

# XGAN: Unsupervised Image-to-Image Translation for Many-to-Many Mappings



Amélie Royer , Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole and Kevin Murphy

**Abstract** Image translation refers to the task of mapping images from a visual domain to another. Given two unpaired collections of images, we aim to learn a mapping between the corpus-level style of each collection, while preserving semantic content shared across the two domains. We introduce XGAN, a dual adversarial auto-encoder, which captures a shared representation of the common domain semantic content in an unsupervised way, while jointly learning the domain-to-domain image translations in both directions. We exploit ideas from the domain adaptation literature and define a *semantic consistency loss* which encourages the learned embedding to preserve semantics shared across domains. We report promising qualitative results for the task of face-to-cartoon translation. The cartoon dataset we collected for this purpose, “CartoonSet”, is also publicly available as a new benchmark for semantic style transfer at <https://google.github.io/cartoonset/index.html>.

**Keywords** Generative models · Style transfer · Domain adaptation

---

A. Royer—Work done while at Google Brain London, UK.

---

A. Royer (✉)  
IST Austria, 3400 Klosterneuburg, Austria  
e-mail: [aroyer@ist.ac.at](mailto:aroyer@ist.ac.at)

K. Bousmalis · S. Gouws  
Google Brain, London, UK  
e-mail: [konstantinos@google.com](mailto:konstantinos@google.com)

S. Gouws  
e-mail: [sgouws@google.com](mailto:sgouws@google.com)

K. Bousmalis  
Deepmind, London, UK

F. Bertsch  
Google Brain, Mountain View, CA, USA

I. Mosseri · F. Cole  
Google Research, Cambridge, MA, USA

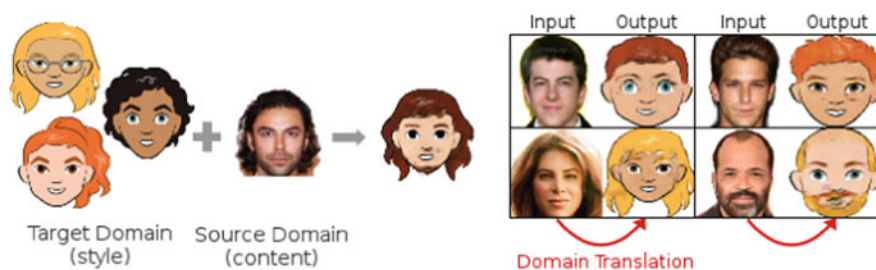
K. Murphy  
Google Research, Mountain View, CA, USA

## 1 Introduction

Image-to-image translation—learning to map images from one domain to another—covers several classical computer vision tasks such as style transfer (rendering an image in the style of a given input [4]), colorization (mapping grayscale images to color images [26]), super-resolution (increasing the resolution of an input image [13]), or semantic segmentation (inferring pixel-wise semantic labeling of a scene [18]). Learning such mappings requires an underlying understanding of the shared information between the two domains. In many cases, supervision encapsulates this knowledge in the form of labels or paired samples. This holds, for instance, for colorization, where ground-truth pairs are easily obtained by generating grayscale images from colored inputs.

In this work, we consider the task of *unsupervised semantic style transfer*: learning to map an image from one domain into the style of another domain without altering its semantic content (see Fig. 1). In particular, we experiment on the task of translating faces to cartoons. Note that without loss of generality, a photo of a face can be mapped to many valid cartoons, and vice-versa. Semantic style transfer is, therefore, a *many-to-many mapping* problem, for which obtaining labeled examples are ambiguous and costly. Furthermore, in this unsupervised setting we do not have access to supervision on shared domain semantic content (e.g., facial attributes such as hair color, eye color, etc.). Instead, we propose an encoder–decoder structure with a bottleneck embedding shared across the two domains to capture common semantics as a latent representation.

The key issue is thus to learn an embedding that preserves semantic facial attributes (hair color, eye color, etc.) between the two domains with little supervision, and to incorporate it within a generative model to produce the actual domain translations. Although this chapter specifically focuses on the face-to-cartoon setting, many other



**Fig. 1** Semantic style transfer is the task of adapting an image to the visual appearance of another domain without altering its semantic content given only two unpaired image collections without pairs supervision (*left*). We define semantic content as characteristic attributes which are shared across domains but do not necessarily appear the same at the pixel-level. For instance, cartoons and faces have a similar range of hair color but with very different appearances, e.g., blonde hair is bright yellow in cartoons. The proposed XGAN applied on the face-to-cartoon task yields a shared representation that preserves important face semantics such as hair style or face shape (*right*)

examples fall under this category: mapping landscape pictures to paintings (where the different scene objects and their composition describe the input semantics), transforming sketches to images, or even cross-domain tasks such as generating images from text. We only rely on two unlabeled training image collections or *corpora*, one for each domain, with no known image pairings across domains. Hence, we are faced with a double *domain shift*, first in terms of global domain appearance, and second in terms of the content distribution of the two collections.

Recent work [1, 6, 10, 25, 27] report good performance using GAN-based models for unsupervised image-to-image translation when the two input domains share similar pixel-level structure (e.g., horses and zebras) but fail for more significant domain shifts (e.g., dogs and cats). Perhaps the best known recent example is CycleGAN [27]. Given two image domains  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , the model is trained with a pixel-level *cycle-consistency loss* which ensures that the mapping  $g_{1 \rightarrow 2}$  from  $\mathcal{D}_1$  to  $\mathcal{D}_2$  followed by its inverse,  $g_{2 \rightarrow 1}$ , yields the identity function; i.e.,  $g_{1 \rightarrow 2} \circ g_{2 \rightarrow 1} = id$ . We argue that such a pixel-level constraint is not sufficient in our setting and that we rather need a constraint in *feature space* to allow for more permissive transformations of the pixel input. To this end, we propose XGAN (“Cross-GAN”), a dual adversarial auto-encoder which learns a shared semantic representation of the two input domains in an unsupervised way, while jointly learning both domain-to-domain translations. More specifically, the domain translation  $g_{1 \rightarrow 2}$  consists of an encoder  $e_1$  taking inputs in  $\mathcal{D}_1$ , followed by a decoder  $d_2$  with outputs in  $\mathcal{D}_2$  (and likewise for  $g_{2 \rightarrow 1}$ ) such that  $e_1$  and  $e_2$ , as well as  $d_1$  and  $d_2$ , are partially shared across domains.

