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

Huan Fu[*1] **Mingming Gong**[* 2,3]

Chaohui Wang[4] **Kayhan Batmanghelich**[2,3] **Kun Zhang**[3] **Dacheng Tao**[1]

[1]UBTECH Sydney AI Centre, SIT, FEIT, The University of Sydney, Australia
[2]Department of Biomedical Informatics, University of Pittsburgh
[3]Department of Philosophy, Carnegie Mellon University
[4]Laboratoire 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 to learn a function to translate domain $\mathcal{X}$ to $\mathcal{Y}$ by a function G_{XY} 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. One of the most prominent constraints 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 the simultaneous training of G_{XY} and G_{YX}, recent studies have shown that one-sided domain mapping can be achieved by preserving pairwise distances between images. Although 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 [62] and DistanceGAN [5] demonstrate the effectiveness of our method.

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 $X \in \mathcal{X}$ denote a random variable representing source domain images and $Y \in \mathcal{Y}$ 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 [43, 24, 56, 55], the progress of unsupervised domain mapping is relatively slow. Unluckily, obtaining paired training examples is expensive and even infeasible in some situations. For example, if we want to learn translators between Monet's paintings and Photographs, how can we collect sufficient well-defined (*Monet's painting, photograph*) pairs for model training? By contrast, collecting unpaired sets is often convenient since infinite images are available online. From this viewpoint, unsupervised domain mapping has great potential for real-world applications in the long term.

[*]equal contribution

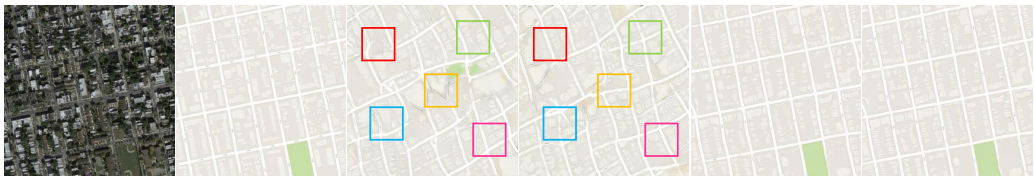


Figure 1: **Geometry consistency.** The original input image is denoted by x, and the predefined function $f(\cdot)$ is a 90° clockwise rotation (*rot*). GAN alone: $G^1_{XY}(x)$. GAN alone (*rot*): $f^{-1}(G^1_{\tilde{X}\tilde{Y}}(f(x)))$. GcGAN: $G^2_{XY}(x)$. GcGAN (*rot*): $f^{-1}(G^2_{\tilde{X}\tilde{Y}}(f(x))$. It can be seen that GAN alone produces geometrically-inconsistent output images, indicating that the learned G_{XY} and $G_{\tilde{X}\tilde{Y}}$ are far away from the correct mapping functions. By enforcing geometry consistency, our method results in more sensible domain mapping. $GcGAN = GAN\ alone + geometry\ consistency$.

In unsupervised domain mapping, from a probabilistic modeling perspective, our 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 [62, 26, 58]. Specifically, when feeding an example $x \in X$ into the networks $G_{XY} \circ G_{YX} : X \to Y \to X$, the output should be a reconstruction of x and vise 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 rather than simultaneously learning both G_{XY} and G_{YX}.

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) for unsupervised domain mapping.

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 $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ 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 $\mathcal{X}$ 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 illustrative example). 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 [62] and DistanceGAN [5] demonstrate the effectiveness of our model in generating realistic images.

2 Related Work

Generative Adversarial Networks. Generative adversarial networks (GANs) [19, 42, 14, 44, 48, 3] learn two networks, *i.e.*, a generator and a discriminator, in a staged zero-sum game fashion to generate images from inputs. Many applications and computer vision tasks have recently been developed based on deep convolutional GANs (DCGANs), such as image inpainting, text to image synthesis, style transfer, and domain adaptation [7, 59, 43, 45, 29, 57, 9, 49, 21, 50, 60, 25, 47]. The

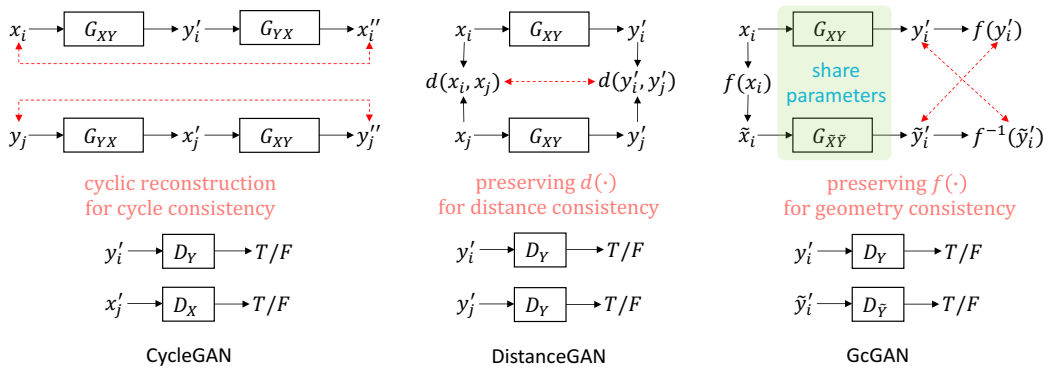


Figure 2: **An illustration of the differences between CycleGAN [62], DistanceGAN [5], and our GcGAN**. x and y are random examples from domain $\mathcal{X}$ and $\mathcal{Y}$, respectively. $d(x_i, x_j)$ is the distance between images x_i and x_j. $f(\cdot)$ is a predefined geometric transformation function for images, which satisfies $f^{-1}(f(x)) = f(f^{-1}(x)) = x$. G_{XY} and $G_{\tilde{X}\tilde{Y}}$ are the generators (or translators) which target the domain translation tasks from $\mathcal{X}$ to $\mathcal{Y}$ and $\tilde{\mathcal{X}}$ to $\tilde{\mathcal{Y}}$, where $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are two domains obtained by applying $f(\cdot)$ on all the images in $\mathcal{X}$ and $\mathcal{Y}$, respectively. D_Y is an adversarial discriminator in domain $\mathcal{Y}$. The red dotted lines denote the unsupervised constraints with respect to cycle consistency ($x \approx G_{YX}(G_{XY}(x))$), distance consistency ($x \approx G_{YX}(G_{XY}(x))$), and our geometry consistency ($f(G_{XY}(x)) \approx G_{\tilde{X}\tilde{Y}}(f(x))$), respectively.

key components enabling GANs is the proposed adversarial constraint, which enforces the generated images to be indistinguishable from real images. Our formulation also benefits from an adversarial constraint to learn translators between two individual domains.

Domain Mapping. Many well-known computer vision tasks, such as scene parsing and image colorization, follow similar settings to domain mapping or image-to-image translation. Specific to recent adversarial domain mapping, this problem has been studied in a supervised or unsupervised manner with respect to paired or unpaired inputs.

There are a variety of literatures [43, 29, 24, 56, 53, 55, 23, 34, 4, 10] on supervised domain mapping. One representative example is conditional GAN [24], which learns 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.* [56] 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-word 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, *i.e.*, unsupervised domain mapping.

