Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping
Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, Dacheng Tao

To cite this version:
Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, et al.. Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping. CVPR 2019 - IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2019, Long Beach, CA, United States. pp.2422-2431, 10.1109/CVPR.2019.00253. hal-03132263

HAL Id: hal-03132263
https://hal.archives-ouvertes.fr/hal-03132263
Submitted on 4 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping

Huan Fu*1 Mingming Gong*2,3 Chaohui Wang4 Kayhan Batmanghelich2 Kun Zhang3 Dacheng Tao1
1UBTECH Sydney AI Centre, School of Computer Science, FEIT, University of Sydney, Darlington, NSW 2008, Australia
2Department of Biomedical Informatics, University of Pittsburgh
3Department of Philosophy, Carnegie Mellon University
4Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, Marne-la-Vallée, France
{hufu6371@uni., dacheng.tao@sydney.edu.au {mig73, kayhan}@pitt.edu chaohui.wang@u-pem.fr kunz1@cmu.edu

Abstract

Unsupervised domain mapping aims to learn a function \( G_{XY} \) to translate domain \( X \) to \( Y \) in the absence of paired examples. Finding the optimal \( G_{XY} \) without paired data is an ill-posed problem, so appropriate constraints are required to obtain reasonable solutions. While some prominent constraints such as cycle consistency and distance preservation successfully constrain the solution space, they overlook the special properties of images that simple geometric transformations do not change the image’s semantic structure. Based on this special property, we develop a geometry-consistent generative adversarial network (GcGAN), which enables one-sided unsupervised domain mapping. 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 coupled with the corresponding geometry-consistency constraint. The geometry-consistency constraint reduces the space of possible solutions while keep the correct solutions in the search space. Quantitative and qualitative comparisons with the baseline (GAN alone) and the state-of-the-art methods including CycleGAN [66] and DistanceGAN [5] demonstrate the effectiveness of our method. Codes will be made available at https://github.com/hufu6371/GcGAN.

1. Introduction

Domain mapping or image-to-image translation, which targets at translating an image from one domain to another, has been intensively investigated over the past few years. Let \( X \in X \) denote a random variable representing source domain images and \( Y \in Y \) represent target domain images. According to whether we have access to a paired sample \( \{(x_i, y_i)\}_{i=1}^N \), domain mapping can be studied in a supervised or unsupervised manner. While several works have successfully produced high-quality translations by focusing on supervised domain mapping with constraints provided by cross-domain image pairs [46, 26, 59, 58], the progress of unsupervised domain mapping is relatively slow.

In unsupervised domain mapping, the goal is to model the joint distribution \( P_{XY} \) given samples drawn from the marginal distributions \( P_X \) and \( P_Y \) in individual domains. Since the two marginal distributions can be inferred from an infinite set of possible joint distributions, it is difficult to guarantee that an individual input \( x \in X \) and the output \( G_{XY}(x) \) are paired up in a meaningful way without additional assumptions or constraints.

To address this problem, recent approaches have exploited the cycle-consistency assumption, i.e., a mapping \( G_{XY} \) and its inverse mapping \( G_{YX} \) should be bijections [66, 28, 61]. Specifically, when feeding an example \( x \in X \) into the networks \( G_{XY} \circ G_{YX} : X \rightarrow Y \rightarrow X \), the output should be a reconstruction of \( x \) and vice versa for \( y \), i.e., \( G_{YX}(G_{XY}(x)) \approx x \) and \( G_{XY}(G_{YX}(y)) \approx y \). Further, DistanceGAN [5] showed that maintaining the distances between images within domains allows one-sided unsupervised domain mapping.

Existing constraints overlook the special properties of images that simple geometric transformations (global geometric transformations without shape deformation) do not change the image’s semantic structure. Here, semantic structure refers to the information that distinguishes different object/staff classes, which can be easily perceived by humans regardless of trivial geometric transformations such as rotation. Based on this property, we develop a geometry-consistency constraint, which helps in reducing the search space of possible solutions while still keeping the correct set of solutions under consideration, and results in a geometry-
consistent generative adversarial network (GcGAN).

