In particular, CoGAN models the problem as to learn a joint distribution over multi-domain images by coupling multiple GANs. Unlike previous methods that require paired training data, CoGAN is able to learn the joint distribution without any paired samples. However, it falls short for some difficult tasks such as the generation of edges and photos, as demonstrated by experiments.

In this paper, we propose a new framework called Regularized Conditional GAN (RegCGAN). Like CoGAN, RegCGAN is also capable of performing multi-domain image generation in the absence of paired samples. RegCGAN is based on the conditional GAN [Mirza and Osindero, 2014] and tries to learn a conditional distribution over multi-domain images, where the domain-specific semantics are encoded in the conditioned domain variables, and the common semantics are encoded in the shared latent variables.

As pointed out in [Liu and Tuzel, 2016], directly using conditional GAN will fail to learn the corresponding semantics. To overcome this problem, we propose two regularizers to guide the model to encode the common semantics in the shared latent variables, which in turn makes the model to generate corresponding images. As shown in Figure 1(a)(b), one regularizer is used in the first layer of the generator. This regularizer penalizes the distances between the first layer's output of the paired input, where the paired input should consist of identical latent variables but different domain variables. As a result, it enforces the generator to decode similar high-level semantics for the paired input, since the first layer decodes the highest level semantics. This strategy is based on the fact that corresponding images from different domains always share some high-level semantics (ref. Figures 2, 3, and 5). As shown in Figure 1 (c)(d), the second regularizer is added to the last hidden layer of the discriminator which is responsible for encoding the highest level semantics. This regularizer enforces the discriminator to output similar losses for the pairs of corresponding images. These similar losses are then used to update the generator, which guides the generator to generate similar (corresponding) images.

One intuitive application of RegCGAN is unsupervised do-

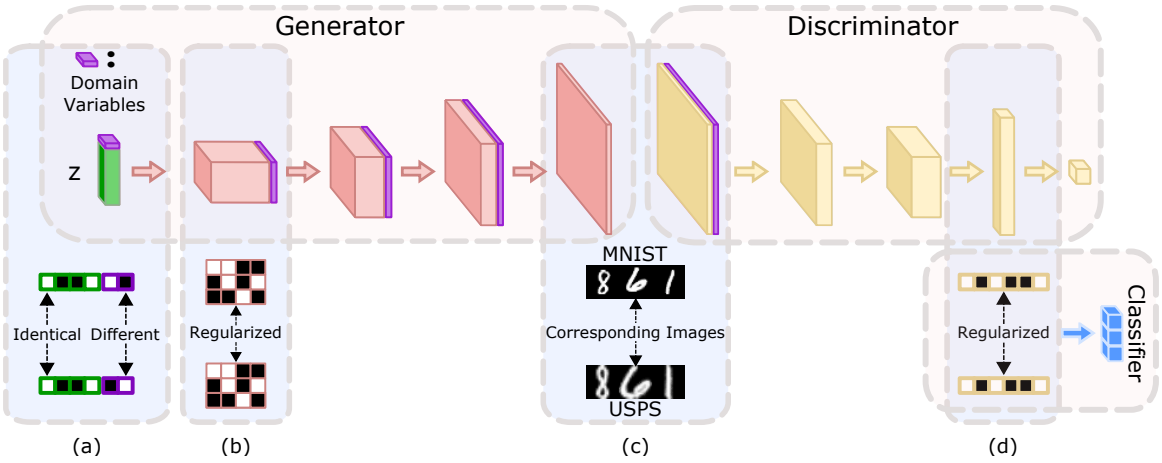



Figure 1: The framework of RegCGAN. The domain variables (in purple) are conditioned to all the layers of the generator and the input image layer of the discriminator. (a): The pairs of input consist of identical latent variables but different domain variables. (b): One regularizer is used in the first layer of the generator. It penalizes the distances between the first layer's output of the input pairs in (a), which guides the generator to decode similar high-level semantics for corresponding images. (c): The generator generates pairs of corresponding images. (d): Another regularizer is used in the last hidden layer of the discriminator. This regularizer enforces the discriminator to output similar losses for the corresponding images. These similar losses are used to update the generator. This regularizer also makes the model output invariant feature representations for the corresponding images from different domains. The learned invariant feature representations can be used for domain adaptation by attaching a classifier.

main adaptation, since the second regularizer (Figure 1(d)) is able to make the last hidden layer to output invariant feature representations for corresponding images (Figure 1(c)). We can attach a classifier to the last hidden layer, and the classifier is jointly trained with the discriminator using the labeled images from the source domain. As a result, the classifier is able to classify the images from the target domain due to the learned invariant feature representations.