The main novelty lies in how we constrain the shared embedding using techniques from the domain adaptation literature, as well as a novel *semantic consistency loss*. The latter ensures that the domain-to-domain translations preserve the semantic representation, i.e.,  $e_1 \approx e_2 \circ g_{1 \rightarrow 2}$  and  $e_2 \approx e_1 \circ g_{2 \rightarrow 1}$ . Therefore, it acts as a form of self-supervision which alleviates the need for paired examples and preserves semantic feature-level information rather than pixel-level content. In the following section, we review relevant recent work before discussing the XGAN model in more detail in Sect. 3. In Sect. 4, we introduce CARTOONSET, our dataset of cartoon faces for research on semantic style transfer. Finally, in Sect. 5 we report experimental results of XGAN on the face-to-cartoon task.

## 2 Related Work

Recent literature suggests two main directions for tackling the semantic style transfer task: traditional style transfer and pixel-level domain adaptation. The first approach is inadequate as it only transfers texture information from a single style image, and therefore does not capture the style of an entire corpus. The latter category also fails in practice as it explicitly enforces pixel-level similarity which does not allow for significant structural change of the input. Instead, we draw inspiration from the domain adaptation and feature-level image-to-image translation literature.

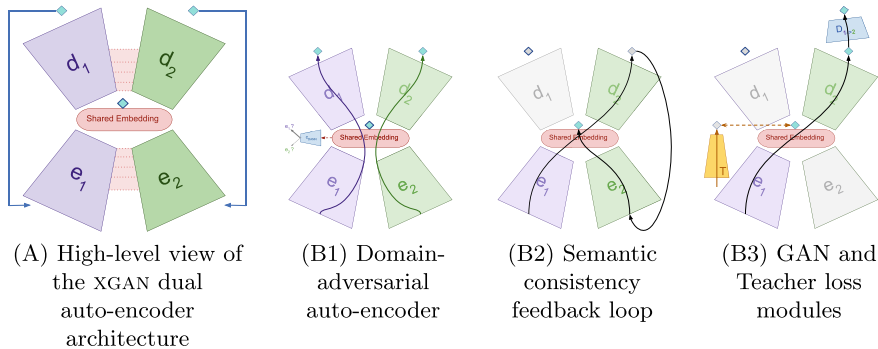
*Style Transfer.* Neural style transfer refers to the task of transferring the texture of a *specific* style image while preserving the pixel-level structure of an input content image [4, 9]. Recently, [14, 15] proposed to instead use a dense local patch-based matching approach in the feature space, as opposed to global feature matching, allowing for convincing transformations between visually dissimilar domains. Still, these models only perform image-specific transfer rather than learning a global *corpus-level* style and do not provide a meaningful shared domain representation. Furthermore, the generated images are usually very close to the original input in terms of pixel structure (e.g., edges) which is not suitable for drastic transformations such as face-to-cartoon.

*Domain adaptation.* XGAN relies on learning a shared feature representation of both domains in an unsupervised setting to capture semantic rather than pixel information. For this purpose, we make use of the domain-adversarial training scheme [3]. Moreover, recent domain adaptation work [1, 2, 22] can be framed as semantic style transfer as they tackle the problem of mapping synthetic images, easy to generate, to natural images, which are more difficult to obtain. The generated samples are then used to train a model later applied to natural images. Contrary to our work, however, they only consider pixel-level transformations.

*Unsupervised Image-to-Image translation.* Recent work [6, 10, 25, 27] tackle the unsupervised pixel-level image-to-image translation task by learning both cross-domain mappings jointly, each as a separate generative adversarial network, via a cycle-consistency loss which ensures that applying each mapping followed by its reverse yields the identity function. This intuitive form of self-supervision leads to good results for pixel-level transformations but often fails to capture significant structural changes [27]. In comparison, our proposed semantic consistency loss acts at the feature-level, allowing for more flexible transformations.

Orthogonal to this line of work is UNIT [7, 16, 19]. This model consists of a coupled VAEGAN architecture [12, 17] with a shared embedding bottleneck, trained with pixel-level cycle-consistency. Similar to XGAN, it learns a joint *feature-level* representation of the two domains, however, UNIT assumes that sharing high-level layers in the architecture is a sufficient constraint, while XGAN’s objective explicitly introduces the semantic consistency component.

Finally, the *Domain Transfer Network* (DTN) [23, 24] is closest to our work in terms of objective and applications. The DTN architecture is a single auto-encoder trained to map images from a source to a target domain with self-supervised semantic consistency feedback. It was also successfully applied to the problem of feature-level image-to-image translation, in particular to the face-to-cartoon problem. Contrary to XGAN however, the DTN encoder is pretrained and fixed, and is assumed to produce meaningful embeddings for both the face and the cartoon domains. This assumption is very restrictive, as off-the-shelf models pretrained on natural images do not usually generalize well to other domains. In fact, we show in Sect. 5 that a fixed encoder does not generalize well in the presence of a large domain shift between the two domains.