Our geometry-consistency constraint is motivated by the fact that a given geometric transformation \( f(\cdot) \) between the input images should be preserved by related translators \( G_{XY} \) and \( G_{\tilde{X}\tilde{Y}} \), if \( X \) and \( Y \) are the domains obtained by applying \( f(\cdot) \) on the examples of \( X \) and \( Y \), respectively. Mathematically, given a random example \( x \) from the source domain \( X \) and a predefined geometric transformation function \( f(\cdot) \), geometry consistency can be expressed as \( f(G_{XY}(x)) \approx G_{\tilde{X}\tilde{Y}}(f(x)) \) and \( f^{-1}(G_{\tilde{X}\tilde{Y}}(f(x))) \approx G_{XY}(x) \), where \( f^{-1}(\cdot) \) is the inverse function of \( f(\cdot) \). Because it is unlikely that \( G_{XY} \) and \( G_{\tilde{X}\tilde{Y}} \) always fail in the same location, \( G_{XY} \) and \( G_{\tilde{X}\tilde{Y}} \) co-regularize each other by the geometry-consistency constraint and thus correct each others’ failures in local regions of their respective translations (see Figure 1 for an illustration). Our geometry-consistency constraint allows one-sided unsupervised domain mapping, i.e., \( G_{XY} \) can be trained independently from \( G_{YX} \). In this paper, we employ two simple but representative geometric transformations as examples, i.e., vertical flipping (vf) and 90 degrees clockwise rotation (rot), to illustrate geometry consistency. Quantitative and qualitative comparisons with the baseline (GAN alone) and the state-of-the-art methods including CycleGAN [66] and DistanceGAN [5] demonstrate the effectiveness of our method.

2. Related Work

Generative Adversarial Networks. Generative adversarial networks (GANs) [21, 45, 14, 47, 51, 3] learn two networks, i.e., a generator and a discriminator, in a staged zero-sum game fashion to generate images from inputs. Many tasks have recently been developed based on deep convolutional GANs, such as image inpainting, style transfer, and domain adaptation [7, 62, 46, 48, 31, 60, 9, 52, 23, 53, 64, 27, 50, 19, 18, 35, 63]. The key component enabling GANs is the adversarial constraint, which enforces the generated images to be indistinguishable from real images.

Domain Mapping. Recent adversarial domain mapping has been studied in a supervised or unsupervised manner with respect to paired or unpaired inputs. There are a variety of literatures [46, 31, 26, 59, 56, 58, 25, 37, 4, 10] on supervised domain mapping. One representative example is conditional GAN [26], which learned the discriminator to distinguish \((x, y)\) and \((x, G_{XY}(x))\) instead of \( y \) and \( G_{XY}(x) \), where \((x, y)\) is a meaningful pair across domains. Further, Wang et al. [59] showed that conditional GANs can be used to generate high-resolution images with a novel feature matching loss, as well as multi-scale generator and discriminator architectures. While there has been significant progress in supervised domain mapping, many real-world applications can not provide aligned images across domains because data preparation is expensive. Thus, different constraints and frameworks have been proposed for image-to-image translation in the absence of training pairs.

In unsupervised domain mapping, only unaligned examples in individual domains are provided, making the task more realistic but more difficult. Unpaired domain mapping has a long history, and some successes in adversarial networks have recently been presented [40, 66, 5, 39, 42, 38, 6, 11]. For example, Liu and Tuzel [40] introduced coupled GAN (CoGAN) to learn cross-domain representations by enforcing a weight-sharing constraint. Subsequently, CycleGAN [66], DiscoGAN [28], and DualGAN [61] enforced that translators \( G_{XY} \) and \( G_{YX} \) should be bijections. Thus, jointly learning \( G_{XY} \) and \( G_{YX} \) by enforcing cycle consistency can help to produce convincing mappings. Since then, many constraints and assumptions have been proposed to improve cycle consistency [8, 17, 24, 32, 34, 11, 2, 67, 20, 44, 39, 36, 1]. Recently, Benaim and Wolf [5] reported that maintaining the distances between samples within domains allows one-sided unsupervised domain mapping. GcGAN is also a one-sided framework coupled with our geometry-consistency constraint, and produces competitive and even better translations than the two-sided CycleGAN in various applications.
3. Preliminaries

Let \( \mathcal{X} \) and \( \mathcal{Y} \) be two domains with unpaired training examples \( \{x_i\}_{i=1}^N \) and \( \{y_j\}_{j=1}^M \), where \( x_i \) and \( y_j \) are drawn from the marginal distributions \( P_X \) and \( P_Y \), where \( X \) and \( Y \) are two random variables associated with \( \mathcal{X} \) and \( \mathcal{Y} \), respectively. In the paper, we exploit style transfer without undesirable semantic distortions, and have two goals. First, we need to learn a mapping \( G_{XY} \) such that \( G_{XY}(X) \) has the same distribution as \( Y \), i.e., \( P_{G_{XY}(X)} \approx P_Y \). Second, the learned mapping function only changes the image style without distorting the semantic structures.