## 2 Related Work

### 2.1 Multi-Domain Image Generation

Image generation is one of the most fundamental problems in computer vision. Classic approaches include Restricted Boltzmann Machine [Tieleman, 2008] and Autoencoder [Bengio *et al.*, 2013]. Recently, two successful approaches, Variational Autoencoder (VAE) [Kingma and Welling, 2014] and Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014], have been proposed. Our model in this paper is based on GAN. The idea of GAN is to find the Nash Equilibrium between the generator network and discriminator network. GAN has achieved great success in image generation, and numerous variants [Radford *et al.*, 2015; Nowozin *et al.*, 2016; Arjovsky *et al.*, 2017; Mao *et al.*, 2017] have been proposed for improving the image quality and training stability.

Multi-domain image generation is an extension problem of image generation in which two or more domain images are provided. A successful model should be able to generate pairs of corresponding images, which means that the image pairs share some common semantics but are of different domain-specific semantics. It has many promising applications such as improving the generated image quality [Dosovitskiy *et al.*, 2015; Wang and Gupta, 2016] and image-to-image translation [Perarnau *et al.*, 2016; Wang *et al.*, 2017]. Early approaches [Dosovitskiy *et al.*, 2015; Wang and Gupta, 2016] are under the supervised setting, where the information of paired images is provided. However, building training datasets with paired information is not always feasible and can be very expensive. The recent proposed CoGAN [Liu and Tuzel, 2016] is able to perform multi-domain image generation in the absence of any paired images. CoGAN consists of multiple GANs and each GAN corresponds to one image domain. Furthermore, the weights of some layers are tied to learn the shared semantics.

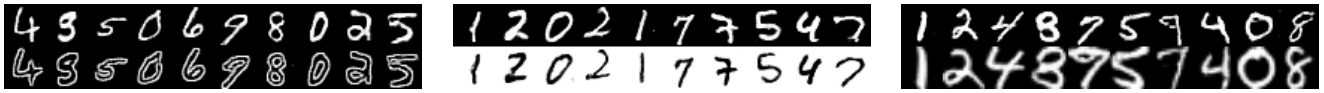
### 2.2 Regularization Methods

Regularization methods have been proven to be effective in GAN learning [Che *et al.*, 2016; Gulrajani *et al.*, 2017; Roth *et al.*, 2017]. Che *et al.* [2016] introduced several types of regularizers which penalize the missing modes. These regularizers are able to relieve the missing modes problem. Gulrajani *et al.* [2017] proposed an effective way of regularizing the gradients of the points sampled between the data distribution and the generator distribution. Moreover, Roth *et al.* [2017] proposed a weighted gradient-based regularizer which can be applied to various GANs. In this paper, we adopt the regularization method for enforcing the model to generate corresponding images.

## 3 Framework and Approach

### 3.1 Generative Adversarial Network

The framework of GAN consists of two roles, the discriminator $D$ and the generator $G$. Given a data distribution $p_{\text{data}}$, $G$ tries to learn the distribution $p_g$ over data $\boldsymbol{x}$. $G$ starts from sampling the noise input $\boldsymbol{z}$ from a simple distribution $p_z(\boldsymbol{z})$, and then maps $\boldsymbol{z}$ to data space $G(\boldsymbol{z}; \theta_g)$. On the other hand,



(a) Digits and edge digits. (b) Digits and negative digits. (c) MNIST and USPS digits.

Figure 2: Generated image pairs on digits.

$D$ aims to distinguish whether a sample is from $p_{\text{data}}$ or from $p_g$. The objective for GAN can be formulated as follows:

$$\min_G \max_D V(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{1}$$

### 3.2 Regularized Conditional GAN

In our approach, the problem of multi-domain image generation is modeled as to learn a conditional distribution $p_{\text{data}}(\boldsymbol{x}|d)$ over data $\boldsymbol{x}$, where $d$ denotes the domain variable. We propose the Regularized Conditional GAN (RegCGAN) for learning $p_{\text{data}}(\boldsymbol{x}|d)$. Our idea is to encode the domain-specific semantics in the domain variable $d$ and to encode the common semantics in the shared latent variables $\boldsymbol{z}$. To achieve this, the conditional GAN is adopted and two regularizers are proposed. One regularizer is added to the first layer of the generator, and the other one is added to the last hidden layer of the discriminator.

