Geometry-Consistent Adversarial Networks for One-Sided Unsupervised Domain Mapping

Huan Fu* 1 Mingming Gong* 2,3
Chaohui Wang4 Kayhan Batmanghelich2,3 Kun Zhang3 Dacheng Tao1

1UBTECH Sydney AI Centre, SIT, FEIT, The University of Sydney, Australia
2Department of Biomedical Informatics, University of Pittsburgh
3Department of Philosophy, Carnegie Mellon University
4Laboratoire d’Informatique Gaspard Monge, Université Paris-Est
{hufu6371@uni., dacheng.tao@sydney.edu.au \{mig73, kayhan\}@pitt.edu
chaohui.wang@u-pem.fr kunz1@cmu.edu

Abstract

Unsupervised domain mapping aims at learning a function to translate domain X to Y (G_{XY} : X \rightarrow Y) in the absence of paired (X, Y) samples. Finding the optimal G_{XY} without paired data is an ill-posed problem and hence appropriate constraints are required to obtain reasonable solutions. One of the most prominent constraint is cycle-consistency, which enforces the translated image by G_{XY} to be translated back to the input image by an inverse mapping G_{YX}. While cycle-consistency requires simultaneous training of G_{XY} and G_{YX}, recent methods have demonstrated one-sided domain mapping (only learn G_{XY}) can be achieved by preserving pairwise distance between images before and after translation. Although cycle-consistency and distance preserving successfully constrain the solution space, they overlook the special properties of images that simple geometric transformations do not change the semantics of an image. Based on this special property, we develop a geometry-consistent adversarial network (GcGAN) which enables one-sided unsupervised domain mapping. Our GcGAN takes the original image and its counterpart image transformed by a predefined geometric transformation as inputs and generates two images in the new domain with the corresponding geometry-consistency constraint. The geometry-consistency constraint eliminates unreasonable solutions and produce more reliable solutions. Quantitative comparisons against baseline (GAN alone) and the state-of-the-art methods, including DistanceGAN and CycleGAN, demonstrate the superiority of our method in generating realistic images.

1 Introduction

Domain mapping or image-to-image translation, which aims at translating an image from one domain to a target domain, has attracted lots of attention in recent years. Various computer vision tasks can be recast as a standard domain mapping task. For instance, scene parsing can be considered as a problem of mapping an natural image to a corresponding a category-level color image. Furthermore, with an ideal translator, effects for model adaption is no longer that important in scene understanding, since we can flexibly translate images between domains.

According to that if the image pairs \{(x_i, y_i)\}_{i=1}^{N} are available, domain mapping can be studied in supervised or unsupervised manners. While different works have focused on supervised domain mapping with the constraints from provided cross-domain image pairs, and produced some powerful translators [1,2,3,4], unsupervised domain mapping suffers from slow progress, is even in the initial stage. However, obtaining paired training examples is difficult and expensive. For example, if we want to learn some translators between Male and Female, how can we collect enough well-defined
Addressing this problem, recent approaches explore some supervision at domain level instead of paired deep convolutional generative adversarial networks (DCGANs), such as image inpainting, text to images from inputs. Recently, many applications and computer vision tasks are developed based on Generative Adversarial Networks (GANs), which can be easily perceived by humans. In this paper, we develop a geometry consistency constraint and formulate a geometric-consistent adversarial network (GcGAN) for unsupervised domain mapping. Our constraint is motivated by a reasonable assumption that a given geometric transformation \( f(\cdot) \) should be preserved by related translaters \( G_{XY} \) and \( \tilde{G}_{\tilde{X}\tilde{Y}} \), if \( \tilde{X} \) and \( \tilde{Y} \) are the domains obtained by applying \( f(\cdot) \) on the sample of \( X \) and \( Y \) respectively, as illustrated in Fig. 1. Mathematically, given a random sample \( x \) from the source domain \( X \) and a predefined geometric transformation function \( f(\cdot) \), the geometry consistency can be expressed as \( f(G_{XY}(x)) \approx (\tilde{G}_{\tilde{X}\tilde{Y}}(f(x))) \) and \( f^{-1}(G_{XY}(x)) \approx \tilde{G}_{\tilde{X}\tilde{Y}}(f(x)) \), where \( f^{-1}(\cdot) \) is the inverse transformation of \( f(\cdot) \) satisfied \( f^{-1}(f(x)) = f(f^{-1}(x)) = x \). Our geometry consistency constraint allows one-sided unsupervised domain mapping, i.e. \( G_{XY} \) can be trained independent with \( G_{YX} \). From an intuition, the geometry consistency constraint can work well because it is able to correct some failure cases in some local regions of each other’s translations. In our experiments, we take two simple but representative geometric transformations as examples, i.e. vertical flip (vf) and 90 degrees rotation (rot), to discuss our geometry-consistent constraint. Quantitative comparisons against baseline (GAN alone) and the state-of-the-art methods, including DistanceGAN and CycleGAN, demonstrates the superiority of our model in generating realistic images.