While many works have modeled the invertibility between \( G_{XY} \) and \( G_{YX} \) for convincing mappings since the success of CycleGAN, here we propose to enforce geometry consistency as a constraint that allows one-sided domain mapping. Let \( f(\cdot) \) be a predefined geometric transformation. We can obtain two extra domains \( \tilde{\mathcal{X}} \) and \( \tilde{\mathcal{Y}} \) with examples \( \{\tilde{x}_i\}_{i=1}^N \) and \( \{\tilde{y}_j\}_{j=1}^M \) by applying \( f(\cdot) \) on \( X \) and \( Y \), respectively. We learn an additional translator \( G_{\tilde{X}\tilde{Y}} : \tilde{\mathcal{X}} \rightarrow \tilde{\mathcal{Y}} \) while learning \( G_{XY} : \mathcal{X} \rightarrow \mathcal{Y} \), and introduce our geometry-consistency constraint based on the predefined transformation such that the two networks can co-regularize each other. Our framework enforces that \( G_{XY}(x) \) and \( G_{\tilde{X}\tilde{Y}}(\tilde{x}) \) should keep the same geometric transformation with the one between \( x \) and \( \tilde{x} \), i.e., \( f(G_{XY}(x)) \approx G_{\tilde{X}\tilde{Y}}(\tilde{x}) \), where \( \tilde{x} = f(x) \). We denote the two adversarial discriminators as \( D_Y \) and \( D_{\tilde{Y}} \) with respect to domains \( \mathcal{Y} \) and \( \tilde{\mathcal{Y}} \), respectively.

4. Proposed Method

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

4.1. Unsupervised Constraints

Cycle-consistency constraint. Following the cycle-consistency assumption [28, 66, 61], through the translators \( G_{XY} \circ G_{YX} : X \rightarrow Y \rightarrow X \) and \( G_{X\tilde{Y}} \circ G_{\tilde{Y}X} : \mathcal{X} \rightarrow \mathcal{Y} \rightarrow \mathcal{X} \), the examples \( x \) and \( y \) in domain \( \mathcal{X} \) and \( \mathcal{Y} \) should recover the original images, i.e., \( x \approx G_{XY}(G_{YX}(x)) \) and \( y \approx G_{X\tilde{Y}}(G_{\tilde{Y}X}(y)) \). Cycle consistency is implemented by a bidirectional reconstruction process that requires \( G_{XY} \) and \( G_{YX} \) to be jointly learned, as shown in Figure 2 (CycleGAN). The cycle consistency loss \( \mathcal{L}_{cyc}(G_{XY}, G_{YX}, X, Y) \) takes the form as:

\[
\mathcal{L}_{cyc} = \mathbb{E}_{x \sim P_X} [\|G_{XY}(G_{YX}(x)) - x\|_1] + \mathbb{E}_{y \sim P_Y} [\|G_{YX}(G_{XY}(y)) - y\|_1].
\]

Distance constraint. The assumption behind the distance constraint is that the distance between two examples \( x_i \) and \( x_j \) in domain \( \mathcal{X} \) should be preserved after mapping to domain \( \mathcal{Y} \), i.e., \( d(x_i, x_j) \approx a \cdot d(G_{XY}(x_i), G_{XY}(x_j)) + b \), where \( d(\cdot) \) is a predefined function to measure the distance between two examples and \( a \) and \( b \) are the linear coefficient and bias. In DistanceGAN [5], the distance consistency loss \( \mathcal{L}_{dis}(G_{XY}, X, Y) \) is the exception to the absolute differ-
ences between distances:

\[
\mathcal{L}_{\text{dis}} = \mathbb{E}_{x_i, x_j \sim P_X} [\| \phi(x_i, x_j) - \psi(x_i, x_j) \|_1],
\]

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

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

where \(\mu_X, \mu_Y, \sigma_X, \sigma_Y\) are the means (standard deviations) of distances of all the possible pairs of \((x_i, x_j)\) within domain \(X\) and \((y_i, y_j)\) within domain \(Y\), respectively.