Specifically, as Figure 1 shows, for an input pair $(\boldsymbol{z}d_i, \boldsymbol{z}d_j)$ with identical $\boldsymbol{z}$ but different $d$, the first regularizer penalizes the distance between the first layer's output of $\boldsymbol{z}d_i$ and $\boldsymbol{z}d_j$, which enforces $G$ to decode similar high-level semantics, since the first layer decodes the highest level semantics. On the other hand, for a pair of corresponding images $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the second regularizer penalizes the distance between the last layer's output of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. As a result, $D$ outputs similar losses for the pairs of corresponding images. When updating $G$, these similar losses guide $G$ to generate similar (corresponding) images. Note that to use the above two regularizers, it requires constructing pairs of input which are of identical $\boldsymbol{z}$ but different $d$.

Formally, when training, we construct mini-batches with pairs of input $(\boldsymbol{z}, d = 0)$ and $(\boldsymbol{z}, d = 1)$, where the noise input $\boldsymbol{z}$ is the same. $G$ maps the noise input $\boldsymbol{z}$ to a conditional data space $G(\boldsymbol{z}|d)$. An L2-norm regularizer is used to enforce $G_{h_0}(\boldsymbol{z}|d)$, the output of $G$'s first layer, to be similar for each paired input. Another L2-norm regularizer is used to enforce $D_{h_i}(G(\boldsymbol{z}|d))$, the output of $D$'s last hidden layer, to be similar for each paired input. Then the objective function for RegCGAN can be formulated as follows:

$$\begin{aligned}
&\min_G \max_D V(G,D) = \mathcal{L}_{\text{GAN}}(G,D) + \lambda\mathcal{L}_{\text{reg}}(G) + \beta\mathcal{L}_{\text{reg}}(D),\\
&\mathcal{L}_{\text{GAN}}(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x}|d)}[\log D(\boldsymbol{x}|d)] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|d)))],\\
&\mathcal{L}_{\text{reg}}(G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\|G_{h_0}(\boldsymbol{z}|d=0) - G_{h_0}(\boldsymbol{z}|d=1)\|^2],\\
&\mathcal{L}_{\text{reg}}(D) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[-\|D_{h_i}(G(\boldsymbol{z}|d=0)) - D_{h_i}(G(\boldsymbol{z}|d=1))\|^2],
\end{aligned} \tag{2}$$

where the scalars $\lambda$ and $\beta$ are used to adjust the weights of the regularization terms, $\|\cdot\|$ denotes the $l^2$-norm, $d = 0$ and $d = 1$ denote the source domain and target domain, respectively, $G_{h_0}(\cdot)$ denotes the output of $G$'s first layer, and $D_{h_i}(\cdot)$ denotes the output of $D$'s last hidden layer.

As stated before, RegCGAN can be applied to unsupervised domain adaptation, since the last hidden layer of $D$ is able to output invariant feature representations for the corresponding images from different domains. Based on the invariant feature representations, we attach a classifier to the last hidden layer of $D$. The classifier is jointly trained with $D$, and the joint objective function is:

$$\min_{G,C} \max_D V(G,D,C) = \mathcal{L}_{\text{GAN}}(G,D) + \lambda\mathcal{L}_{\text{reg}}(G) + \beta\mathcal{L}_{\text{reg}}(D) + \gamma\mathcal{L}_{\text{cls}}(C), \tag{3}$$

where the scalars $\lambda$, $\beta$, and $\gamma$ are used to adjust the weights of the regularization terms and the classifier, and $\mathcal{L}_{\text{cls}}(C)$ is a typical cross-entropy loss.

Note that our approach to domain adaptation is different from the method used in [Ganin *et al.*, 2016] which tries to minimize the difference between the overall distribution of the source and target domains. In contrast, the minimization of our approach is among samples belonging to the same category, because we only penalize the distances between the pairs of corresponding images which belong to the same category.

## 4 Experiments

### 4.1 Implementation Details

Except for the tasks about digits (i.e., MNIST and USPS), we adopt LSGAN [Mao *et al.*, 2017] for training the models due to the fact that LSGAN generates higher quality images and perform more stably. For digits tasks we still adopt standard GAN since we find that LSGAN will sometimes generate unaligned digit pairs.