In unsupervised domain mapping, only unaligned examples in individual domains are provided, making the task more practical but more difficult. Unpaired domain mapping has a long history, and some successes in adversarial networks have recently been presented [37, 62, 5, 36, 39, 35, 6, 11]. For example, Liu and Tuzel [37] introduced coupled GAN (CoGAN) to learn cross-domain representations by enforcing a weight-sharing constraint. Subsequently, CycleGAN [62], DiscoGAN [26], and DualGAN [58] 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, 22, 30, 32, 11, 2, 63, 18, 41, 36, 33, 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 in unsupervised domain mapping, 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, *i.e.*, learning G_{XY} without simultaneously learning G_{YX}. Let $f(\cdot)$ be a predefined geometric transformation. We can obtain two extra domains $\tilde{\mathcal{X}}$ 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 image-to-image translator $G_{\tilde{X}\tilde{Y}} : \tilde{X} \rightarrow \tilde{Y}$ while learning $G_{XY} : X \rightarrow Y$, and introduce our geometry-consistency constraint based on the predefined transformation such that the two networks can 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 main differences between these constraints is shown in Figure 2.

4.1 Unsupervised Constraints

Cycle-consistency constraint. Following the cycle-consistency assumption [26, 62, 58], through the translators $G_{XY} \circ G_{YX} : X \rightarrow Y \rightarrow X$ and $G_{YX} \circ G_{XY} : Y \rightarrow X \rightarrow Y$, the examples x and y in domain $\mathcal{X}$ and $\mathcal{Y}$ should recover the original images, *i.e.*, $x \approx G_{YX}(G_{XY}(x))$ and $y \approx G_{XY}(G_{YX}(y))$. 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:

$$\begin{aligned}\mathcal{L}_{cyc}(G_{XY}, G_{YX}, X, Y) = \mathbb{E}_{x\sim P_X}[\|G_{YX}(G_{XY}(x)) - x\|_1] \\ + \mathbb{E}_{y\sim P_Y}[\|G_{XY}(G_{YX}(y)) - y\|_1].\end{aligned} \tag{1}$$

Distance constraint. The assumption behind the distance constraint is that the distance between two examples x_i and x_j in domain X should be preserved after mapping to domain 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 differences between distances:

$$\begin{aligned}\mathcal{L}_{dis}(G_{XY}, X, Y) &= \mathbb{E}_{x_i,x_j\sim P_X}[|\phi(x_i, x_j) - \psi(x_i, x_j)|], \\ \phi(x_i, x_j) &= \frac{1}{\sigma_X}(\|x_i - x_j\|_1 - \mu_X), \\ \psi(x_i, x_j) &= \frac{1}{\sigma_Y}(\|G_{XY}(x_i) - G_{XY}(x_j)\|_1 - \mu_Y),\end{aligned} \tag{2}$$

where μ_X, μ_Y (σ_X, σ_Y) are the means (standard deviations) of distances of all the possible pairs of (x_i, x_j) within domain $\mathcal{X}$ and (y_i, y_j) within domain $\mathcal{Y}$, respectively, and are precomputed. Distance preservation makes one-sided unsupervised domain mapping possible.

4.2 Geometry-consistent Generative Adversarial Networks

Adversarial constraint. Taking G_{XY} as an example, an adversarial loss $\mathcal{L}_{gan}(G_{XY}, D_Y, X, Y)$ [19] enforces G_{XY} and D_Y to simultaneously optimize each other in an minimax game, *i.e.*, $\min_{G_{XY}} \max_{D_Y} \mathcal{L}_{gan}(G_{XY}, D_Y, X, Y)$. 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

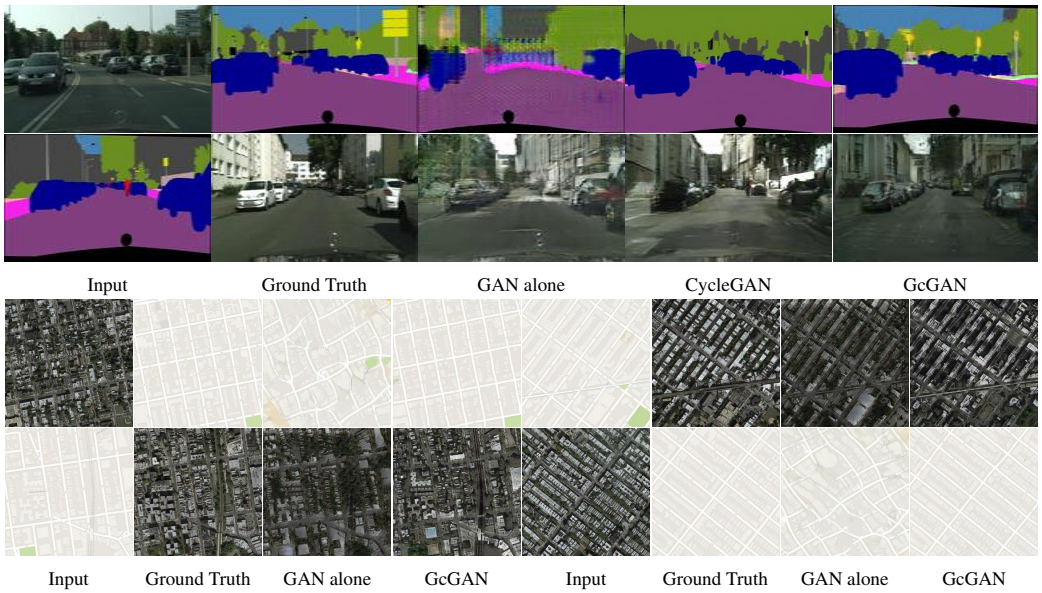


Figure 3: Qualitative comparison on Cityscapes (Parsing $\rightleftharpoons$ Image) and Google Maps (Map $\rightleftharpoons$ Aerial photo). GAN alone suffers from mode collapse. Translated images by GcGAN contain more details. $GcGAN = GAN\ alone + geometry\ consistency$.

fake example $y' = G_{XY}(x)$ as a sample satisfying $y' \sim P_Y$. The objective can be expressed as:

$$\begin{aligned}\mathcal{L}_{gan}(G_{XY}, D_Y, X, Y) = &\mathbb{E}_{y\sim P_Y}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim P_X}[\log(1 - D_Y(G_{XY}(x)))].\end{aligned} \tag{3}$$

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

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 $\tilde{y}' \approx f(y')$ like x and $\tilde{x}$. Considering both $f(\cdot)$ and the inverse geometric transformation function $f^{-1}(\cdot)$, our complete geometry consistency loss $\mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y)$ has the following form:

$$\begin{aligned}\mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y) = &\mathbb{E}_{x\sim P_X}[\|G_{XY}(x) - f^{-1}(G_{\tilde{X}\tilde{Y}}(f(x)))\|_1] \\ &+ \mathbb{E}_{x\sim P_X}[\|G_{\tilde{X}\tilde{Y}}(f(x)) - f(G_{XY}(x))\|_1].\end{aligned} \tag{4}$$