### 4.2. Geometry-consistent Generative Adversarial Networks

**Adversarial constraint.** Taking \(G_{XY}\) as an example, an adversarial loss \(\mathcal{L}_{\text{gan}}(G_{XY}, D_Y, X, Y)\) [21] enforces \(G_{XY}\) and \(D_Y\) to simultaneously optimize each other in an min-max game, i.e., \(\min_{G_{XY}} \max_{D_Y} \mathcal{L}_{\text{gan}}(G_{XY}, D_Y, X, Y)\). In other words, \(D_Y\) aims to distinguish real examples \{\(y\)\} from translated samples \{\(G_{XY}(x)\)\}. By contrast, \(G_{XY}\) aims to fool \(D_Y\) so that \(D_Y\) can label a fake example \(y' = G_{XY}(x)\) as a sample satisfying \(y' \sim P_Y\). The objective can be expressed as:

\[
\mathcal{L}_{\text{gan}} = \mathbb{E}_{y \sim P_Y} [\log D_Y(y)] + \mathbb{E}_{x \sim P_X} [\log(1 - D_Y(G_{XY}(x)))].
\]

In the transformed domains \(\tilde{X}\) and \(\tilde{Y}\), we employ the adversarial loss \(\mathcal{L}_{\text{gan}}(G_{\tilde{X} \tilde{Y}}, D_{\tilde{Y}}, \tilde{X}, \tilde{Y})\) that has the same form to \(\mathcal{L}_{\text{gan}}(G_{XY}, D_Y, X, Y)\).

**Geometry-consistency constraint.** As shown in Figure 2 (GcGAN), given a predefined geometric transformation function \(f(\cdot)\), we feed the images \(x \in X\) and \(\tilde{x} = f(x)\) into the translators \(G_{XY}\) and \(G_{\tilde{X} \tilde{Y}}\), respectively. Following our geometry-consistency constraint, the outputs \(y' = G_{XY}(x)\) and \(\tilde{y}' = G_{\tilde{X} \tilde{Y}}(\tilde{x})\) should also satisfy \(y' \approx f(y')\) like \(x\) and \(\tilde{x}\). Considering both \(f(\cdot)\) and the inverse geometric transformation function \(f^{-1}(\cdot)\), our complete geometry consistency loss \(\mathcal{L}_{\text{geo}}(G_{XY}, G_{\tilde{X} \tilde{Y}}, X, Y)\) has the following form:

\[
\mathcal{L}_{\text{geo}} = \mathbb{E}_{x \sim P_X} [\|G_{XY}(x) - f^{-1}(G_{\tilde{X} \tilde{Y}}(f(x)))\|_1] + \mathbb{E}_{\tilde{x} \sim P_X} [\|G_{\tilde{X} \tilde{Y}}(f(x)) - f(G_{XY}(x))\|_1].
\]

This geometry-consistency loss can be seen as a reconstruction loss that relies on the predefined geometric transformation function \(f(\cdot)\). In this paper, we only take two common geometric transformations as examples, namely vertical flipping (\(yf\)) and 90° clockwise rotation (\(rot\)), to demonstrate the effectiveness of our geometry-consistency constraint. Note that, \(G_{XY}\) and \(G_{\tilde{X} \tilde{Y}}\) have the same architecture and share all the parameters.

**Full objective.** By combining our geometry-consistency constraint with the standard adversarial constraint, a remarkable one-sided unsupervised domain mapping can be targeted. The full objective for our GcGAN \(L_{\text{GcGAN}}(G_{XY}, G_{\tilde{X} \tilde{Y}}, D_Y, D_{\tilde{Y}}, X, Y)\) will be:

\[
L_{\text{GcGAN}} = L_{\text{gan}}(G_{XY}, D_Y, X, Y) + L_{\text{gan}}(G_{\tilde{X} \tilde{Y}}, D_{\tilde{Y}}, X, Y) + \lambda L_{\text{geo}}(G_{XY}, G_{\tilde{X} \tilde{Y}}, X, Y),
\]

where \(\lambda (\lambda = 20.0\) in all the experiments\) is a trade-off hyperparameter to weight the contribution of \(L_{\text{gan}}\) and \(L_{\text{geo}}\) during the model training. Carefully tuning \(\lambda\) may give preferable results to specific translation tasks.

**Network architecture.** The full framework of our GcGAN is illustrated in Figure 2. Our experimental settings, network architectures, and learning strategies follow CycleGAN. We employ the same discriminator and generator as CycleGAN depending on the specific tasks. Specifically, the generator is a standard encoder-decoder, where the encoder contains two convolutional layers with stride 2 and several residual blocks [22] (6 / 9 blocks with respect to 128 × 128 / 256 × 256 of input resolution), and the decoder contains two deconvolutional layers also with stride 2. The discriminator distinguishes images at the patch level following PatchGANS [26, 33]. Like CycleGAN, we also use an identity mapping loss [55] in all of our experiments (except SVHN → MNIST, including our baseline (GAN alone). For other details, we use LeakyReLU as nonlinearity for the discriminators and instance normalization [57] to normalize convolutional feature maps.