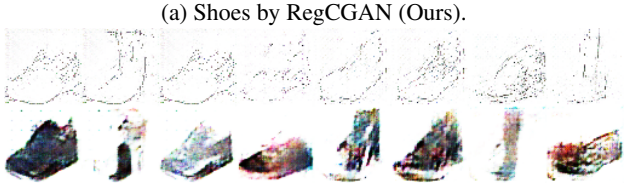
We use Adam optimizer with the learning rates of $0.0005$ for LSGAN and $0.0002$ for standard GAN. For the hyper-parameters in Equations 2 and 3, we set $\lambda = 0.1$, $\beta = 0.004$, and $\gamma = 1.0$ found by grid search. Our implementation is available at https://github.com/xudonmao/RegCGAN.

### 4.2 Digits

We first evaluate RegCGAN on MNIST and USPS datasets. Since the image sizes of MNIST and USPS are different, we resize the images in USPS to the same resolution (i.e., $28 \times 28$) of MNIST. We train RegCGAN for the following three tasks. Following literature [Liu and Tuzel, 2016], the first two tasks are to perform the generations of 1) digits and edge digits; 2) digits and negative digits. The third one is

(a) Shoes by RegCGAN (Ours).

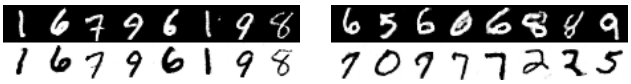
(b) Handbags by RegCGAN (Ours).



(c) Shoes by CoGAN.

(d) Handbags by CoGAN.

Figure 3: Generated image pairs on shoes and handbags.



(a) With regularizer. (b) Without regularizer.

Figure 4: Comparison experiments between the models with and without the regularizer.

to perform the generation of MNIST and USPS digits. For these tasks, we design the network architecture following the suggestions in [Radford *et al.*, 2015], where the generator consists of four transposed convolutional layers and the discriminator is a variant of LeNet [Lecun *et al.*, 1998]. The generated image pairs are shown in Figure 2, where we can see clearly that RegCGAN succeeds to generate corresponding digits for all the three tasks.

**Without Regularizer** If we remove the proposed regularizers in RegCGAN, the model will fail to generate corresponding digits as Figure 4 shows. This demonstrates that the proposed regularizers play an important role in generating corresponding images.

### 4.3 Edges and Photos

We also train RegCGAN for the task of generating corresponding edges and photos. The Handbag [Zhu *et al.*, 2016] and Shoe [Yu and Grauman, 2014] datasets are used for this task. We randomly shuffle the edge images and realistic photos to avoid utilizing the pair information. We resize all the images to a resolution of $64 \times 64$. For the network architecture, both the generator and the discriminator consist of four transposed/strided-convolutional convolutional layers. As shown in Figure 3(a)(b), RegCGAN is able to generate corresponding images of edges and photos.

**Comparison with CoGAN** We also train CoGAN, which is the current state-of-the-art method, on edges and photos using the official implementation of CoGAN. We evaluate two network architectures for CoGAN: (1) the architecture used in CoGAN [Liu and Tuzel, 2016] and (2) the same architecture to RegCGAN. We also evaluate the standard GAN loss and least squares loss (LSGAN) for CoGAN. But all of these settings fail to generate corresponding images of edges and photos. The results are shown in Figure 3(c)(d).

### 4.4 Faces

In this task, we evaluate RegCGAN on the CelebA dataset [Liu *et al.*, 2014]. We first apply a pre-processing method to crop the facial region in the center of the images [Karras *et al.*, 2017], and then resize all the cropped images to a resolution of $112 \times 112$. The network architecture used in this task is similar to the one in Section 4.3 except for the output dimensions of the layers. We investigate the following two tasks: 1) female with blond and black hair; and 2) female and male. The results are presented in Figure 5(a)(b). We observe that RegCGAN is able to generate corresponding face images with different attributes, and the corresponding faces are of very similar appearances.

**Comparison with CoGAN** The generated image pairs by CoGAN are also presented in Figure 5, where the image pairs of black and blond hair by CoGAN are duplicated from [Liu and Tuzel, 2016]. We observe that the image pairs generated by RegCGAN are more consistent and of better quality than the ones by CoGAN, especially for the task of female and male, which is more difficult than the task of blond and black hair.

**Comparison with CycleGAN** We also compare RegCGAN with CycleGAN [Zhu *et al.*, 2017] which is the state-of-the-art method in image-to-image transition. To compare with CycleGAN, we first generate some image pairs using RegCGAN and then use the generated images in one domain as the input for CycleGAN. The results are presented in Figure 6. Compared with RegCGAN, CycleGAN introduces some blur to the generated images. Moreover, the color of the image pairs by RegCGAN is more consistent than the ones by CycleGAN.