**Fig. 2** The XGAN (A) objective encourages the model to learn a meaningful joint embedding (B1) ( $\mathcal{L}_{rec}$  and  $\mathcal{L}_{dann}$ ), which should be preserved through domain translation (B2) ( $\mathcal{L}_{sem}$ ), while producing output samples of good quality (B3) ( $\mathcal{L}_{gan}$  and  $\mathcal{L}_{teach}$ )

### 3 Proposed Model: XGAN

Let  $\mathcal{D}_1$  and  $\mathcal{D}_2$  be two domains that differ in terms of *visual appearance* but share common *semantic content*. It is often easier to think of domain semantics as a high-level notion, e.g., semantic attributes, however, we do not require such annotations in practice, but instead consider learning a feature-level representation that automatically captures these shared semantics. Our goal is thus to learn in an unsupervised fashion, i.e., without paired examples, a joint domain-invariant embedding: semantically similar inputs across domains will be embedded nearby in the learned feature space.

Architecture-wise, XGAN is a dual auto-encoder on domains  $\mathcal{D}_1$  and  $\mathcal{D}_2$  Fig. 2A. We denote by  $e_1$  the encoder and by  $d_1$  the decoder for domain  $\mathcal{D}_1$ ; likewise  $e_2$  and  $d_2$  for  $\mathcal{D}_2$ . For simplicity, we also denote by  $g_{1 \rightarrow 2} = d_2 \circ e_1$  the transformation from  $\mathcal{D}_1$  to  $\mathcal{D}_2$ ; likewise  $g_{2 \rightarrow 1}$  for  $\mathcal{D}_2$  to  $\mathcal{D}_1$ .

The training objective can be decomposed into five main components: the *reconstruction* loss,  $\mathcal{L}_{rec}$ , encourages the learned embedding to encode meaningful knowledge for each domain; the *domain-adversarial* loss,  $\mathcal{L}_{dann}$ , pushes embeddings from  $\mathcal{D}_1$  and  $\mathcal{D}_2$  to lie in the same subspace, bridging the domain gap at the semantic level; the *semantic consistency* loss,  $\mathcal{L}_{sem}$ , ensures that input semantics are preserved after domain translation;  $\mathcal{L}_{gan}$  is a simple generative adversarial (GAN) objective, encouraging the model to generate more realistic samples, and finally,  $\mathcal{L}_{teach}$  is an optional teacher loss that distils prior knowledge from a fixed pretrained teacher embedding, when available. The total loss function is defined as a weighted sum over these five loss terms:

$$\mathcal{L}_{XGAN} = \mathcal{L}_{rec} + \omega_d \mathcal{L}_{dann} + \omega_s \mathcal{L}_{sem} + \omega_g \mathcal{L}_{gan} + \omega_t \mathcal{L}_{teach},$$

where the  $\omega$  hyperparameters control the contributions from each of the individual objectives. An overview of the model is given in Fig. 2, and we discuss each objective in more detail in the rest of this section.

*Reconstruction loss,  $\mathcal{L}_{rec}$ .*  $\mathcal{L}_{rec}$  encourages the model to encode enough information on each domain to perfectly reconstruct the input. More specifically  $\mathcal{L}_{rec} = \mathcal{L}_{rec,1} + \mathcal{L}_{rec,2}$  is the sum of reconstruction losses for each domain.

$$\mathcal{L}_{rec,1} = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{D}_1}} (\|\mathbf{x} - d_1(e_1(\mathbf{x}))\|_2), \text{ likewise for domain } \mathcal{D}_2 \quad (1)$$

*Domain-adversarial loss,  $\mathcal{L}_{dann}$ .*  $\mathcal{L}_{dann}$  is the domain-adversarial loss between  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , as introduced in [3]. It encourages the embeddings learned by  $e_1$  and  $e_2$  to lie in the same subspace. In particular, it guarantees the soundness of the cross-domain transformations  $g_{1 \rightarrow 2}$  and  $g_{2 \rightarrow 1}$ . More formally, this is achieved by training a binary classifier,  $c_{dann}$ , on top of the embedding layer to categorize encoded images from *both* domains as coming from either  $\mathcal{D}_1$  or  $\mathcal{D}_2$  (see Fig. 2B1).  $c_{dann}$  is trained to maximize its classification accuracy while the encoders  $e_1$  and  $e_2$  simultaneously strive to minimize it, i.e., to confuse the domain-adversarial classifier. Denoting model parameters by  $\theta$  and a classification loss function by  $\ell$  (e.g., cross-entropy), we optimize

$$\min_{\theta_{e_1}, \theta_{e_2}} \max_{\theta_{dann}} \mathcal{L}_{dann}, \text{ where} \quad (2)$$

$$\mathcal{L}_{dann} = \mathbb{E}_{p_{\mathcal{D}_1}} \ell(1, c_{dann}(e_1(\mathbf{x}))) + \mathbb{E}_{p_{\mathcal{D}_2}} \ell(2, c_{dann}(e_2(\mathbf{x})))$$

*Semantic consistency loss,  $\mathcal{L}_{sem}$ .* Our key contribution is a semantic consistency feedback loop that acts as self-supervision for the cross-domain translations  $g_{1 \rightarrow 2}$  and  $g_{2 \rightarrow 1}$ . Intuitively, we want the semantics of input  $\mathbf{x} \in \mathcal{D}_1$  to be preserved when translated to the other domain,  $g_{1 \rightarrow 2}(\mathbf{x}) \in \mathcal{D}_2$ , and similarly for the reverse mapping. However, this consistency property is hard to assess at the pixel-level as we do not have paired data and pixel-level metrics are suboptimal for image comparison. Instead, we introduce a feature-level semantic consistency loss, which encourages the network to preserve the learned embedding during domain translation. Formally,  $\mathcal{L}_{sem} = \mathcal{L}_{sem,1 \rightarrow 2} + \mathcal{L}_{sem,2 \rightarrow 1}$ , where

$$\mathcal{L}_{sem,1 \rightarrow 2} = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{D}_1}} \|e_1(\mathbf{x}) - e_2(g_{1 \rightarrow 2}(\mathbf{x}))\|, \text{ likewise for } \mathcal{L}_{sem,2 \rightarrow 1}. \quad (3)$$

$\|\cdot\|$  denotes a distance between vectors.

