DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks

Shuang Ma
The State Universtiy of New York at Buffalo
shuangma@buffalo.edu

Jianlong Fu
Microsoft Research
jianf@microsoft.com

Chang Wen Chen
The State Universtiy of New York at Buffalo
chencw@buffalo.edu

Tao Mei
Microsoft Research
tmei@microsoft.com

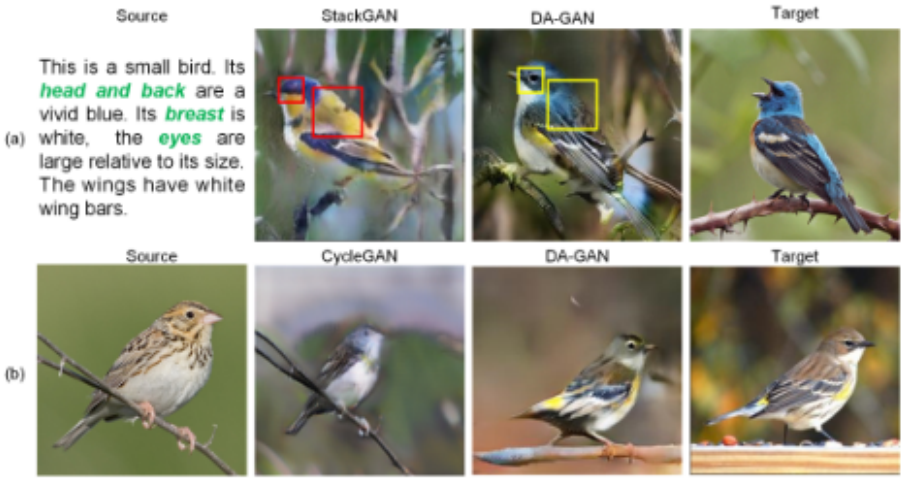

Figure 1: (a) text-to-image generation. (b) object configuration. We can observe that the absence of instance-level correspondences results in both semantic artifacts (labeled by red boxes) exist in StackGAN and geometry artifacts exist in CycleGAN. Our approach successfully produces the correct correspondences (labeled by yellow boxes) because of the proposed instance-level translating. Details can be found in Sec. 1

- We decompose the task to instance-level image translation such that the constraints could be exploited on both instance-level and set-level by adopting the proposed compound loss.
- To the best of our knowledge, we are the first that integrate the attention mechanism into Generative Adversarial Network.
- We introduce a novel framework DA-GAN, which produces visually appealing results and is applicable in a large variety of tasks.

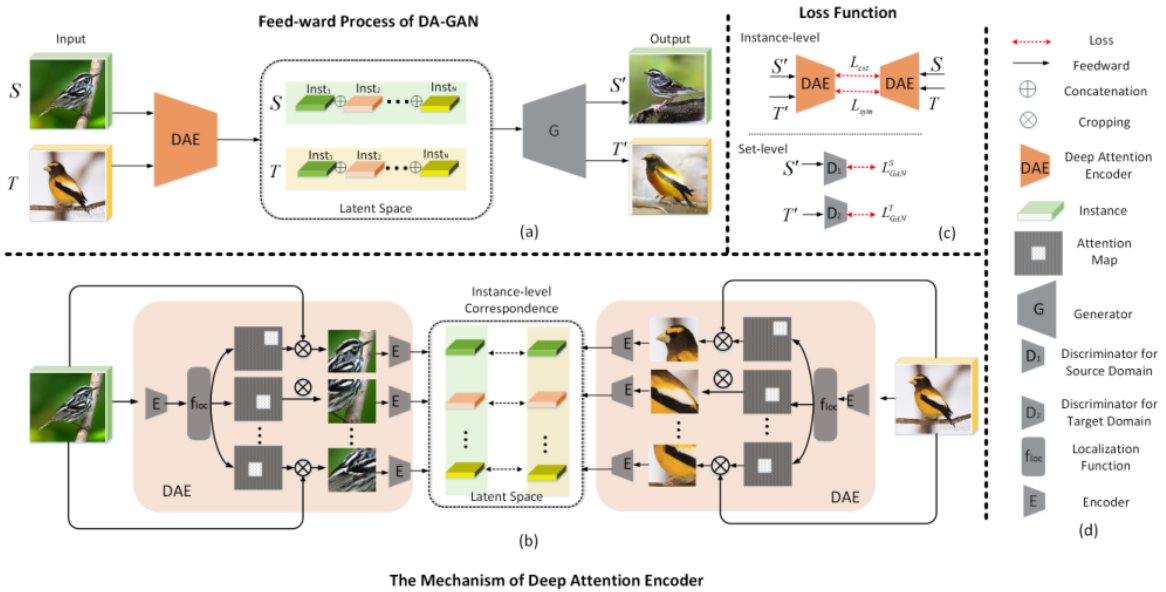

Figure 2: A pose morphing example for illustration the pipeline of DA-GAN. Given two images of birds from source domain S and target domain T, the goal of pose morphing is to translate the pose of source bird s into the pose of target one t, while still remain the identity of s. The feed-ward process is shown in (a), where two input images are fed into DAE which projects them into a latent space (labeled by dashed box). Then G takes these highly-structured representations ($DAE(s)$ and $DAE(t)$) from the latent space to generated the translated samples, i.e.$s' = G(DAE(s))$, $t' = G(DAE(t))$. The details of the proposed DAE (labeled by orange block) is shown in (b). Given an image X, a localization function f_{loc} will first predict N attention regions' coordinates from the feature map of X, (i.e. $E(X)$, where E is an encoder, which can be utilized in any form). Then N attention masks are generated and activated on X to produce N attention regions $\{R_i\}_{i=1}^{N}$. Finally, each region's feature consists the instance-level representations $\{Inst_i\}_{i=1}^{N}$. By operating the same way on both S and T, the instance-level correspondences can consequently be found in the latent space. We exploit constraints on both instance-level and set-level for optimization, it is illustrated in (c). All of the notations are listed in (d). [Best viewed in color.]