### 4.5 Quantitative Evaluation

To further evaluate the effectiveness of RegCGAN, we conduct a user study on Amazon Mechanical Turk (AMT). For this evaluation, we also use the task of the female and male generation. In particular, given two image pairs randomly selected from RegCGAN and CoGAN, the AMT annotators are asked to choose a better one based on the image quality, perceptual realism, and appearance consistency of female and male. With $3,000$ votes totally, a majority of the annotators preferred the image pairs from RegCGAN in $77.6\%$, demon-

(a) Blond and black hair by RegCGAN (Ours).

(b) Female and male by RegCGAN (Ours).

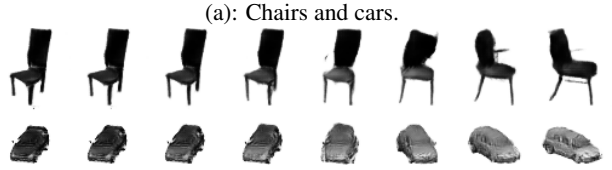
(c) Blond and black hair by CoGAN.

(d) Female and male by CoGAN.

Figure 5: Generated image pairs on faces with different attributes. The image pairs of black and blond hair by CoGAN are duplicated from the CoGAN paper.


Figure 6: Comparison results between RegCGAN and CycleGAN for the task of female and male. The top two rows are generated by RegCGAN. The third row is generated by CycleGAN using the first row as input.

(a): Chairs and cars.



(b): Interpolation between two random points in noise input.

Figure 7: Generated image pairs on chairs and cars, where the orientations are highly correlated.

| | CoGAN | RegCGAN (Ours) |
|---|---|---|
| User Choice | 673 / 3000 (22.4%) | **2327** / 3000 (**77.6%**) |

Table 1: A user study on the task of female and male generation. With $3,000$ votes totally, $77.6\%$ of the annotators preferred the image pairs from RegCGAN.

strating that the overall image quality of our model is better than the one of CoGAN.

### 4.6 More Applications

**Chairs and Cars** In this task, we use two visually completely different datasets, Chairs [Aubry *et al.*, 2014] and Cars [Fidler *et al.*, 2012]. Both datasets contain synthesized samples with different orientations. We train RegCGAN on these two datasets to study whether it is able to generate corresponding images sharing the same orientations. The generated results are shown in Figure 7, where the image resolution is $64 \times 64$. We further perform interpolation between two random points in the latent space as shown in Figure 7(b). The interpolation shows smooth transitions of chairs and cars both in viewpoint and style, while the chairs and cars keep facing the same direction.

**Photos and Depths** The NYU depth dataset [Silberman *et al.*, 2012] is used for learning a RegCGAN over photos and depth images. In this task, we first resize all the images to a resolution of $120 \times 160$ and then randomly crop $112 \times 112$ patches for training. Figure 8 shows the generated image pairs.

**Photos and Monet-Style Images** In this task we train RegCGAN on the Monet-style dataset [Zhu *et al.*, 2017]. We use the same pre-processing method as in Section 4.6. Figure 9 shows the generated image pairs.

Figure 8: Generated image pairs on photos and depth images.

Figure 9: Generated image pairs on photos and Monet-style images.

**Summer and Winter** We also train RegCGAN on the Summer and Winter dataset [Zhu *et al.*, 2017]. We use the same pre-processing method as in Section 4.6. Figure 10 shows the generated image pairs.

### 4.7 Unsupervised Domain Adaptation

As mentioned in Section 3.2, RegCGAN can be applied to unsupervised domain adaptation. In this experiment, MNIST and USPS datasets are used, where one is used as the source domain and the other one is used as the target domain. We set $\lambda = 0.1$, $\beta = 0.004$, and $\gamma = 1.0$ found by grid search. We use the same network architecture as in Section 4.2 and attach a classifier at the end of the last hidden layer of the discriminator. Following the experiment protocol in [Liu and Tuzel, 2016; Tzeng *et al.*, 2017], we randomly sample $2,000$ images from MNIST and $1,800$ images from USPS.

We conduct two comparison experiments between RegCGAN and the baseline methods, including DANN [Ganin *et al.*, 2016], ADDA [Tzeng *et al.*, 2017], and CoGAN [Liu and Tuzel, 2016]. One is to evaluate the classification accuracy directly on the sampled images of the target domain, which is adopted in [Liu and Tuzel, 2016; Tzeng *et al.*, 2017]. To further evaluate the generalization error, we further evaluate the classification accuracy on the standard test sets of the target domain.