2 Related Work

Generative Adversarial Networks. Generative adversarial networks (GANs) [10][11][12][13][14][15] learn two networks, i.e. a generator and a discriminator, in a staging zero-sum game fashion to generate images from inputs. Recently, many applications and computer vision tasks are developed based on deep convolutional generative adversarial networks (DCGANs), such as image inpainting, text to
Wolf \cite{9} indicate that maintaining the distance between samples within domains allows one-sided mapping to be effective. While great progress is achieved in supervised domain mapping, many real-world applications can benefit from unsupervised domain mapping. Wang \cite{1} propose a novel feature matching loss, as well as multi-scale generator and discriminator architectures. Works is conditional GAN \cite{3}, which suggests to learn the discriminator to distinguish images as distinguishing $G_X$ from $G_Y$. Thus, different constraints and frameworks have been proposed for image-to-image translation in the absence of training pairs \cite{19, 5, 9}, i.e. unsupervised domain mapping. Unpaired domain mapping does have a long history, and obtain some successes in adversarial networks recently. For example, Liu and Tuzel \cite{19} introduce coupled generative adversarial network (CoGAN) to learn cross-domain representation by enforcing a weight-sharing constraint. Subsequently, CycleGAN \cite{5}, DiscoGAN \cite{7} and DualGAN \cite{8} present that translators $G_{XY}$ and $G_{YX}$ should be bijections. Thus, jointly learning $G_{XY}$ and $G_{YX}$ by enforcing cycle consistency could be effective in unsupervised domain mapping. Since then, many constraints and assumptions have been proposed to improve cycle consistency. Recently, Benaim and Wolf \cite{9} indicate that maintaining the distance between samples within domains allows one-sided unsupervised domain mapping, which is another important work. Our GcGAN is also an one-sided framework based on our geometry consistency assumption, and can even obtain more impressive translation than the two-sided CycleGAN.
3 Preliminaries

Let $X$ and $Y$ be two domains with unpaired training samples $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^M$, where $x_i$ and $y_j$ are drawn from the marginal distributions $P_X$ and $P_Y$. Our goal here lies in two aspects. On the one hand, we have to learn the mapping $G_{XY}$ so that $\{G_{XY}(x)\}_{i=1}^N$ has the same distribution with $\{y_j\}_{j=1}^M$, i.e. $P_{G_{XY}(X)} \approx P_Y$. On the other hand, we must ensure that the learned mapping function can map an individual input to a desired output.

While many works model the invertibility between $G_{XY}$ and $G_{YX}$ for convincing mappings since the success of CycleGAN, we propose to enforce geometry consistency as a constraint which allows one-sided domain mapping, i.e. learning $G_{XY}$ without simultaneously learning $G_{YX}$. Let $f(\cdot)$ be a predefined geometric transformation. We can obtain two extra domains $\tilde{X}$ and $\tilde{Y}$ with samples $\{\tilde{x}_i\}_{i=1}^N$ and $\{\tilde{y}_j\}_{j=1}^M$ by applying $f(\cdot)$ on all the samples of $X$ and $Y$ respectively. We learn an additional image-to-image translator $G_{\tilde{X}\tilde{Y}} : \tilde{X} \rightarrow \tilde{Y}$ while optimizing $G_{XY} : X \rightarrow Y$, and introduce our geometry consistency constraint based on the transformation so that the two networks can constrain and benefit each other. Our framework implies a reasonable geometry consistency assumption that $G_{XY}(x)$ and $G_{X\tilde{Y}}(\tilde{x})$ should keep the same geometric transformation with the one between $x$ and $\tilde{x}$, i.e. $f(G_{XY}(x)) \approx G_{XY}(G_{X\tilde{Y}}(\tilde{x}))$, where $\tilde{x} = f(x)$. In the following, we denote the two adversarial discriminators as $D_Y$ and $D_{\tilde{Y}}$ respecting to the domains $Y$ and $\tilde{Y}$ respectively.