3.1. Deep Attention Encoder

To project samples into the latent space, we integrate attention mechanism to jointly learn an Deep Attention Encoder DAE. Given a feature map $E(X)$ of an input image X (where E is an encoder that could be utilized in any form), we first adopt a localization function $f_{loc}(\cdot)$ to predict a set of attention regions' location, which is given by:

$$f_{loc}(E(X)) = [x_i, y_i]_{i=1}^{N'}, \tag{1}$$

where $[x_i, y_i]$ denotes a region's center coordinates, N' denotes the number of regions predicted. Once the the location of an attended region is hypothesized, we generate an attention mask $\mathcal{M}_i$. Specifically, we denote w and h as half of the width and half of the height of X. Then we can adopt

the parameterizations of attend region by:

$$\begin{aligned} x_i^{left} &= x_i - w, \quad x_i^{right} = x_i + w, \\ y_i^{top} &= y_i - h, \quad y_i^{bottom} = y_i + h. \end{aligned} \tag{2}$$

The cropping operation can therefore be achieved by an element-wise multiplication applied on X, i.e. $R_i = X^{\circ}\mathcal{M}_i$, which produces the attended regions $\{R_i\}_{i=1}^{N'}$. Then instance-level representations of X in the latent space are defined by:

$$\{E(R_i)\}_{i=1}^{N'} = \{Inst\}_{i=1}^{N'}, \tag{3}$$

To allow backpropagation, here we adopt the attention mask as:

$$\begin{aligned} \mathcal{M}_i = [\sigma(x - x_i^{left}) - \sigma(x - x_i^{right})] \cdot \\ [\sigma(y - y_i^{top}) - \sigma(y - y_i^{bottom})], \end{aligned} \tag{4}$$

where $\sigma(\cdot) = 1/(1+\exp^{-kx})$ is a sigmoid function. In theory, when k is large enough, $\sigma(\cdot)$ is approximated as a step function and $\mathcal{M}_i$ will become a two dimensional rectangular function, then the derivation could be approximated. For learning these attention regions, we add a geometric regularization $\mathbb{E}_{X\sim P_{data}(X)}[d(Y, DAE(X))]$. Y is the label of image X, and d is some similarity metrics in the data space, In practice, there are many options for the distance measure d. For instance, a VGG classifier.

3.2. Instance-Level Image Translation

As the DAE projects s and t into a shared latent space, we can constrain them to be matched with each other in this latent space. Therefore, we adopt a consistency loss on the samples from source domain $\{s_i\}_{i=1}^N$ and the according translated samples $\{s'_i\}_{i=1}^N$:

$$\mathcal{L}_{cst} = \mathbb{E}_{s\sim P_{data}(s)}\, d(DAE(s),\ DAE(F(s)), \quad (5)$$

On the other hand, we also consider the samples from the target domain to further enforce the mapping to be deterministic. In theory, if a mapping is bijective (one-to-one corresponding), the operation from a set to itself form a symmetric group. The mapping can then be considered as a permutation operation on itself. We therefore exploit a symmetry loss to enforce F can map samples from T to themselves, i.e. $t_i \approx F(t_i)$. The loss function is defined as:

$$\mathcal{L}_{sym} = \mathbb{E}_{t\sim P_{data}(t)}\, d(DAE(t),\ DAE(F(t)), \quad (6)$$

this can also be considered as an auto-encoder type of loss applied on samples from T, where d is a distance measure. In theory, there are many options for d. For instance, the L^n distance, or the distance of learned features by the discriminator or by other networks, such as a VGG classifier.

3.3. Set-Level Image Translation

It is straight-forward to use a discriminator D_1 to distinguish the translated samples $\{s'_i\}_{i=1}^{N}$ from the real samples in the target domain $\{t\}_{i=1}^{M}$, and generator is forced to translate samples that is indistinguishable from real samples in target domain, which is given by:

$$\mathcal{L}^{s}_{GAN} = \mathbb{E}_{t\sim P_{data}(t)}[\log D_1(t)] + \mathbb{E}_{t\sim P_{data}(s)}[\log(1 - D_1(F(s)))]. \quad (7)$$

While there still exists another issue - mode collapse. In theory, large modes usually have a much higher chance of attracting the gradient of discriminator, and the generator is not penalized for missing modes. In practice, all input samples map to the same output, and the optimization fails to make progress. This issue asks for adding penalty on generator for missing modes.

As we mentioned before, $DAE \circ G$ can be considered as an auto-encoder for $\{t_i\}_{i=1}^{M}$. Then for every modes in T, $F(t)$ is expected to generate very closely located modes. We therefore add another discriminator D_2 for samples from the target domain to enforce the reconstructed t' is indistinguishable from t. An additional optimization objective for the generator is hence added $\mathbb{E}_{t\sim p_{data}(t)}[\log D_2(F(t))]$. The objective function is given by:

$$\mathcal{L}^{t}_{GAN} = \mathbb{E}_{t\sim P_{data}(t)}[\log D_2(t)] + \mathbb{E}_{t\sim P_{data}(t)}[\log(1 - D_2(F(t)))]. \quad (8)$$

This multi-adversarial training procedure is critical for penalizing the missing modes, it encourage $F(t)$ to move towards a nearby mode of the data generating distribution. In this way, we can achieve fair probability mass distribution across different modes.