The results are presented in Table 2. The reported accuracies are averaged over 10 trails with different random samplings. For the evaluation on the standard test set, RegCGAN significantly outperforms all the baseline methods, especially for the task of USPS to MNIST. This shows that RegCGAN is of smaller generalization error when compared with the baseline methods. For the evaluation on the sampled set, RegCGAN outperforms all the baseline methods for the task of MNIST to USPS, and achieves comparable performance for the task of USPS to MNIST.

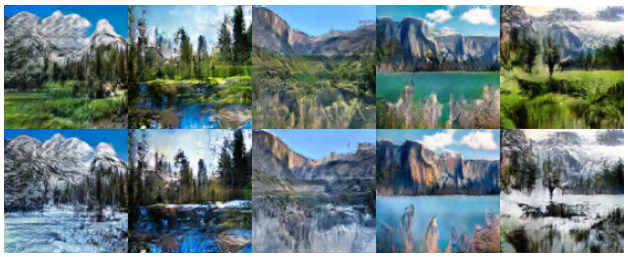


Figure 10: Generated image pairs on summer Yosemite and winter Yosemite.

| Method | MNIST→USPS | USPS→MNIST |
|---|---|---|
| | Evaluated on the sampled set | |
| DANN | 0.771 | 0.730 |
| ADDA | $0.894 \pm 0.002$ | **0.901** $\pm 0.008$ |
| CoGAN | $0.912 \pm 0.008$ | $0.891 \pm 0.008$ |
| RegCGAN (Ours) | **0.931** $\pm 0.007$ | $0.895 \pm 0.009$ |
| | Evaluated on the test set | |
| ADDA | $0.836 \pm 0.035$ | $0.849 \pm 0.058$ |
| CoGAN | $0.882 \pm 0.018$ | $0.822 \pm 0.081$ |
| RegCGAN (Ours) | **0.901** $\pm 0.009$ | **0.888** $\pm 0.015$ |

Table 2: Accuracy results for unsupervised domain adaptation. The top section presents the classification accuracy evaluated on the sampled set of the target domain. The bottom section presents the classification accuracy evaluated on the standard test set of the target domain. The reported accuracies are averaged over 10 trails with different random samplings.

## 5 Conclusions

To tackle the problem of multi-domain image generation, we have proposed the Regularized Conditional GAN, where the domain information is encoded in the conditioned domain variables. Two types of regularizers are proposed. One is added to the first layer of the generator, guiding the generator to decode similar high-level semantics. The other one is added to the last hidden layer of the discriminator, enforcing the discriminator to output similar losses for the corresponding images. Various experiments on multi-domain image generation have been conducted. The experimental results show that RegCGAN succeeds to generate pairs of corresponding images for all these tasks, and outperforms all the baseline methods. We have also introduced a method of applying RegCGAN to domain adaptation.

## Acknowledgments

This work is supported by a research grant (project number: 9360153) and a special grant (account number: 9610367) from City University of Hong Kong, and a grant (No. 2016A010101012) from Science and Technology Program of Guangdong Province, China.

## References


[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv:1701.07875*, 2017.

[Aubry *et al.*, 2014] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Bengio *et al.*, 2013] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. *arXiv:1305.6663*, 2013.

[Che *et al.*, 2016] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv:1612.02136*, 2016.

[Dosovitskiy *et al.*, 2015] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Fidler *et al.*, 2012] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.

[Goodfellow *et al.*, 2014] 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 (NIPS)*, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[Karras *et al.*, 2017] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv:1710.10196*, 2017.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[Lecun *et al.*, 1998] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[Liu and Tuzel, 2016] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[Liu *et al.*, 2014] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *arXiv:1411.7766*, 2014.

[Mao *et al.*, 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.

[Nowozin *et al.*, 2016] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv:1606.00709*, 2016.

[Perarnau *et al.*, 2016] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. *arXiv:1611.06355*, 2016.

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

[Roth *et al.*, 2017] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *arXiv:1705.09367*, 2017.

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012.

[Tieleman, 2008] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *International Conference on Machine Learning (ICML)*, 2008.

[Tzeng *et al.*, 2017] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Wang and Gupta, 2016] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[Wang *et al.*, 2017] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao. Tag disentangled generative adversarial networks for object image re-rendering. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[Yu and Grauman, 2014] Aron Yu and Kristen Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Zhu *et al.*, 2016] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.