4 Formulation

We present our geometry consistency constraint as well as our GcGAN beginning with a review of the cycle consistency constraint and the distance constraint. An illustration of the main differences between these constraints is shown in Fig. 2.

4.1 Unsupervised Constraints

**Cycle consistency constraint.** Following the cycle consistency assumption [7, 5, 8], through the translators $G_{XY} \circ G_{YX} : X \rightarrow Y \rightarrow X \setminus G_{XY} \circ G_{XY} : Y \rightarrow X \rightarrow Y$, a sample $x$ in domain $X$ and $y$ in domain $Y$ should be brought back to the original image, i.e. $x \approx G_{XY}(G_{XY}(x)) \setminus y \approx G_{YX}(G_{YX}(y))$. The cycle consistency constraint is implemented by a bidirectional reconstruction process which requires $G_{XY}$ and $G_{YX}$ to be jointly learned, as shown in Fig. 2 (CycleGAN). The cycle consistency loss $L_{cyc}(G_{XY}, G_{YX}, X, Y)$ takes the form as:

$$
L_{cyc}(G_{XY}, G_{YX}, X, Y) = \mathbb{E}_{x \sim P_X} \left[ \| G_{XY}(G_{XY}(x)) - x \|_1 \right] + \mathbb{E}_{y \sim P_Y} \left[ \| G_{YX}(G_{YX}(y)) - y \|_1 \right].
$$

(1)

In practice, the $L1$ norm in this loss is more preferable than $L2$ norm.

**Distance constraint.** The assumption behind the distance constraint is that the distance between two samples $x_i$ and $x_j$ in domain $X$ should be persevered after mapping to domain $Y$, i.e. $d((x_i, x_j)) \approx d(G_{XY}(x_i), G_{XY}(x_j))$, where $d(\cdot)$ is a predefined function to measure the distance between two samples. In DistanceGAN [9], the distance consistency loss $L_{dis}(G_{XY}, X, Y)$ is the exception of the absolute differences between distances in each domain:

$$
L_{dis}(G_{XY}, X, Y) = \mathbb{E}_{x_i, x_j \sim P_X} \left[ \| \phi(x_i, x_j, X) - \psi(x_i, x_j, Y, G_{XY}) \| ight],
$$

$$
\phi(x_i, x_j, X) = \frac{1}{\sigma_X} (\|x_i - x_j\|_1 - \mu_X),
$$

$$
\psi(x_i, x_j, Y, G_{XY}) = \frac{1}{\sigma_Y} (\|G_{XY}(x_i) - G_{XY}(x_j)\|_1 - \mu_Y),
$$

(2)

where $\sigma_X, \sigma_Y, (\mu_X, \mu_Y)$ are the means (standard deviations) of $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^M$ respectively, and are precomputed. The distance constraint makes one-sided unsupervised domain mapping possible, but suffers from stability.

4.2 Geometry-consistent Adversarial Networks

**Adversarial constraint.** Taking $G_{XY}$ as an example, an adversarial loss $L_{gan}(G_{XY}, D_Y, X, Y)$ [10] enforces $G_{XY}$ and $D_Y$ simultaneously optimizing each other in a minimax game,
We apply our GcGAN on a wide range of applications, and make both quantitative and qualitative comparisons with baseline (GAN alone) and previous state-of-the-art methods, including DistanceGAN and CycleGAN. We also study different ablations to analyze our geometry consistency constraint. Since adversarial networks are always not that stable, every independent experiments could result some slightly different scores. The scores in quantitative comparisons are computed by averagely voting on three independent experiments for our GcGAN.

5 Experiments

We apply our GcGAN on a wide range of applications, and make both quantitative and qualitative comparisons with baseline (GAN alone) and previous state-of-the-art methods, including DistanceGAN and CycleGAN. We also study different ablations to analyze our geometry consistency constraint. Since adversarial networks are always not that stable, every independent experiments could result some slightly different scores. The scores in quantitative comparisons are computed by averagely voting on three independent experiments for our GcGAN.
Figure 3: Qualitative comparisons on Cityscapes (Parsing $\leftrightarrow$ Image), Google Maps (Map $\leftrightarrow$ Aerial photo), and Horse $\leftrightarrow$ Zebra from top to bottom.