**Learning and inference.** We use the Adam solver [29] with a learning rate of 0.0002 and coefficients of (0.5, 0.999), where the latter is used to compute running averages of gradients and their squares. The learning rate is fixed in the initial 100 epochs, and linearly decays to zero over the next 100 epochs. Following CycleGAN, the negative log likelihood objective is replaced with a more stable and effective least-squares loss [43] for \(L_{\text{gan}}\). The discriminator is updated with random samples from a history of generated images stored in an image buffer [54] of size 50. The generator and discriminator are optimized alternately. In the inference phase, we feed an image only into the learned generator \(G_{XY}\) to obtain a translated image.

### 5. Experiments

We apply our GcGAN to a wide range of applications and make both quantitative and qualitative comparisons with the baseline (GAN alone) and previous state-of-the-art methods including DistanceGAN and CycleGAN. We also study different ablations (based on \(rot\)) to analyze
our geometry-consistency constraint. Since adversarial networks are not always stable, every independent experiment could result in slightly different scores. The scores in the quantitative analysis are computed by the average on three independent experiments.

### 5.1. Quantitative Analysis

The results demonstrate that our geometry-consistency constraint can not only partially filter out the candidate solutions having mode collapse or semantic distortions and thus produce more sensible translations, but also compatible with other unsupervised constraints such as cycle consistency [66] and distance preservation [5].

**Cityscapes.** Cityscapes [12] contains 3975 image-label pairs, with 2975 used for training and 500 for validation (test in this paper). For a fair comparison with CycleGAN, the translators are trained at a resolution of $128 \times 128$ in an unaligned fashion. We evaluate our domain mappers using FCN scores and scene parsing metrics following previous works [41, 12, 66]. Specifically, for parsing $\rightarrow$ image, we assume that a high-quality translated image should produce qualitative semantic segmentation like real images when feeding it into a scene parser. Thus, we employ the pretrained FCN-8s [41] provided by pix2pix [26] to predict semantic labels for the 500 translated images. The label maps are then resized to the original resolution of $1024 \times 2048$ and compared against the ground truth labels using some standard scene parsing metrics including pixel accuracy, class accuracy, and mean IoU [41]. For image $\rightarrow$ parsing, since the fake labels are in the RGB format, we simply convert them into class-level labels using the nearest neighbor search strategy. In particular, we have 19 (category labels) + 1 (ignored label) categories for Cityscapes, each with a corresponding color value (RGB). For a pixel $i$ in a translated parsing, we compute the distances between the 20 groundtruth color values and the color value of pixel $i$. The label of pixel $i$ should be the one with the smallest distance. Then, the aforementioned metrics are used to evaluate our mapping on the 19 category labels.

The parsing scores are presented in Table 1. Our GcGAN outperforms the baseline (GAN alone) by a large margin. We take the average of pixel accuracy, class accuracy, and mean IoU as the final score for analysis [65], i.e., score = (pixel acc + class acc + mean IoU)/3. For image $\rightarrow$ parsing, GcGAN (32.6%) yields a slightly higher score than CycleGAN (32.0%). For parsing $\rightarrow$ image, GcGAN (29.0% $\sim$ 29.5%) obtains a convincing improvement of 1.3% $\sim$ 1.8% over distanceGAN (27.7%).

We next perform ablation studies to further discuss GcGAN. The scores are reported in Table 1. Specifically, GcGAN-not-Separate shows that the generator $G_{XY}$ employed in GcGAN is sufficient to handle both the style transfers (without shape deformation) $X \rightarrow Y$ and $\tilde{X} \rightarrow \tilde{Y}$.
GeGAN-Mix-\{comb, rand\} demonstrate that persevering a geometric transformation can filter out most of the candidate solutions having mode collapse or undesired shape deformation, but preserving more ones can not leach more. Besides, GeGAN-Mix-rand performs slightly worse than GeGAN-Mix-comb. One of the possible reasons is that neither $X_{\text{rot}} \rightarrow Y_{\text{rot}}$ nor $X_{\text{vf}} \rightarrow Y_{\text{vf}}$ are sufficiently trained in the random case, which would decrease the effect of the aforementioned co-regularization mechanism. For GeGAN-\textit{rot} + Cycle, we set the trade-off parameter for $L_{\text{GAN}}$ to 10.0 as published in CycleGAN. The consistent improvement is a credible support that our geometry-consistency constraint is compatible with the widely-used cycle-consistency constraint. Moreover, when setting $\lambda = 0$ in $L_{\text{GAN}}$, both $G_{XY}$ and $G_{YX}$ perform badly. One of the possible reasons is that, without the geometry consistency constraint, jointly modeling $X \rightarrow Y$ and $X \rightarrow \hat{Y}$ with the shared generator $G_{XY}$ would decrease the performance due to domain diversities caused by the geometric transformations. When removing $L_{\text{gan}}(G_{\hat{X} \hat{Y}}, D_{\hat{Y}})$, the obtained scores are much higher than baseline (GAN alone) because $Y'$ can partially correct $\hat{Y}'$ so that $G_{XY}$ is able to handle the mapping $X \rightarrow \hat{Y}$, and then $\hat{Y}'$ can constrain the mapping $X \rightarrow Y$. As the analysis, when learning both $L_{\text{gan}}(G_{X \hat{Y}}, D_{\hat{Y}})$ and $L_{\text{gan}}(G_{\hat{X} Y}, D_{Y})$ with $L_{\text{geo}}$, the co-regularization help generate preferable translations.

**SVHN → MNIST.** We apply our approach to the SVHN → MNIST translation task. The translation models are trained on 73257 and 60000 training images of resolution $32 \times 32$ contained in the SVHN and MNIST training sets, respectively. The experimental settings follow DistanceGAN [5], including the default trade-off parameters for $L_{\text{cyc}}$ and $L_{\text{dis}}$. We compare our GeGAN with both DistanceGAN and CycleGAN in this translation task. To obtain quantitative results, we feed the translated images into a pretrained classifier trained on the MNIST training split, as done in [5].

Classification accuracies are reported in Table 2. Both GeGAN-\textit{rot} and GeGAN-\textit{vf} outperform DistanceGAN and CycleGAN by a large margin (about $6\% \sim 7\%$). From the ablations, adding our geometry-consistency constraint to current unsupervised domain mapping frameworks will achieve different levels of improvements against the original ones. Note that, it seems that the distance-preservation

Table 1: **Parsing scores on Cityscapes.** $L_{\text{GAN}}$: The objective in Eqn. 5 with \textit{rot}. GeGAN-\textit{rot}-Separate: $G_{XY}$ and $G_{\hat{X} \hat{Y}}$ do not share parameters. GeGAN-Mix-comb: Training GeGAN with both $vf$ and $rot$ in each iteration. GeGAN-Mix-rand: Training GeGAN with randomly chosen $vf$ and $rot$ in each iteration. GeGAN-\textit{rot} + Cycle: GeGAN-\textit{rot} with the cycle-consistency constraint.

<table>
<thead>
<tr>
<th>method</th>
<th>parsing (\rightarrow) image</th>
<th>image (\rightarrow) parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pixel acc</td>
<td>class acc</td>
</tr>
<tr>
<td>Benchmark Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GcGAN [40]</td>
<td>0.45</td>
<td>0.11</td>
</tr>
<tr>
<td>BiGAN/ALI [15, 16]</td>
<td>0.41</td>
<td>0.13</td>
</tr>
<tr>
<td>SimGAN [54]</td>
<td>0.47</td>
<td>0.11</td>
</tr>
<tr>
<td>CycleGAN (Cycle) [66]</td>
<td>0.58</td>
<td>0.22</td>
</tr>
<tr>
<td>DistanceGAN [5]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GAN alone (baseline)</td>
<td>0.514</td>
<td>0.160</td>
</tr>
<tr>
<td>GcGAN-\textit{rot}</td>
<td>0.574</td>
<td>0.234</td>
</tr>
<tr>
<td>GcGAN-\textit{vf}</td>
<td>0.576</td>
<td>0.232</td>
</tr>
</tbody>
</table>

Ablation Studies (Robustness & Compatibility)

<table>
<thead>
<tr>
<th>Method</th>
<th>$L_{\text{GAN}}$ w/o $L_{\text{geo}}(\lambda = 0)$</th>
<th>$L_{\text{GAN}}$ w/o $L_{\text{gan}}(\hat{X}, \hat{Y})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeGAN-\textit{rot}-Separate</td>
<td>0.486</td>
<td>0.163</td>
</tr>
<tr>
<td>GeGAN-Mix-comb</td>
<td>0.549</td>
<td>0.199</td>
</tr>
<tr>
<td>GeGAN-\textit{vf}</td>
<td>0.575</td>
<td>0.233</td>
</tr>
<tr>
<td>GeGAN-\textit{vf}</td>
<td>0.573</td>
<td>0.229</td>
</tr>
<tr>
<td>GeGAN-\textit{vf}</td>
<td>0.564</td>
<td>0.217</td>
</tr>
<tr>
<td>GeGAN-\textit{rot} + Cycle</td>
<td>0.587</td>
<td>0.246</td>
</tr>
</tbody>
</table>

Table 2: Quantitative scores for SVHN \(\rightarrow\) MNIST.
constraint is not compatible with the cycle-consistency constraint on this task, but our geometry-consistency constraint can improve both ones.

![DistanceGAN [5] vs GcGAN](image)

**Figure 4:** Qualitative comparison for SVHN → MNIST.

**Google Maps.** We obtain 2194 (map, aerial photo) pairs of images in and around New York City from Google Maps [26], and split them into training and test sets with 1096 and 1098 pairs, respectively. We train Map → Aerial photo translators with an image size of $256 \times 256$ using the training set in an unsupervised manner (unpaired) by ignoring the pair information. For Aerial photo → Map, we make comparisons with CycleGAN using 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 consider this is an accurate prediction. Since maps only contain a limited number of different RGB values, it is reasonable to compute pixel accuracy using this strategy ($\delta_1 = 5$ and $\delta_2 = 10$ in this paper). For Map → Aerial photo, we only show some qualitative results in Figure 3.

<table>
<thead>
<tr>
<th>method</th>
<th>RMSE</th>
<th>acc ($\delta_1$)</th>
<th>acc ($\delta_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Benchmark Performance</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CycleGAN [66]</td>
<td>28.15</td>
<td>41.8</td>
<td>63.7</td>
</tr>
<tr>
<td>GAN alone (baseline)</td>
<td>33.27</td>
<td>19.3</td>
<td>42.0</td>
</tr>
<tr>
<td>GcGAN-rot</td>
<td>28.31</td>
<td>41.2</td>
<td>63.1</td>
</tr>
<tr>
<td>GcGAN-vf</td>
<td>28.50</td>
<td>37.3</td>
<td>58.9</td>
</tr>
<tr>
<td><strong>Ablation Studies (Robustness &amp; Compatibility)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GcGAN-rot-Separate</td>
<td>30.25</td>
<td>40.7</td>
<td>60.8</td>
</tr>
<tr>
<td>GcGAN-Mix-comb</td>
<td>27.98</td>
<td>42.8</td>
<td>64.6</td>
</tr>
<tr>
<td>GcGAN-rot + Cycle</td>
<td>28.21</td>
<td>40.6</td>
<td>63.5</td>
</tr>
</tbody>
</table>

**Table 3:** Quantitative scores for Aerial photo → Map.

From the scores presented in Table 3, GcGAN produces superior translations to the baseline (GAN alone). In particular, GcGAN yields an $18.0\% \sim 21.9\%$ improvement over the baseline with respect to pixel accuracy when $\delta = 5.0$, demonstrating that the fake maps obtained by our GcGAN contain more details. In addition, GcGANs achieve competitive scores compared with CycleGAN.

### 5.2. Qualitative Evaluation

The qualitative results are shown in Figure 3, Figure 4, and Figure 5. Our geometry-consistency constraint improve the training of GAN alone, and helps to generate empirically more impressive translations on various applications. The following applications are trained in the image size of $256 \times 256$ with the rot geometric transformation.

**Horse → Zebra.** We apply GcGAN to the widely studied object transfiguration application task, i.e., Horse → Zebra. The images are randomly sampled from ImageNet [13] using the keywords (i.e., wild horse and zebra). The numbers of training images are 939 and 1177 for horse and zebra, respectively. We find that training GcGAN without parameter sharing would produce preferable translations for the task.

**Synthetic = Real.** We employ the 2975 training images from Cityscapes as the real-world scenes, and randomly select 3060 images from SYNTHIA-CVPR16 [49], which is a virtual urban scene benchmark, as the synthetic images.

**Summer = Winter.** The images used for the season translation tasks are provided by CycleGAN. The training set sizes for Summer and Winter are 1273 and 854.

**Photo = Artistic Painting.** We translate natural images to artistic paintings with different art styles, including Monet, Cezanne, Van Gogh, and Ukiyo-e. We also perform GcGAN on the translation task of Monet’s paintings → photographs. We use the photos and paintings (Monet: 1074, Cezanne: 584, Van Gogh: 401, Ukiyo-e: 1433, and Photographs: 6853) collected by CycleGAN for training.

**Day = Night.** We randomly extract 4500 training images for both Day and Night from the 91 webcam sequences captured by [30].