*GAN objective,  $\mathcal{L}_{gan}$ .* We find that generating realistic image transformations has a crucial positive effect for learning a joint meaningful and semantically consistent embedding as the produced samples are fed back through the encoders when computing the semantic consistency loss: making the transformed distribution  $p(g_{2 \rightarrow 1}(\mathcal{D}_2))$  as close as possible to the original domain  $p(\mathcal{D}_1)$  ensures that the encoder  $e_1$  does not have to cope with an additional domain shift.

Thus, to improve sample quality, we add a generative adversarial loss [5]  $\mathcal{L}_{gan} = \mathcal{L}_{gan,1 \rightarrow 2} + \mathcal{L}_{gan,2 \rightarrow 1}$ , where  $\mathcal{L}_{gan,1 \rightarrow 2}$  is a state-of-the-art GAN objective [5] where the generator  $g_{1 \rightarrow 2}$  is paired against the discriminator  $D_{1 \rightarrow 2}$  (and likewise for  $g_{2 \rightarrow 1}$  and  $D_{2 \rightarrow 1}$ ). In this scheme, a discriminator  $D_{1 \rightarrow 2}$  strives to distinguish generated samples from real ones in  $\mathcal{D}_2$ , while the generator  $g_{1 \rightarrow 2}$  aims to produce samples that confuse the discriminator. The formal objective is

$$\min_{\theta_{g_{1 \rightarrow 2}}} \max_{\theta_{D_{1 \rightarrow 2}}} \mathcal{L}_{gan,1 \rightarrow 2} \quad (4)$$

$$\mathcal{L}_{gan,1 \rightarrow 2} = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{D}_2}} (\log(D_{1 \rightarrow 2}(\mathbf{x}))) + \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{D}_1}} (\log(1 - D_{1 \rightarrow 2}(g_{1 \rightarrow 2}(\mathbf{x}))))$$

Likewise,  $\mathcal{L}_{gan,2 \rightarrow 1}$  is defined for the transformation from  $\mathcal{D}_2$  to  $\mathcal{D}_1$ .

Note that the combination of the  $\mathcal{L}_{gan}$  and  $\mathcal{L}_{sem}$  objectives should subsume the role of the domain-adversarial loss  $\mathcal{L}_{dann}$  in theory. However,  $\mathcal{L}_{dann}$  plays an important role at the beginning of training to bring embeddings across domains closer, as the generated samples are typically poor and not yet representative of the actual input domains  $\mathcal{D}_1$  and  $\mathcal{D}_2$ .

*Teacher loss,  $\mathcal{L}_{teach}$ .* We introduce an optional component to incorporate prior knowledge in the model when available, e.g., in a semi-supervised setting.  $\mathcal{L}_{teach}$  encourages the learned embeddings to lie in a region of the subspace defined by the output representation of a pretrained teacher network,  $T$ . In other words, we distill feature-level knowledge from  $T$  and constrains the embeddings to a more meaningful sub-region, relative to the task on which  $T$  was trained. This can be seen as a form of regularization of the learned embedding. Moreover,  $\mathcal{L}_{teach}$  is asymmetric by definition. It should not be used for both domains simultaneously as each term would potentially push the learned embedding in two different directions. Formally,  $\mathcal{L}_{teach}$  (applied to domain  $\mathcal{D}_1$ ) is defined as

$$\mathcal{L}_{teach} = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{D}_1}} \|T(\mathbf{x}) - e_1(\mathbf{x})\|, \quad (5)$$

where  $\|\cdot\|$  is a distance between vectors.

### 3.1 Architecture and Training Procedure

We use a simple mirrored convolutional architecture for the auto-encoder. It consists of five convolutional blocks for each encoder, the two last ones being shared across domains, and likewise for the decoders (five deconvolutional blocks with the two first ones shared). This encourages the model to learn shared representations at different levels of the architecture rather than only in the middle layer. A more detailed description is given in Table 1. For the teacher network, we use the highest convolutional layer of FaceNet [21], a state-of-the-art face recognition model trained on natural images.

**Table 1** Overview of the XGAN architecture used in practice. The encoder and decoder have the same architecture for both domains, and (//) indicates that the layer is shared across domain

Layer	Size	Layer	Size	Layer	Size
Inputs	64x64x3	Inputs	1x1x1024	Inputs	64x64x3
conv1	32x32x32	(//) deconv1	4x4x512	conv1	32x32x16
conv2	16x16x64	(//) deconv2	8x8x256	conv2	16x16x32
(//) conv3	8x8x128	deconv3	16x16x128	conv3	8x8x32
(//) conv4	4x4x256	deconv4	32x32x64	conv4	4x4x32
(//) FC1	1x1x1024	deconv5	64x64x3	FC1	1x1x1
(//) FC2	1x1x1024				

(a) Encoder

(b) Decoder

(c) Discriminator

The XGAN training objective is to minimize (Eq. 1). In particular, the two adversarial losses ( $\mathcal{L}_{gan}$  and  $\mathcal{L}_{dann}$ ) lead to min-max optimization problems requiring careful optimization. For the GAN loss  $\mathcal{L}_{gan}$ , we use a standard adversarial training scheme [5]. Furthermore, for simplicity we only use one discriminator in practice, namely,  $D_{1 \rightarrow 2}$  which corresponds to the face-to-cartoon path, our target application. We first update the parameters of the generators  $g_{1 \rightarrow 2}$  and  $g_{2 \rightarrow 1}$  in one step. We then keep these fixed and update the parameters for the discriminator  $D_{1 \rightarrow 2}$ . We iterate this alternating process throughout the training. The adversarial training scheme for  $\mathcal{L}_{dann}$  can be implemented in practice by connecting the classifier  $c_{dann}$  and the embedding layer *via* a gradient reversal layer [3]: the feed-forward pass is unaffected, however, the gradient is backpropagated to the encoders with a sign-inversion representing the min-max alternation. We perform this update simultaneously when computing the generator parameters. Finally, we train the model with ADAM optimizer [11] and an initial learning rate of 1e-4.