Figure 4: Qualitative results for MNIST $\rightarrow$ SVHN, Summer $\leftrightarrow$ Winter, and Synthetic $\leftrightarrow$ Real.
5.1 Quantitative Results

All the scores below demonstrate that our geometry consistency constraint can not only improve other translators, but also drive one-sided unsupervised domain mapping to a higher stage.

**Cityscapes.** Cityscapes [27] contains 3975 image-label pairs, where 2975 for training and 500 for validation (test in our paper). To make a fair comparison against CycleGAN, the translators are trained in the resolution of $128 \times 128$ in unpaired fashion. We evaluate our domain mappers via FCN scores and scene parsing metrics following previous works [28] [27]. In specific, for parsing $\rightarrow$ image, we suppose that a high-quality fake image should produce qualitative semantic segmentation like real images when feeding it into a scene parser. Thus, we adopt the pretrained FCN parser [28] provided by pix2pix [3] to predict semantic labels for the 500 fake images. The label maps are then resized to the original resolution ($1024 \times 2048$), and compared against the ground truth labels using some standard scene parsing metrics including pixel accuracy, mean accuracy, and mean IoU [28]. For image $\rightarrow$ parsing, since the fake labels are in RGB format, we simply convert them to class-level labels via the nearest search strategy. Specifically, we have 19 (category labels) + 1 (ignored label) categories for Cityscapes, with a corresponding color value (RGB) respectively. For a pixel $i$ in a fake parsing image, we compute the distances between the 20 ground-truth color values and the color value of pixel $i$. The label of pixel $i$ should be the one with the minimal distance. Then, aforementioned metrics are used to evaluate our mapping in the resolution of $128 \times 128$ on the 19 category labels.

![Table 1: Parsing scores on Cityscapes.](image)

We report the parsing scores in Tab. 1 for both image $\rightarrow$ parsing and parsing $\rightarrow$ image tasks. Our GcGAN outperforms baseline (GAN alone) in a large margin. We take the mean of pixel accuracy, mean accuracy, and mean IoU as the final score for analyzation convenience, i.e. score $= (\text{pixel acc} + \text{mean acc} + \text{mean IoU}) / 3$. For image $\rightarrow$ parsing, our GcGAN (33.5% $\sim$ 33.9%) obtains a 1.5% $\sim$ 1.9% improvement compared with CycleGAN (32%). For parsing $\rightarrow$ image, our GcGAN also (27.9% $\sim$ 28.8%) yield higher scores than CycleGAN (26.6%).

We run an ablation that directly merges our geometry consistency constraint into CycleGAN. The default trade-off parameter for $L_{\text{cyc}}$ is 10.0 in their released code. Our parameter $\lambda$ for $L_{\text{geo}}$ is still 20.0 as aforementioned. We find that combining $L_{\text{cyc}}$ and $L_{\text{geo}}$ obtain slightly higher scores for image $\rightarrow$ parsing, but decrease the performance for parsing $\rightarrow$ image. One possible reason is that adversarial networks are not that stable themselves, and the trade-off parameters for $L_{\text{gan}}, L_{\text{geo}},$ and $L_{\text{cyc}}$ should be carefully treated.

**SVHN $\rightarrow$ MNIST.** We apply our approach for the SVHN $\rightarrow$ MNIST translation task. The $G_{\text{mnist}}$ is trained on 531131 \ 60000 training images with resolution of 32 $\times$ 32 in SVHN and MNIST training sets respectively. The experimental setting follows DistanceGAN, including the default trade-off parameters for $L_{\text{cyc}}$ and $L_{\text{geo}}$, and the network architectures for the generators and the discriminators. We take both DistanceGAN and CycleGAN as baselines on this translation task. In order to obtain numerical results, we feed the translated images into a pretrained classifier which is trained on MNIST training split. Note that, the experimental setting for domain mapping (GcGAN) and domain adaptation is totally different, so as the obtained classification accuracy.
Table 2: Quantitative scores for SVHN → MNIST.

| Benchmark Performance | Ablation Studies (rot) |
|-----------------------|------------------------|
| method                | class acc (%)          | method                        | class acc (%) |
| DistanceGAN (Dist.)   | 26.8                   | Cycle + Dist. [9]             | 18.0          |
| CycleGAN (Cycle)      | 26.1                   | GcGAN + Dist.                 | 30.3          |
| Self-Distance [9]     | 25.2                   | GcGAN + Cycle                | 32.6          |
| GcGAN-rot             | 35.8                   | GcGAN + Dist. + Cycle        | 32.8          |
| GcGAN-vf              | 35.4                   | GcGAN                        | 35.8          |

The classification accuracy is reported in Tab. 2. Both our GcGAN-rot and GcGAN-vf outperform DistanceGAN and CycleGAN in a large margin (about 8% ~ 9%). From the ablations, adding our geometry consistency constraint to current unsupervised domain mapping frameworks will achieve different levels of improvements against the original ones. However, our GcGAN along still yield the highest score. We have given an explanation before for this observation.