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 (*vf*) 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 $\mathcal{L}_{GcGAN}(G_{XY}, G_{\tilde{X}\tilde{Y}}, D_Y, D_{\tilde{Y}}, X, Y)$ will be:

$$\begin{aligned}\mathcal{L}_{GcGAN}(G_{XY}, G_{\tilde{X}\tilde{Y}}, D_Y, D_{\tilde{Y}}, X, Y) = &\mathcal{L}_{gan}(G_{XY}, D_Y, X, Y) \\ &+ \mathcal{L}_{gan}(G_{\tilde{X}\tilde{Y}}, D_{\tilde{Y}}, X, Y) \\ &+ \lambda\mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y),\end{aligned} \tag{5}$$

where λ ($\lambda = 20.0$ in all the experiments) is a trade-off hyperparameter to weight the contribution of $\mathcal{L}_{gan}$ and $\mathcal{L}_{geo}$ during the model training. Because that we do not make great effects to choose λ,

method	image → parsing			parsing → image		
	pixel acc	class acc	mean IoU	pixel acc	class acc	mean IoU
Benchmark Performance						
CoGAN [37]	0.45	0.11	0.08	0.40	0.10	0.06
BiGAN/ALI [15, 16]	0.41	0.13	0.07	0.19	0.06	0.02
SimGAN [51]	0.47	0.11	0.07	0.20	0.10	0.04
CycleGAN (Cycle) [62]	**0.58**	0.22	0.16	0.52	0.17	0.11
DistanceGAN [5]	-	-	-	0.53	0.19	0.11
GAN alone (baseline)	0.514	0.160	0.104	0.437	0.161	0.098
GcGAN-*rot*	0.574	**0.234**	0.170	**0.551**	**0.197**	**0.129**
GcGAN-*vf*	0.576	0.232	**0.171**	0.548	0.196	0.127
Ablation Studies (Robustness & Compatibility)						
GcGAN-*rot*-Seperate	0.575	0.233	0.170	0.545	0.196	0.124
GcGAN-Mix	0.573	0.229	0.168	0.545	0.197	0.128
GcGAN-*rot* + Cycle	**0.587**	**0.246**	**0.182**	**0.557**	**0.201**	**0.132**

Table 1: **Parsing scores on Cityscapes.** GcGAN-emphrot-Separate: G_{XY} and $G_{\tilde{X}\tilde{Y}}$ do not share parameters. GcGAN-Mix: GcGAN with a mixture of transformations (*rot* and *vf*). GcGAN-*rot* + Cycle: GcGAN-*rot* with the cycle-consistency constraint.

heavily tuning λ 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 [20] (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 [24, 31]. Like CycleGAN, we also use an identity mapping loss [52] 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 [54] to normalize convolutional feature maps.

Learning and inference. We use the Adam solver [27] 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 [40] for $\mathcal{L}_{gan}$. The discriminator is updated with random samples from a history of generated images stored in an image buffer [51] 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 [62]

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 128×128 in an unaligned fashion. We evaluate our domain mappers using FCN scores and scene parsing metrics following previous works [38, 12, 62]. 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 [38] provided by pix2pix [24] to predict semantic labels for the 500 translated images. The label maps are then resized to the original resolution of 1024×2048 and compared against the ground truth labels using some standard scene parsing metrics including pixel accuracy, class accuracy, and mean IoU [38]. 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 for both image $\rightarrow$ parsing and parsing $\rightarrow$ image tasks 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 [61], *i.e.*, score $=$ $(\text{pixel acc} + \text{class acc} + \text{mean IoU})/3$. 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 the state-of-the-art approach distanceGAN (27.7%).

We next perform ablation studies to investigate the robustness and compatibility of GcGAN, including GcGAN-*rot*-Seperate, GcGAN-Mix, and GcGAN-*rot* + Cycle. The scores are reported in Table 1. Specifically, GcGAN-*rot*-Seperate shows that the generator G_{XY} employed in GcGAN is sufficient to handle both the style transfers (without shape deformation) $X \rightarrow Y$ and $\tilde{X} \rightarrow \tilde{Y}$. GcGAN-Mix demonstrates 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. For GcGAN-*rot* + Cycle, we set the trade-off parameter for $\mathcal{L}_{cyc}$ 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.

method	class acc (%)
Benchmark Performance	
DistanceGAN (Dist.) [5]	26.8
CycleGAN (Cycle) [62]	26.1
Self-Distance [5]	25.2
GcGAN-*rot*	32.5
GcGAN-*vf*	**33.3**
Ablation Studies (Compatibility)	
Cycle + Dist. [5]	18.0
GcGAN-*rot* + Dist.	**34.0**
GcGAN-*rot* + Cycle	33.8
GcGAN-*rot* + Dist. + Cycle	33.2

Table 2: Quantitative scores for SVHN $\rightarrow$ MNIST.

SVHN $\rightarrow$ MNIST. We then apply our approach to the SVHN $\rightarrow$ MNIST translation task. The translation models are trained on 73257 and 60000 training images of resolution 32×32 contained in the SVHN and MNIST training sets, respectively. The experimental settings follow DistanceGAN [5], including the default trade-off parameters for $\mathcal{L}_{cyc}$ and $\mathcal{L}_{dis}$, and the network architectures for the generators and the discriminators. We compare our GcGAN 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]. Note that, the experimental settings for domain mapping (GcGAN) and domain adaptation are totally different, so is the captured

classification accuracy. Domain adaptation methods have access to the source domain digit labels while image translation does not.

Classification accuracies are reported in Table 2. Both GcGAN-*rot* and GcGAN-*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 constraint is not compatible with the cycle-consistency constraint, but our geometry-consistency constraint can improve both ones.

DistanceGAN [5] GcGAN

Figure 4: Qualitative comparison for SVHN $\rightarrow$ MNIST.

Google Maps. We obtain 2194 (map, aerial photo) pairs of images in and around New York City from Google Maps [24], and split them into training and test sets with 1096 and 1098 pairs, respectively. We train Map $\rightleftharpoons$ Aerial photo translators with an image size of 256×256 using the training set in an unsupervised manner (unpaired) by ignoring the pair information. For Aerial photo $\rightarrow$ 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 $\rightarrow$ Aerial photo, we only show some qualitative results in Figure 3.

method	RMSE	acc (δ_1)	acc (δ_2)
Benchmark Performance			
CycleGAN [62]	**28.15**	**41.8**	**63.7**
GAN alone (baseline)	33.27	19.3	42.0
GcGAN-*rot*	28.31	41.2	63.1
GcGAN-*vf*	28.50	37.3	58.9
Ablation Studies (Robustness & Compatibility)			
GcGAN-*rot*-Separate	30.25	40.7	60.8
GcGAN-Mix	**27.98**	**42.8**	**64.6**
GcGAN-*rot* + Cycle	28.21	40.6	63.5

Table 3: Quantitative scores for Aerial photo $\rightarrow$ Map.

From the scores presented in Table 3, it can be seen that 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, our one-sided GcGANs achieve competitive even slightly better scores compared with the two-sided CycleGAN.

5.2 Qualitative Evalutation

The qualitative results are shown in Figure 3, Figure 4, and Figure 5. While GAN alone suffers from mode collapse, our geometry-consistency constraint can provide an effective remedy, thus helps to generate empirically more impressive translations on various applications. The following applications are trained in the image size of 256×256 with the *rot* geometric transformation.

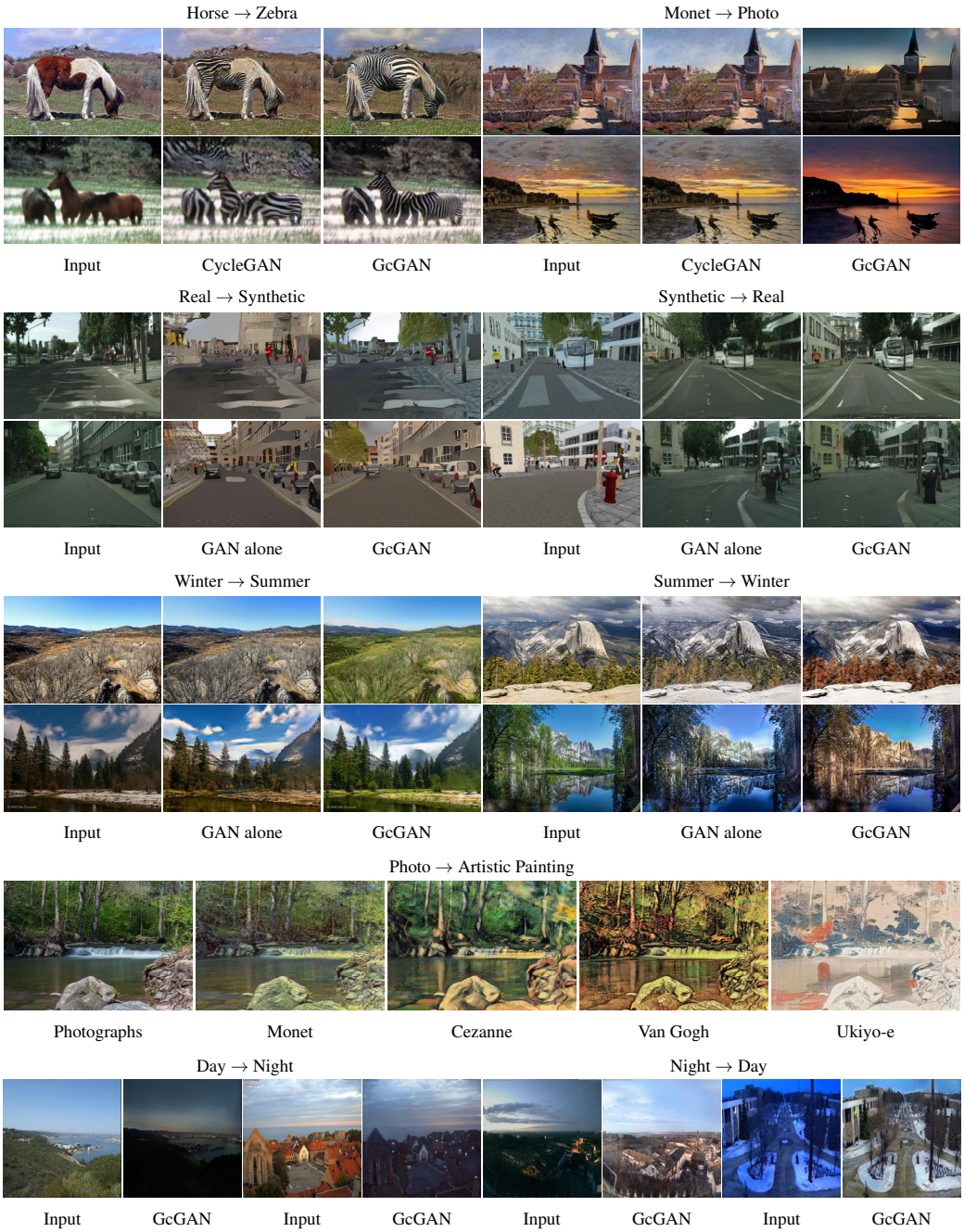


Figure 5: Qualitative results on different applications, including Horse → Zebra, Monet → Photo, Synthetic ⇌ Real, Summar ⇌ Winter, Photo → Artist Painting, and Day ⇌ Night. GcGAN has the potential to produce realistic images. Zoom in for better view.

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 [46], which is a virtual urban scene benchmark, as the synthetic images.

Summer $\rightleftharpoons$ 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 $\rightleftharpoons$ 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 $\rightarrow$ 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 $\rightleftharpoons$ Night. We randomly extract 4500 training images for both Day and Night from the 91 webcam sequences captured by [28].

6 Conclusion

In this paper, we propose to enforce geometry consistency as a constraint, which can be viewed as a predefined geometric transformation $f(\cdot)$ preserving the geometry of a scene for unsupervised domain mapping. 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. Our 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.

References

[1] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *ICML*, 2018.

[2] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool. Combogan: Unrestrained scalability for image domain translation. In *CVPRW*, 2018.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell. Multi-content gan for few-shot font style transfer. In *CVPR*, 2018.

[5] S. Benaim and L. Wolf. One-sided unsupervised domain mapping. In *NIPS*, 2017.

[6] S. Benaim and L. Wolf. One-shot unsupervised cross domain translation. *NIPS*, 2018.

[7] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016.

[8] H. Chang, J. Lu, F. Yu, and A. Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *CVPR*, 2018.

[9] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stereoscopic neural style transfer. In *CVPR*, 2018.

[10] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017.

[11] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR)*, 2016.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a? laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[15] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[16] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[17] A. Gokaslan, V. Ramanujan, D. Ritchie, K. I. Kim, and J. Tompkin. Improving shape deformation in unsupervised image-to-image translation. *ECCV*, 2018.

[18] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. *NIPS*, 2018.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *ICML*, 2018.

[22] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *ECCV*, 2018.

[23] T. Isokane, F. Okura, A. Ide, Y. Matsushita, and Y. Yagi. Probabilistic plant modeling via multi-view image-to-image translation. *CVPR*, 2018.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[25] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[26] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM TOG*, 33(4):149, 2014.

[29] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[30] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[31] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016.

[32] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang. Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks. *ECCV*, 2018.

[33] X. Liang, H. Zhang, and E. P. Xing. Generative semantic manipulation with contrasting gan. *NIPS*, 2017.

[34] J. Lin, Y. Xia, T. Qin, Z. Chen, and T.-Y. Liu. Conditional image-to-image translation. In *CVPR*, 2018.

[35] A. Liu, Y.-C. Liu, Y.-Y. Yeh, and Y.-C. F. Wang. A unified feature disentangler for multi-domain image translation and manipulation. *NIPS*, 2018.

[36] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[37] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[38] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[39] S. Ma, J. Fu, C. W. Chen, and T. Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *CVPR*, 2018.

[40] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[41] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim. Unsupervised attention-guided image to image translation. *NIPS*, 2018.

[42] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

[43] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[44] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[45] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[46] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[47] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Moressi, F. Cole, and K. Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. *ICLR*, 2018.

[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

[49] F. Shen, S. Yan, and G. Zeng. Neural style transfer via meta networks. In *CVPR*, 2018.

[50] L. Sheng, Z. Lin, J. Shao, and X. Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, 2018.

[51] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[52] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation, 2016.

[53] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *ICML*, 2016.

[54] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[55] C. Wang, H. Zheng, Z. Yu, Z. Zheng, Z. Gu, and B. Zheng. Discriminative region proposal adversarial networks for high-quality image-to-image translation. In *ECCV*, 2018.

[56] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

[57] Y. Wang, J. van de Weijer, and L. Herranz. Mix and match networks: encoder-decoder alignment for zero-pair image translation. In *CVPR*, 2018.

[58] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *CVPR*, 2017.

[59] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[60] Y. Zhang, Y. Zhang, and W. Cai. Separating style and content for generalized style transfer. In *CVPR*, 2018.

[61] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[62] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*, 2017.

[63] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017.

Network Architecture

The generator and discriminator (except for SVHN $\rightarrow$ 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).

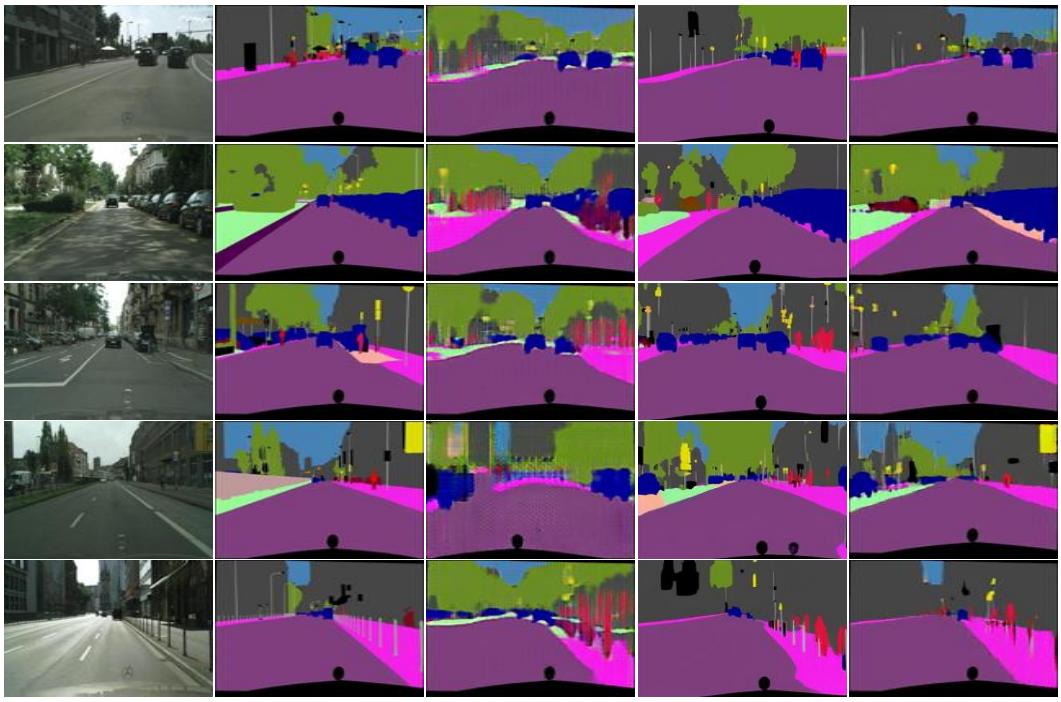
Generator				
Index	Layer	C	K	S
1	Conv + ReLU	64	7	1
2	Conv + ReLU	128	3	2
3	Conv + ReLU	256	3	2
4	ResBlk + ReLU	256	3	1
5	ResBlk + ReLU	256	3	1
6	ResBlk + ReLU	256	3	1
7	ResBlk + ReLU	256	3	1
8	ResBlk + ReLU	256	3	1
9	ResBlk + ReLU	256	3	1
10	ResBlk + ReLU	256	3	1
11	ResBlk + ReLU	256	3	1
12	ResBlk + ReLU	256	3	1
12	Deconv + ReLU	128	3	2
13	Deconv + ReLU	64	3	2
14	Conv	3	7	1
15	Tanh	-	-	-
Discriminator				
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 $\rightarrow$ MNIST is reported in Tab. 5.

Table 5: The network architecture for SVHN $\rightarrow$ MNIST.

Generator				
Index	Layer	C	K	S
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
5	Deconv + LeakyReLU	1	4	2
15	Tanh	-	-	-
Discriminator				
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

Photo → Parsing



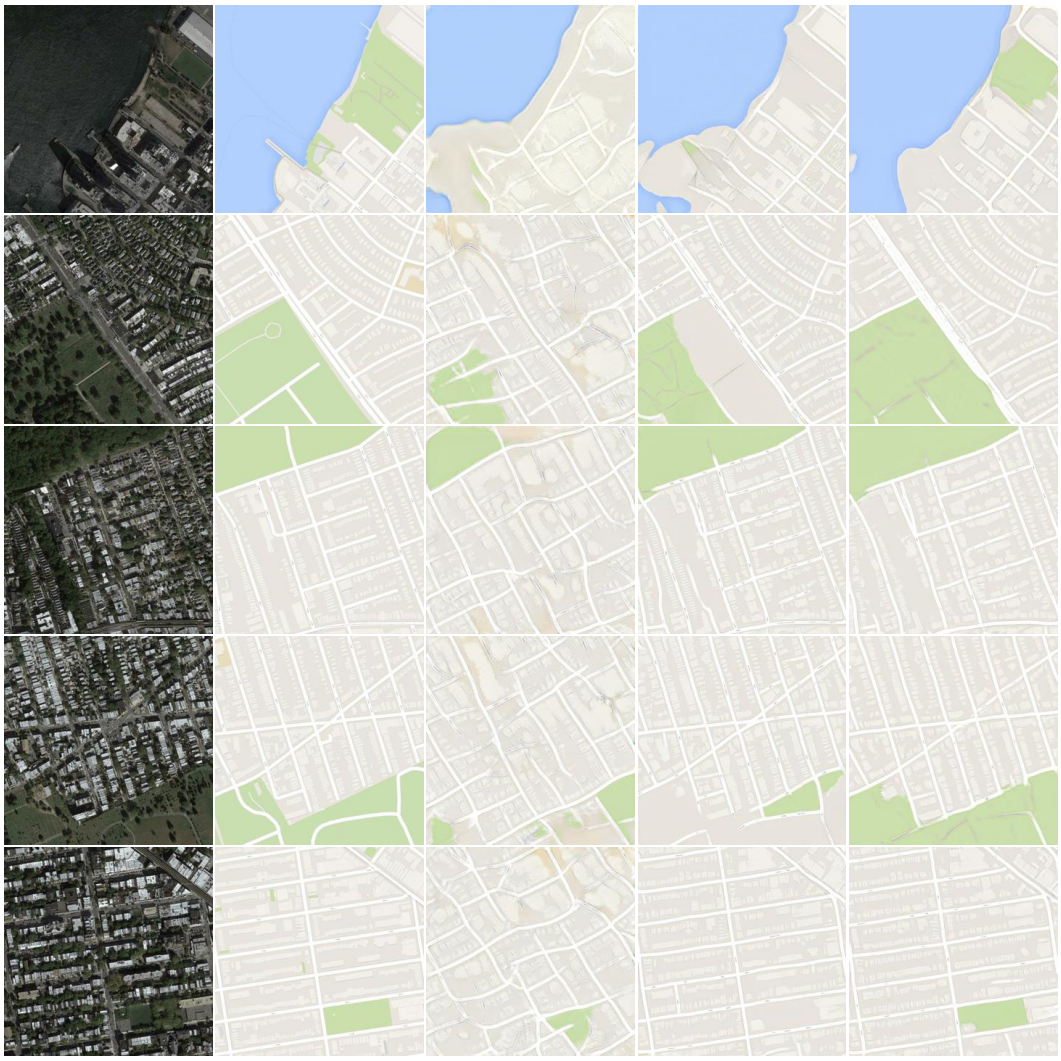
Input | Ground Truth | GAN alone | CycleGAN | GcGAN

Parsing → Photo

Input | Ground Truth | GAN alone | CycleGAN | GcGAN

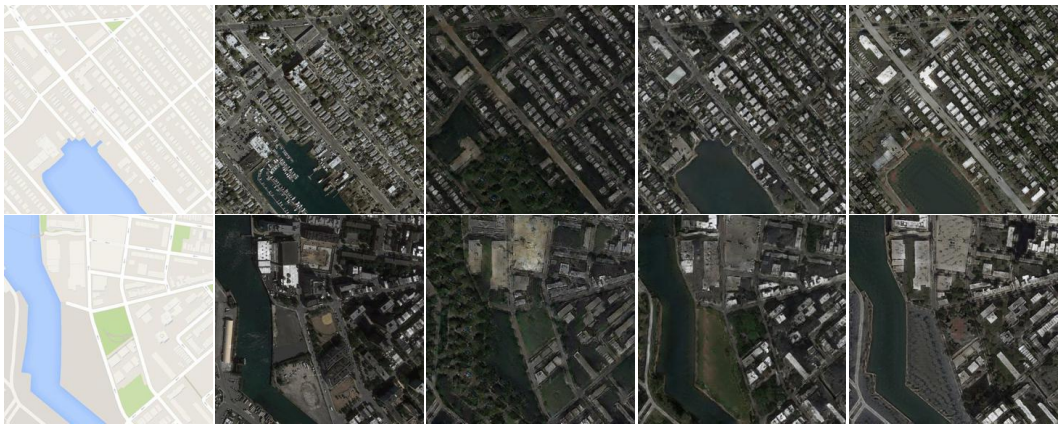
Figure 6: **Cityscapes (Parsing ⇌ Image).** The results for CycleGAN [62] are produced by the officially provided pretrained PyTorch models. GcGAN denotes GcGAN-*rot*.

Aerial photo → Map



Input | Ground Truth | GAN alone | CycleGAN | GcGAN

Map → Aerial photo



Input | Ground Truth | GAN alone | CycleGAN | GcGAN

Figure 7: **Google Mpas (Aerial photo ⇌ Map).** For Map → Aerial photo, GcGAN produces competitive translations compared with CycleGAN [62].

SVHN → MNIST

CycleGAN (24/64)

DistanceGAN (26/64)

GcGAN (35/64)

Figure 8: **SVHN → 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 GcGAN, respectively.

Horse → Zebra

Input GAN alone CycleGAN GcGAN

Figure 9: **Horse → Zebra.** For this task, GcGAN generates slightly better translations for some images, but can not perform better than CycleGAN [62] generally.

Monet → Photo

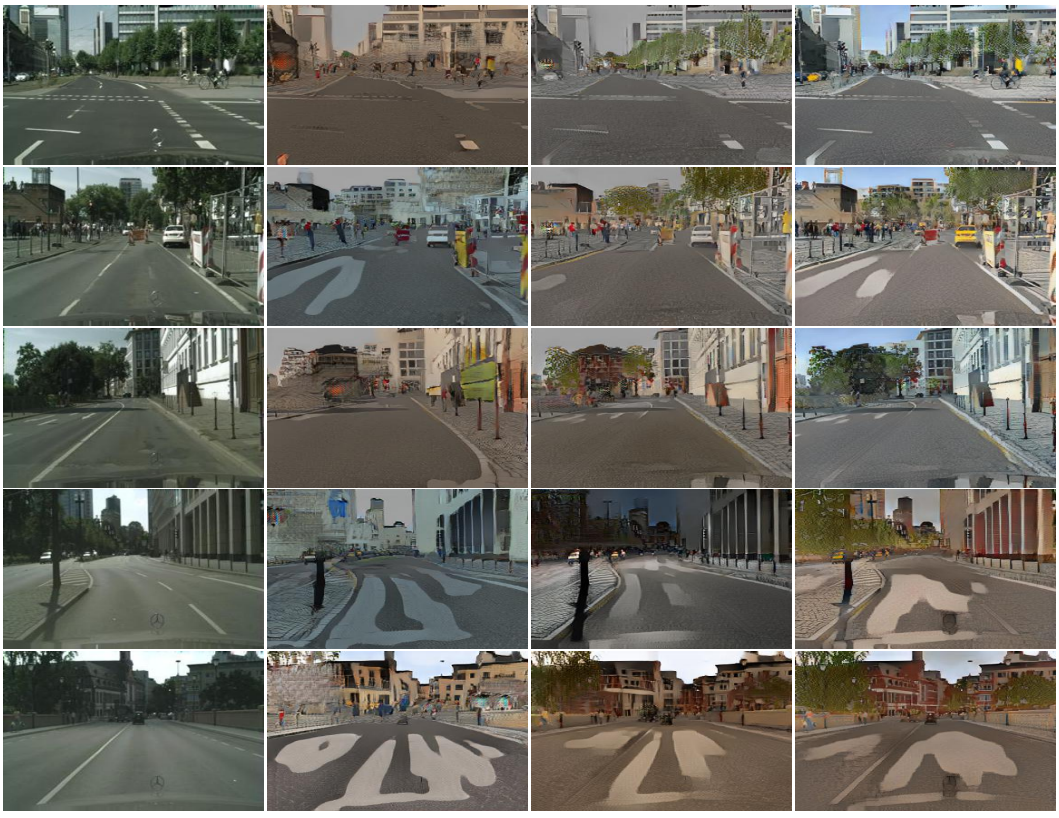
Input GAN alone CycleGAN GcGAN

Figure 10: **Monet → Photo.** GcGAN is superior in generating realistic images. Zoom in for better view.

Synthetic → Real

Input GAN alone CycleGAN GcGAN

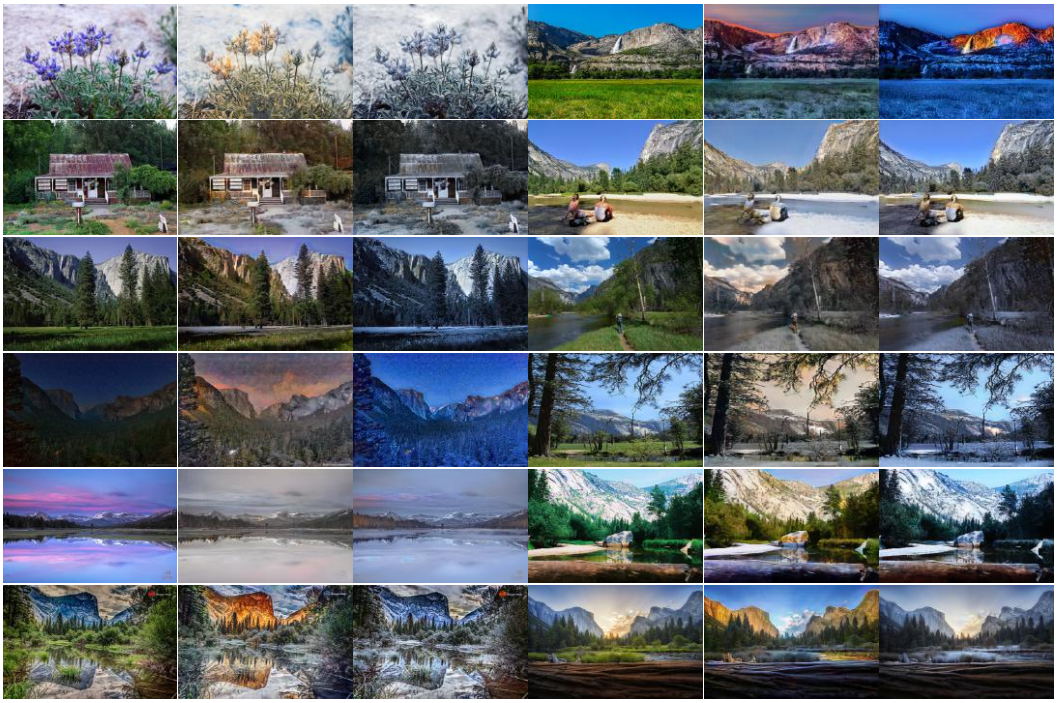
Real → Synthetic



Input GAN alone CycleGAN GcGAN

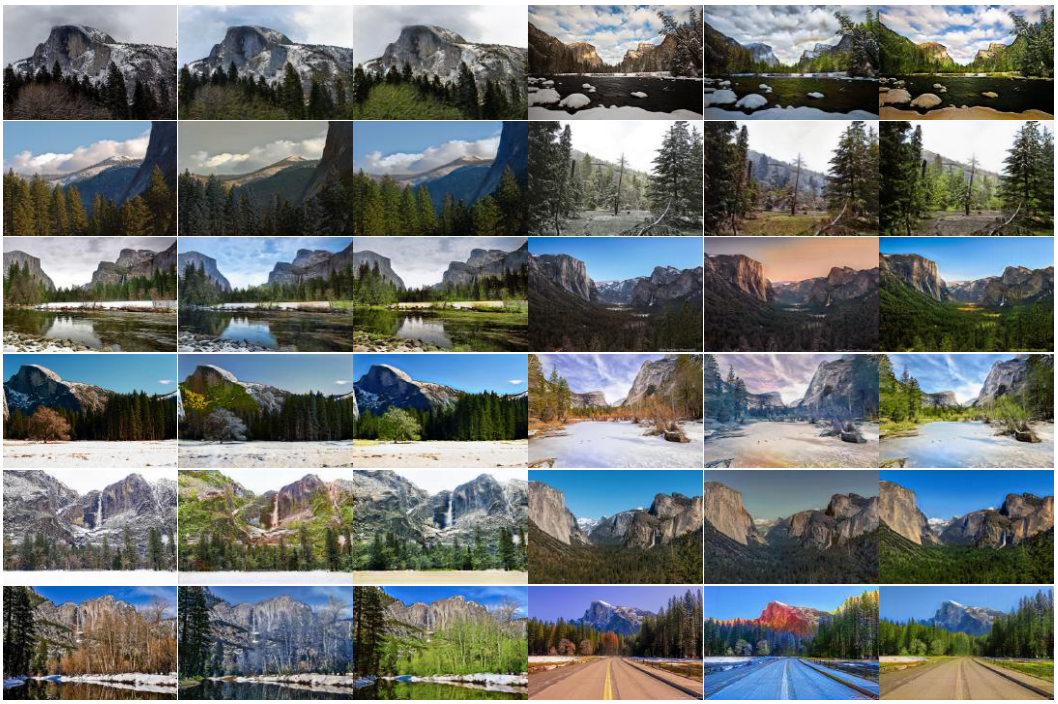
Figure 11: **Synthetic ⇌ Real.** We train CycleGAN [62] using the released PyTorch codes. The results produced by GcGAN contain more details. Zoom in for better view.

Summer → Winter



Input CycleGAN GcGAN Input CycleGAN GcGAN

Winter → Summer



Input CycleGAN GcGAN Input CycleGAN GcGAN

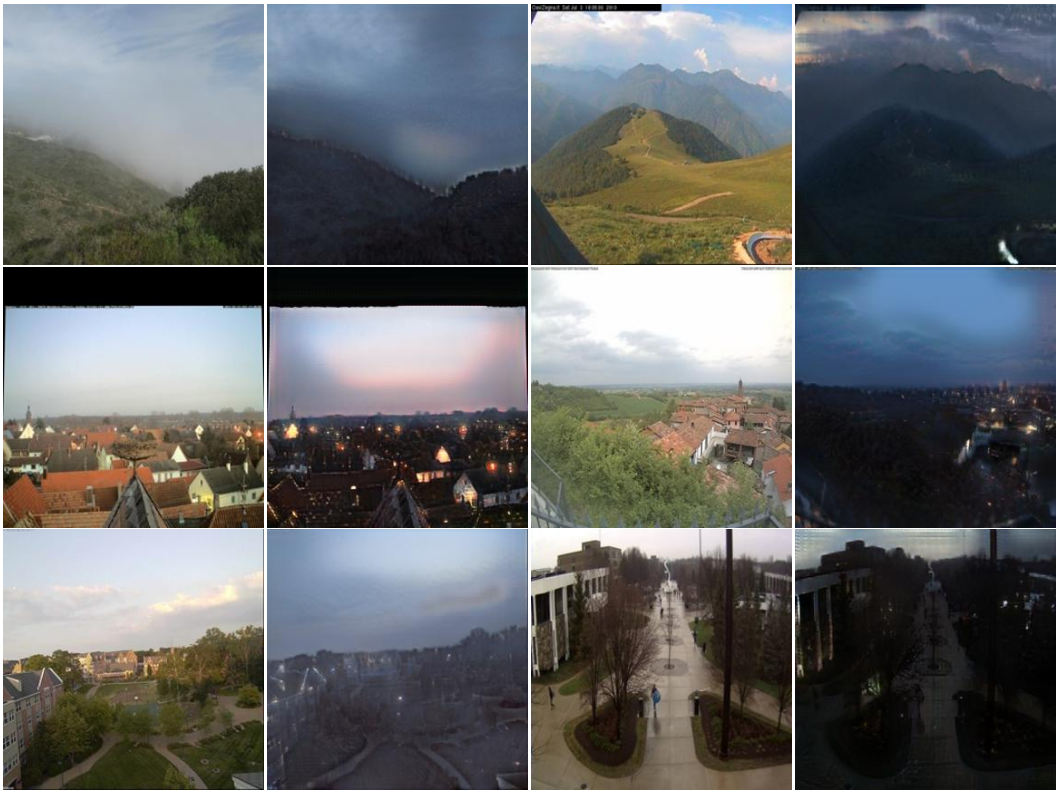
Figure 12: **Summer ⇌ Winter.** Here, GcGAN represents GcGAN-*rot*-Seperate. Zoom in for better view.

Photographs → Artist paintings

Input | Monet | Van Gogh | Cezanne | U-kiyoe

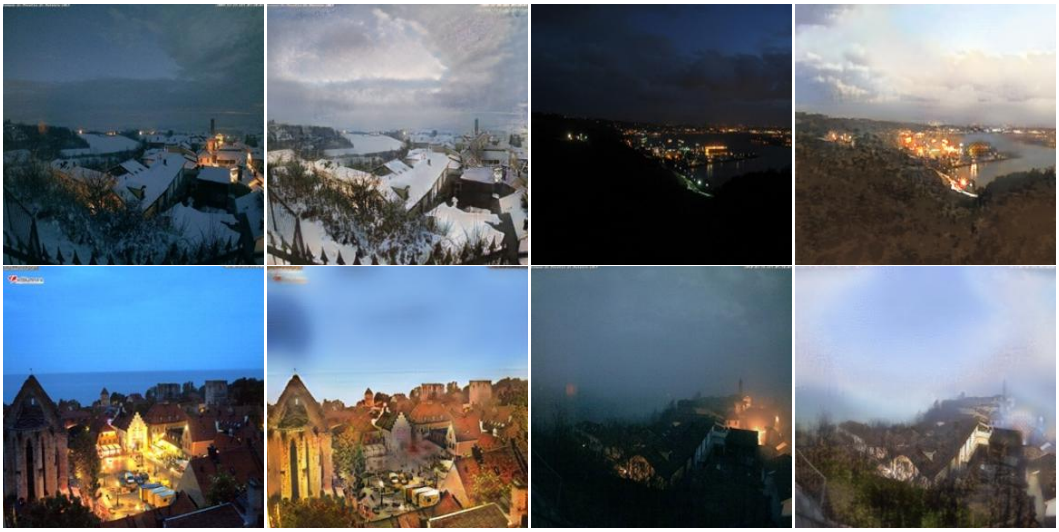
Figure 13: **Photographs → Artist paintings.** We translate a photo to the artistic styles of Monet, Van Gogh, Cezanne, and U-kiyoe.

Day → Night



Input GcGAN Input GcGAN

Night → Day



Input GcGAN Input GcGAN

Figure 14: **Day ⇌ Night.**

Failure Cases

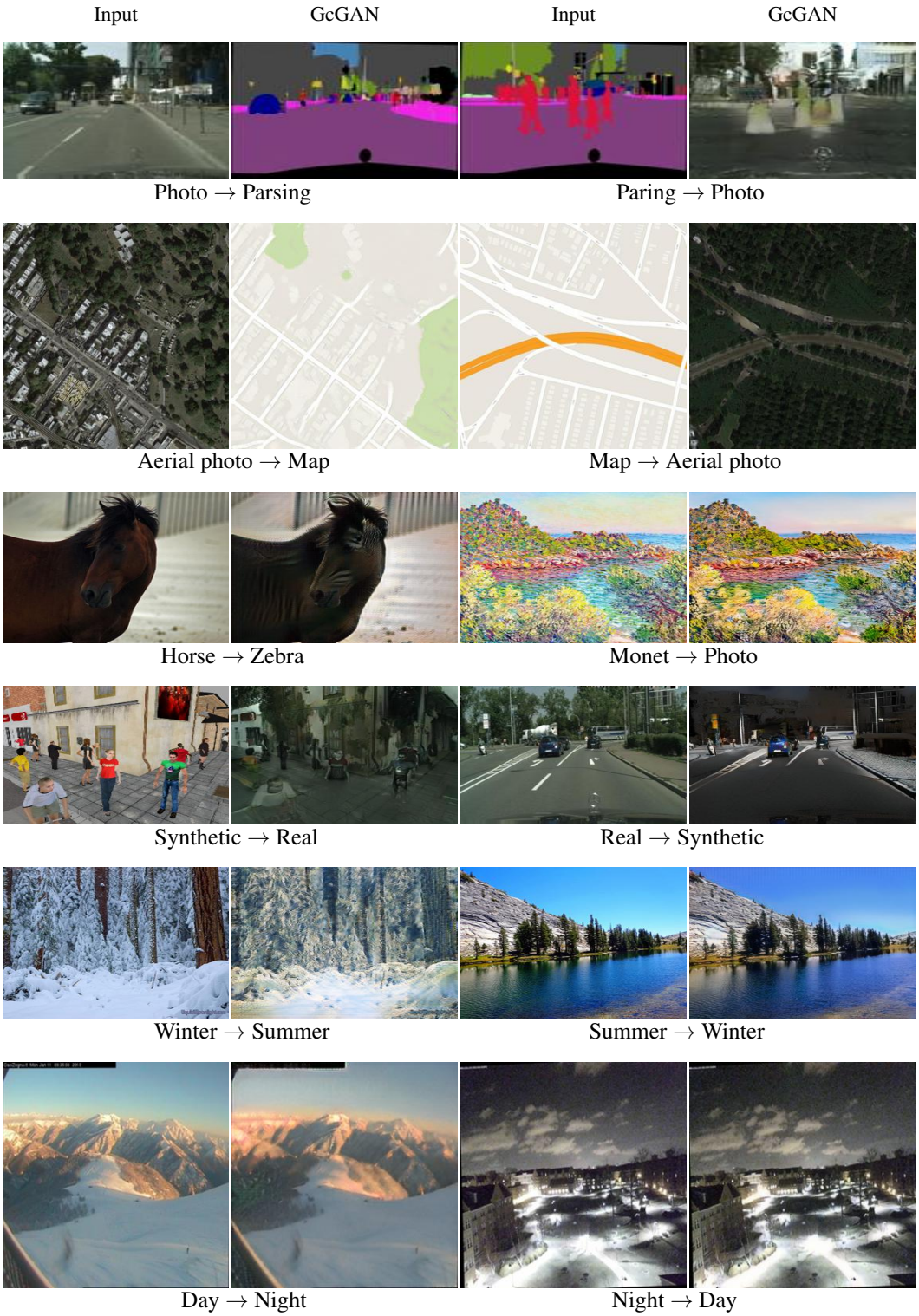


Figure 15: **Failure Cases.** GcGAN cannot guarantee reasonable translations for all the cases as previous works. Thus, more assumptions and constraints should be investigated to improve unsupervised domain mapping.