## 4 The CartoonSet Dataset

Although previous work has tackled the task of transforming frontal faces to a specific cartoon style, there is currently no such dataset publicly available. For this purpose, we introduce a new dataset, CartoonSet,<sup>1</sup> which we release publicly to further aid research on this topic.

Each cartoon face is composed of 16 components including 12 facial attributes (e.g., facial hair, eye shape, etc), and 4 color attributes (such as skin or hair color) which are chosen from a discrete set of RGB values. The number of options per attribute category ranges from 3 to 111, for the largest category, hairstyle. Each

<sup>1</sup>CartoonSet, <https://github.com/google/cartoonset>.



of these components and their variation were drawn by the same artist, resulting in approximately 250 cartoon components artworks and  $10^8$  possible combinations. The artwork components are divided into a fixed set of layers that define a Z-ordering for rendering. For instance, face shape is defined on a layer below eyes and glasses, so that the artworks are rendered in the correct order. For instance, hairstyle needs to be defined on two layers, one behind the face and one in front. There are eight total layers: hair back, face, hair front, eyes, eyebrows, mouth, facial hair, and glasses. The mapping from attribute to artwork is also defined by the artist such that any random selection of attributes produces a visually appealing cartoon without any misaligned artwork, which sometimes involves handling interaction between attributes, e.g., the appearance of “short beard” will change depending on the face shape. For example, the proper way to display a “short beard” changes for different face shapes, which required the artist to create a “short beard” artwork for each face shape. We create the CartoonSet dataset from arbitrary cartoon faces by randomly sampling value for each attribute. We then filter out unusual hair colors (pink, green, etc) or unrealistic attribute combinations, which results in a final dataset of approximately 9,000 cartoons. In particular, the filtering step guarantees that the dataset only contains realistic cartoons, while being completely unrelated to the source dataset.

## 5 Experiments

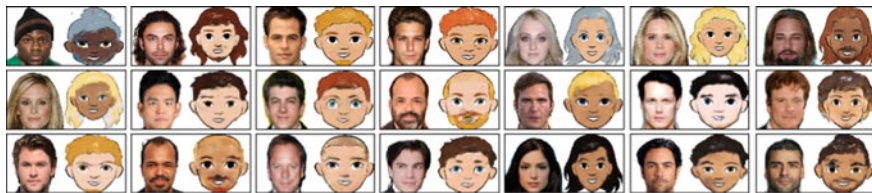
We experimentally evaluate our XGAN model on *semantic style transfer*; more specifically, on the task of converting images of frontal faces (source domain) to images of cartoon avatars (target domain) given an unpaired collection of such samples in each domain. Our source domain is composed of real-world frontal-face images from the VGG-Face dataset [20]. In particular, we use an image collection consisting of 18,054 uncropped celebrity frontal face pictures. As a preprocessing step, we align the faces based on eyes and mouth location and remove the background. The target domain is the CartoonSet dataset introduced in the previous section. Finally, we randomly select and take out 20% of the images from each dataset for testing purposes, and use the remaining 80% for training. For our experiments, we also resize all images to  $64 \times 64$ . As shown in Figs. 3 and 4, the two domains vary significantly in



Fig. 3 Random samples from our cartoon dataset, CartoonSet



Fig. 4 Random centered aligned samples from VGG-Face. We preprocess them with automatic portrait matting to avoid dealing with background noise



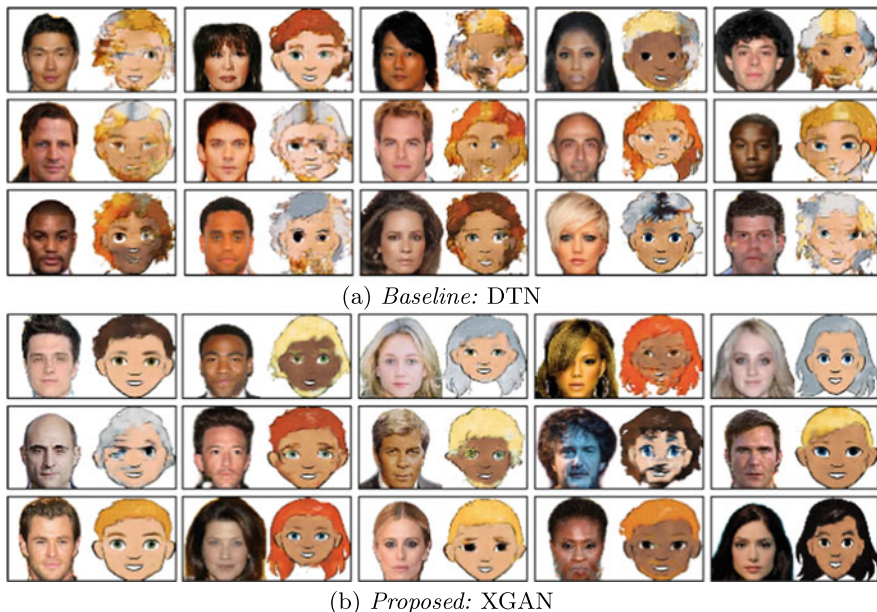
**Fig. 5** Selected samples generated by XGAN on the VGG-Face (left) to CartoonSet (right) task. The figure reads row-wise: for each face-cartoon pair, the target image (cartoon) on the right was generated from the source image (face) on the left

appearance. In particular, cartoon faces are rather simplistic compared to real faces and do not display as much variety (e.g., noses or eyebrows only have a few shape options). Furthermore, we observe a major *content distribution shift* between the two domains due to the way we collected the data: for instance, certain hair color shades (e.g., bright red, gray) are overrepresented in the cartoon domain compared to real faces. Similarly, the cartoon dataset contains many samples with eyeglasses while the source dataset only has a few (Fig. 5).