### 6. Conclusion

In this paper, we propose to enforce geometry consistency as a constraint for unsupervised domain mapping, which can be viewed as a predefined geometric transformation $f(\cdot)$ preserving the geometry of a scene. The geometry-consistency constraint makes the translation networks on the original images and transformed images co-regularize each other, which not only provides an effective remedy to the mode collapse problem suffered by standard GANs, but also reduces the semantic distortions in the translation. We evaluate our model, i.e., the geometry-consistent generative adversarial network (GcGAN), both qualitatively and quantitatively in various applications. The experimental results demonstrate that GcGAN achieves competitive and sometimes even better translations than the state-of-the-art methods including DistanceGAN and CycleGAN. Finally, our geometry-consistency constraint is compatible with other well-studied unsupervised constraints.

### 7. Acknowledgement

This research was supported by Australian Research Council Projects FL-170100117, DP-180103424, and IH-
Figure 5: Qualitative results on different applications, including Horse → Zebra, Monet → Photo, Synthetic ↔ Real, Summer ↔ Winter, Photo → Artistic Painting, and Day ↔ Night. GcGAN has the potential to produce realistic images. Zoom in for better view.
References

[16] Aaron Gokaslan, Vivek Ramanujan, Daniel Ritchie, Kwang In Kim, and James Tompkin. Improving shape deformation in unsupervised image-to-image translation. ECCV, 2018. 2
[18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. ECCV, 2018. 2

[33] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016. 4


8. Network Architecture

The generator and discriminator (except for 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>
<thead>
<tr>
<th>Generator</th>
<th>C</th>
<th>K</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conv + ReLU</td>
<td>64</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Conv + ReLU</td>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Conv + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>ResBlk + ReLU</td>
<td>256</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>Deconv + ReLU</td>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>Conv</td>
<td>64</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Tanh</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discriminator</th>
<th>C</th>
<th>K</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conv + LeakyReLU</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Conv + LeakyReLU</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Conv + LeakyReLU</td>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Conv + LeakyReLU</td>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Deconv + LeakyReLU</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Deconv + LeakyReLU</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Tanh</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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

9. Other Qualitative Results

We show other qualitative results for the fourteen applications in our submission (Figure 6 to Figure 14). Our geometry-consistency constraint makes the translation networks on the original images and transformed images co-regularize each other, which not only provides an effective remedy to the mode collapse problem suffered by standard GANs, but also reduces the semantic distortions in the translation. As a result, GcGAN can produce empirically impressive translations.

10. Failure Cases

Unluckily, GcGAN cannot guarantee reasonable translations for all the cases as previous works. Thus, more assumptions and constraints should be investigated to improve unsupervised domain mapping. Some failure cases are shown in Figure 15.

The network architecture for SVHN → MNIST is reported in Tab. 5.
Figure 6: Cityscapes (Parsing ⇆ Image). The results for CycleGAN [62] are produced by the officially provided pretrained PyTorch models. GcGAN denotes GcGAN-rot.
Figure 7: Google Mpas (Aerial photo $\rightarrow$ Map). For Map $\rightarrow$ Aerial photo, GeGAN produces competitive translations compared with CycleGAN [62].
Figure 8: SVHN $\rightarrow$ MNIST. The qualitative results for both CycleGAN [62] and DistanceGAN [5] come from DistanceGAN [5]. The correct translations are about 24, 26, and 35 for CycleGAN, DistanceGAN, and GeGAN, respectively.
Horse $\rightarrow$ Zebra

Figure 9: Horse $\rightarrow$ Zebra. For this task, GcGAN generates slightly better translations for some images, but can not perform better than CycleGAN [62] generally.
Figure 10: **Monet → Photo.** GcGAN is superior in generating realistic images. Zoom in for better view.
Figure 11: **Synthetic \(\rightarrow\) Real.** We train CycleGAN [62] using the released PyTorch codes. The results produced by GcGAN contain more details. Zoom in for better view.
Figure 12: *Summer ↔ Winter*. Here, GeGAN represents GeGAN-rot-Seperate. Zoom in for better view.

Figure 14: *Summer → Winter*. Here, GcGAN represents GcGAN-rot-Seperate. Zoom in for better view.
Photographs $\rightarrow$ Artist paintings

Figure 13: Photographs $\rightarrow$ Artist paintings. We translate a photo to the artistic styles of Monet, Van Gogh, Cezanne, and U-kiyoe.
Figure 13: Day $\rightarrow$ Night.

Figure 14: Day $\iff$ Night.
Figure 15: **Failure Cases.** GcGAN cannot guarantee reasonable translations for all the cases as previous works. Thus, more assumptions and constraints should be investigated to improve unsupervised domain mapping.