Table 3: Quantitative scores for Google Maps.

| Map → Aerial photo | Aerial photo → Map |
|--------------------|--------------------|
| method             | RMSE               | pixel acc (δ1) | pixel acc (δ2) |
| CycleGAN [5]       | 75.46              | 40.8           | 62.6           |
| GAN alone (baseline) | 79.63             | 12.7           | 43.3           |
| GcGAN-rot          | 76.38              | 35.6           | 56.8           |
| GcGAN-vf           | 76.49              | 37.6           | 57.7           |
| GcGAN + Cycle      | 74.94              | 42.7           | 65.8           |

Google Maps. We have 2194 (map, aerial photo) pairs which are scraped from Google Maps [3] in and around New York City, and split into training and test sets with 1096 and 1098 pairs respectively. We train Map ← Aerial photo translators with the image size of 256 × 256 using the training set in unsupervised manner (unpaired). For Aerial photo → Map, we make a comparison with CycleGAN via average RMSE and pixel accuracy (%). Given a pixel \( i \) with the ground-truth RGB value \((r_i, g_i, b_i)\) and the predicted RGB value \((r'_i, g'_i, b'_i)\), if \(\max(|r_i - r'_i|, |g_i - g'_i|, |b_i - b'_i|) < \delta\), we think this is an accurate prediction. Since maps only contain several different RGB values, it’s reasonable to compute pixel accuracy following this strategy (\(\delta_1 = 5\) and \(\delta_2 = 10\) in this paper). For Map → Aerial photo, we only report average RMSE.

From the scores in Tab. 3 our GcGAN can produce superior translations than baseline (GAN alone) based on RMSE (76.49 \(\times\) 76.58 vs. 70.63 and 31.03 \(\times\) 30.44 vs. 33.72). Especially, our GcGAN yield a 22.9% ~ 24.9% improvement against baseline in pixel accuracy with \(\delta = 5.0\), which demonstrates that the fake maps obtained by our GcGAN contain more details. Though the two-sided CycleGAN performs better than our one-sided GcGAN, our geometry consistency constraint can further improve it (GcGAN + Cycle vs. CycleGAN).

5.2 Qualitative Results

All the qualitative results are shown in Fig. 3 and Fig. 4. Our GcGAN can realize empirically impressive translations for various applications.

Horse ← Zebra. We apply our GcGAN on object transfiguration applications, which targets at translating one object class to another object class. The images are randomly sampled from ImageNet [31] by the keywords (i.e. wild horse and zebra), and scaled to 256 × 256. The number of training images are 939 and 1177 for horse and zebra respectively.

Synthetic ← Real. We use the 2975 training images from Cityscapes as the real world scenes, and randomly selected 3060 images from SYNTHIA-CVPR16 [32], which is an virtual urban scene benchmark, as the synthetic images. We train our translators in the image resolution of 256 × 256.

Summar ← Winter. The images used for the season translation tasks are provided by CycleGAN, and resized to 256 × 256 in the training procedure. The training size for Summer and Winter is 1273 and 854 respectively.
6 Conclusion

In this paper, we propose to enforce geometry consistency as a constraint, which can be viewed as a predefined geometric transformation function $f(\cdot)$ preserving the geometry of a scene, for unsupervised domain mapping. By combining our geometry consistency constraint with a standard adversarial constraint, one-sided domain mapping can be targeted. We discuss our model, geometry-consistent adversarial network (GcGAN), qualitatively and quantitatively on a various of domain mapping tasks. Experimental results demonstrate that our GcGAN can achieve competitive even better translations than state-of-the-art methods. Last but not least, our geometry consistency constraint is compatible with other unsupervised constraints.