*Comparison to the DTN baseline.* Our first evaluation is a qualitative comparison between the Domain Transfer Network (DTN) [23] and XGAN on the semantic style transfer problem outlined above. To the best of our knowledge, DTN is the current state of the art for semantic style transfer given unpaired image corpora from two domains with significant visual shift. In particular, DTN was also applied to the task of transferring face pictures to cartoons (bitmojis) in the original chapter.<sup>2</sup> Figure 6 shows the results of both DTN and XGAN applied to random VGG-Face samples from the test set to produce their cartoon counterpart. Evaluation metrics for style transfer are still an active research topic with no good unbiased solution yet. Hence, we choose optimal hyperparameters by manually evaluating the quality of resulting samples, focusing on accurate transfer of semantic attributes, similarity of the resulting sample to the target domain, and crispness of samples.

It is clear from Fig. 6 that DTN fails to capture the transformation function that semantically stylizes frontal faces to cartoons from our target domain. In contrast, XGAN is able to produce sensible cartoons both in terms of the style domain—the resulting cartoons look crisp and respect the specific CartoonSet style—and in terms of semantic similarity to the input samples from VGG-Face. There are some failure cases such as hair or skin color mismatch, which emerge from the weakly supervised nature of the task and the significant content shift between the two domains (e.g., red hair is overrepresented in the target cartoon dataset). In Fig. 5 we report selected XGAN samples that we think best illustrate its semantic consistency abilities, showing that the model learns a meaningful shared representation that preserves common face semantics. Additional random samples are also reported in Fig. 7.

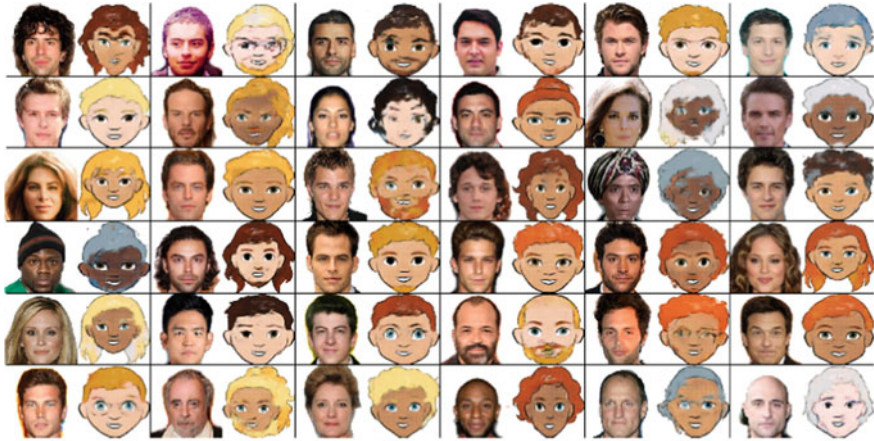
<sup>2</sup>The original DTN code and dataset is not publicly available, hence, we instead report results from our implementation applied to the VGG-Face to CartoonSet setting.



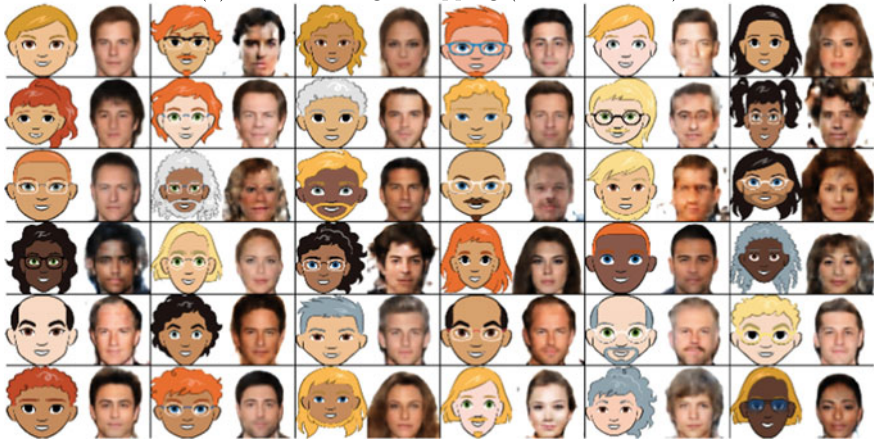
**Fig. 6** A qualitative comparison between DTN and XGAN. In both cases we present random test samples for the face-to-cartoon transformation. The tables are organized row-wise where each face input is mapped to the cartoon face immediately on its right

We believe the failure of DTN is primarily due to its assumption of a fixed joint encoder for both domains. Although the decoder learns to reconstruct inputs from the target domain almost perfectly, the semantics are not well preserved across domains and the decoder yields samples of poor quality for the domain transfer. In fact, FaceNet was originally trained on real faces inputs, hence there is no guarantee that it can produce a meaningful representation for CartoonSet samples. In contrast to our dataset, the target bitmoji domain in [23] is visually closer to real faces, as bitmojis are more realistic and customizable than the cartoon style domain we use here. This might explain the original work performance even with a fixed encoder. Our experiments suggest that using a fixed encoder is too restrictive and does not adapt well to new scenarios. We also train a DTN with a fine-tuned encoder which yields samples of better quality than the original DTN. However, this setup is very sensitive to hyperparameters choice during training and prone to mode collapse.

*Comparison to CycleGAN.* As we have mentioned in the related work section, CycleGAN [27], DiscoGAN [10], and DualGAN [25] form another family of closely related work for image-to-image translation problems. However, differently from DTN and the proposed XGAN, these models only consider a pixel-level cycle-consistency loss and do not use a shared domain embedding. Consequently, they fail to capture high-level shared semantics between significantly different domains.



(a) Source to target mapping (face-to-cartoon)



(b) Target to source mapping (cartoon-to-face)

**Fig. 7** Random samples obtained by applying XGAN on faces and cartoons from the testing set for both cross-domain mappings

To explore this problem, we experiment with CycleGAN<sup>3</sup> on the face-to-cartoon task. We train a CycleGAN with a pix2pix [8] generator as in the original chapter, which is close to the generator we use in XGAN in terms of architecture choices and size (depth and width of the network). As shown in Fig. 8, this approach yields poor results, which is explained by the explicit pixel-level cycle-consistency loss and the fact that the pix2pix architecture contains backward connections (U-net) between the encoder and the decoder; both these features enhance pixel structure similarities which are not desirable for this task.

<sup>3</sup>CycleGAN-tensorflow, <https://github.com/xhujoy/CycleGAN-tensorflow>.

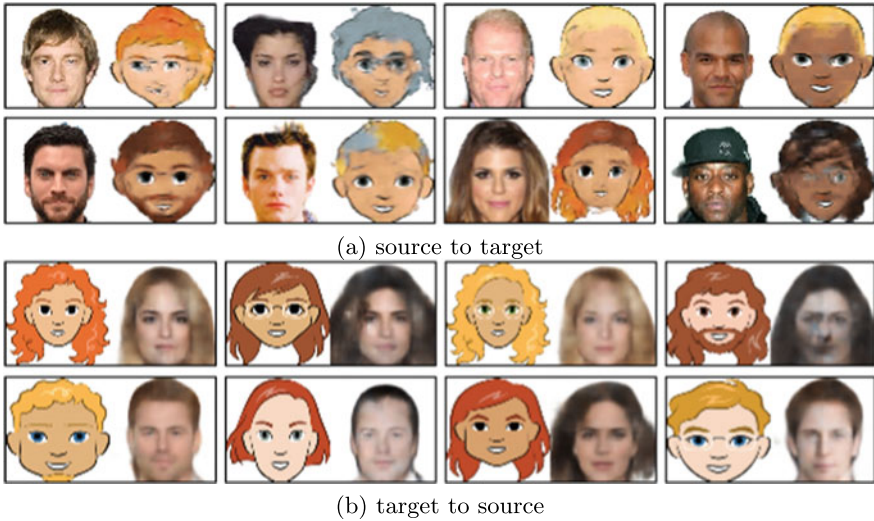


**Fig. 8** The default CycleGAN model is not suitable for transformation between domains with very dissimilar appearances as it enforces pixel-level structural similarities

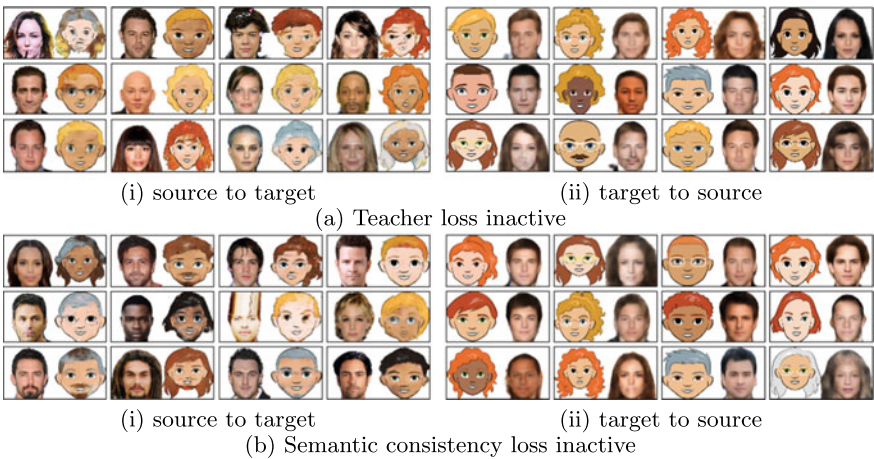
*Ablation study.* We conduct a number of insightful ablation experiments on XGAN. We first consider training only with the reconstruction loss  $\mathcal{L}_{rec}$  and domain-adversarial loss  $\mathcal{L}_{dann}$ . In fact, these form the core domain adaptation component in XGAN and, as we will show, are already able to capture basic semantic knowledge across domains in practice. Second, we experiment with the semantic consistency loss and teacher loss. We show that both have complementary constraining effects on the embedding space which contributes to improving the sample consistency.

We first experiment on XGAN with only the reconstruction and domain-adversarial losses active. These components prompt the model to (i) encode enough information for each decoder to correctly reconstruct images from the corresponding domain and (ii) to ensure that the embedding lies in a common subspace for both domains. In practice in this setting, the model is robust to hyperparameter choice and does not require much tuning to converge to a good regime, i.e., low reconstruction error and around 50% accuracy for the domain-adversarial classifier. As a result of (ii), applying each decoder to the output of the other domain’s encoder yields reasonable cross-domain translations, albeit of low quality (see Fig. 9). Furthermore, we observe that some simple semantics such as skin tone or gender are overall well preserved by the learned embedding due to the shared auto-encoder structure. For comparison, failure modes occur in extreme cases, e.g., when the model capacity is too small, in which case transferred samples are of poor quality, or when the weight  $\omega_d$  is too low. In the latter case, the source and target embeddings are easily distinguishable and the cross-domain translations do not look realistic.