References

[1] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016.
[2] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017.
[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017.
[4] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. arXiv preprint arXiv:1711.11585, 2017.
[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In CVPR, 2017.
[6] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In NIPS, 2017.
[7] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In ICML, 2017.
[8] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In CVPR, 2017.
[9] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In NIPS, 2017.
[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, 2014.
[11] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In NIPS, 2016.
[12] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a? laplacian pyramid of adversarial networks. In NIPS, 2015.
[13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.
[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In NIPS, 2016.
[15] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.
[16] Konstantinos Bouwmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In NIPS, 2016.
[17] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In ICML, 2016.
[18] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In ICML, 2016.
[19] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In NIPS, 2016.
[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016.

[22] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation, 2016.

[23] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

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

[25] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[26] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wanda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[27] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[29] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[30] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[32] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.
Network Architecture

The generator and discriminator (except SVHN → MNIST) presented before are shown in Tab. 4. For convenience, we use the following abbreviation: C = Feature channel, K = Kernel size, S = Stride size, Deconv/Conv = Deconvolutional/Convolutional layer, and ResBlk = A residual block.

Table 4: The generator and discriminator used in our experiments (256 × 256).

| Generator | C | K | S |
|-----------|---|---|---|
| Index     | Layer        |   |   |   |
| 1         | Conv + ReLU  | 64| 7 | 1 |
| 2         | Conv + ReLU  | 128| 3| 2 |
| 3         | Conv + ReLU  | 256| 3| 2 |
| 4         | ResBlk + ReLU| 256| 3| 1 |
| 5         | ResBlk + ReLU| 256| 3| 1 |
| 6         | ResBlk + ReLU| 256| 3| 1 |
| 7         | ResBlk + ReLU| 256| 3| 1 |
| 8         | ResBlk + ReLU| 256| 3| 1 |
| 9         | ResBlk + ReLU| 256| 3| 1 |
| 10        | ResBlk + ReLU| 256| 3| 1 |
| 11        | ResBlk + ReLU| 256| 3| 1 |
| 12        | ResBlk + ReLU| 256| 3| 1 |
| 13        | ResBlk + ReLU| 256| 3| 1 |
| 14        | Deconv + ReLU| 128| 3| 2 |
| 15        | Tanh         | - | - | - |

| Discriminator | C | K | S |
|---------------|---|---|---|
| Index         | Layer            |   |   |   |
| 1             | Conv + LeakyReLU | 64| 4| 2 |
| 2             | Conv + LeakyReLU | 128| 4| 2 |
| 3             | Conv + LeakyReLU | 256| 4| 2 |
| 4             | Conv + LeakyReLU | 512| 4| 1 |
| 5             | Conv           | 512| 4| 1 |

The network architecture for SVHN → MNIST is reported in Tab. 5.

Table 5: The network architecture for SVHN → MNIST.

| Generator | C | K | S |
|-----------|---|---|---|
| Index     | Layer            |   |   |   |
| 1         | Conv + LeakyReLU | 64| 4| 2 |
| 2         | Conv + LeakyReLU | 128| 4| 2 |
| 3         | Conv + LeakyReLU | 128| 3| 1 |
| 4         | Conv + LeakyReLU | 128| 3| 1 |
| 5         | Deconv + LeakyReLU| 64| 4| 2 |
| 6         | Deconv + LeakyReLU| 128| 4| 2 |
| 7         | Deconv + LeakyReLU| 256| 4| 2 |
| 8         | Deconv + LeakyReLU| 512| 4| 1 |
| 9         | Conv           | 512| 4| 1 |

Discriminator

| Index     | Layer            |   |   |   |
| 1         | Conv + LeakyReLU | 64| 4| 2 |
| 2         | Conv + LeakyReLU | 128| 4| 2 |
| 3         | Conv + LeakyReLU | 256| 4| 2 |
| 4         | Conv + LeakyReLU | 512| 4| 1 |
| 5         | Conv           | 512| 4| 1 |
Other Qualitative Results

Our GcGAN performs much better than GAN alone. GAN alone even does not work in some translation tasks.

Figure 5: Cityscapes: Parsing $\leftrightarrow$ Image. We can see that GAN alone does not work for Image $\rightarrow$ Parsing task, and even generates the same parsing for different inputs.
Figure 6: **Google Maps: Map ⇌ Aerial photo.** Translated images by our GcGAN contain more details.
Figure 7: Horse $\Leftarrow$ Zebra.
Figure 8: **Synthetic $\iff$ Real.**
Figure 9: **Summer** $\Rightarrow$ **Winter.** GAN alone does not work for these translation tasks.