Second, we investigate the benefits of adding semantic consistency in XGAN via the following three components: *sharing high-level layers* in the auto-encoder leads the model to capture common semantics earlier in the architecture. In general, high-level layers in convolutional neural networks are known to encode semantic information. We performed experiments with sharing only the middle layer in the dual auto-encoder. As expected, the resulting embedding does not capture relevant shared domain semantics. Second, we use the *semantic consistency loss* as self-supervision for the learned embedding, ensuring that it is preserved through the cross-domain transformations. It also reinforces the action of the domain-adversarial loss as it constrains embeddings from the two input domains to lie close to each other. Finally, the optional *teacher loss* leads the learned source embedding to lie near the teacher



**Fig. 9** Test results for XGAN with the reconstruction ( $\mathcal{L}_{rec}$ ) and domain-adversarial ( $\mathcal{L}_{dann}$ ) losses active only in the training objective  $\mathcal{L}_{XGAN}$



**Fig. 10** Results of ablating the teacher loss ( $\mathcal{L}_{teach}$ ) (top) and semantic consistency loss ( $\mathcal{L}_{sem}$ ) (bottom) in the XGAN objective  $\mathcal{L}_{XGAN}$

output (in our case, FaceNet’s representation layer), which is meant for real faces. It acts in conjunction with the domain-adversarial loss and semantic consistency loss, whose role is to bring the source and target embedding distributions closer to each other.

In Fig. 10 we report random test samples for both domain translations when ablating the teacher loss and semantic consistency loss, respectively. While it is hard to

draw conclusions from visual inspections, it seems that the teacher network has a positive regularization effect on the learned embedding by guiding it to a more realistic region: training the model without the teacher loss (Fig. 10a) yields more distorted samples, especially when the input is an outlier, e.g., person wearing a hat, or cartoons with unusual hairstyle. Conversely, when the semantic consistency is inactive (Fig. 10b), the generated samples overall display less variety. In particular, rare attributes (e.g., unusual hairstyle) are not as well preserved as when the semantic consistency term is present.

*Discussions and Limitations.* Our initial aim was to tackle the *semantic style transfer* problem in a fully unsupervised framework by combining techniques from domain adaptation and image-to-image translation: We first observe that using a simple setup where a partially shared dual auto-encoder is trained with reconstruction and domain-adversarial losses already suffice to produce an embedding that captures basic semantics rather well (for instance, skin tone). However, the generated samples are of poor quality and fine-grained attributes such as facial hair are not well captured. These two problems are greatly diminished after adding the GAN loss and the proposed semantic consistency loss, respectively. Failure cases still exist, especially on non-representative input samples (e.g., a person wearing a hat) which are mapped to unrealistic cartoons. Adding the teacher loss mitigates this problem by regularizing the learned embedding, however, it requires additional supervision and makes the model dependent on the specific representation provided by the teacher network.

Future work will focus on evaluating XGAN on different domain transfer tasks. In particular, though we introduced XGAN for semantic style transfer, we think the model goes beyond this scope and can be applied to classical domain adaptation problems, where quantitative evaluation becomes possible: while the pixel-level transformations are not necessary for learning the shared embedding, they are beneficial for learning a meaningful representation across visual domains, when combined with the self-supervised semantic consistency loop.

## 6 Conclusions

In this work, we introduced XGAN, a model for unsupervised domain translation applied to the task of semantically consistent style transfer. In particular, we argue that similar to the domain adaptation task, learning image-to-image translation between two structurally different domains requires learning a high-level joint semantic representation while discarding local pixel-level dependencies. Additionally, we proposed a semantic consistency loss acting on both domain translations as a form of self-supervision.

We reported promising experimental results on the task of face-to-cartoon that outperform the current baseline. We also showed that additional weak supervision, such as a pretrained feature representation, can easily be added to the model in the form of teacher knowledge. It acts as a good regularizer for the learned embeddings and generated samples. This is particularly useful for natural image datasets, for which off-the-shelf pretrained models are abundant.

## References

1. Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D (2017) Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR
2. Bousmalis K, Trigeorgis G, Silberman N, Krishnan D, Erhan D (2016) Domain separation networks. In: NIPS
3. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. *J Mach Learn Res*
4. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: CVPR
5. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
6. Hoffman J, Tzeng E, Park T, Zhu J, Isola P, Saenko K, Efros AA, Darrell T (2017) CyCADA: cycle-consistent adversarial domain adaptation. *CoRR* [arXiv:abs/1711.03213](https://arxiv.org/abs/1711.03213)
7. Huang X, Liu MY, Belongie S, Kautz J (2018) Multimodal unsupervised image-to-image translation. [arXiv:1804.04732](https://arxiv.org/abs/1804.04732)
8. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: CVPR
9. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: ECCV
10. Kim T, Cha M, Kim H, Lee JK, Kim J (2017) Learning to discover cross-domain relations with generative adversarial networks. In: ICML
11. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: ICLR
12. Larsen ABL, Sønderby SK, Winther O (2016) Autoencoding beyond pixels using a learned similarity metric. In: ICML
13. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR
14. Li C, Wand, M (2016) Combining Markov random fields and convolutional neural networks for image synthesis. In: CVPR
15. Liao J, Yao Y, Yuan L, Hua G, Kang SB (2017) Visual attribute transfer through deep image analogy. *ACM Trans Graph*
16. Liu M, Breuel T, Kautz J (2017) Unsupervised image-to-image translation networks. In: NIPS
17. Liu MY, Tuzel O (2016) Coupled generative adversarial networks. In: NIPS
18. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR
19. Ma L, Jia X, Georgoulis S, Tuytelaars T, Van Gool L (2018) Exemplar guided unsupervised image-to-image translation. [arXiv:1805.11145](https://arxiv.org/abs/1805.11145)
20. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. In: BMVC
21. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: a unified embedding for face recognition and clustering. In: CVPR
22. Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R (2017) Learning from simulated and unsupervised images through adversarial training. In: CVPR



23. Taigman Y, Polyak A, Wolf L (2017) Unsupervised cross-domain image generation. In: ICLR
24. Wolf L, Taigman Y, Polyak A (2017) Unsupervised creation of parameterized avatars. In: ICCV
25. Yi Z, Zhang H, Tan P, Gong M (2017) DualGan: unsupervised dual learning for image-to-image translation. In: ICCV
26. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. In: ECCV
27. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV