Towards Unsupervised Deformable-Instances Image-to-Image Translation

Sitong Su , *[*]*Jingkuan Song** , **Lianli Gao** and **Junchen Zhu**
Center for Future Media, University of Electronic Science and Technology of China
sitongsu9796@gmail.com, jingkuan.song@gmail.com, lianli.gao@uestc.edu.cn,
junchen.zhu@hotmail.com

Abstract

Replacing objects in images is a practical functionality of Photoshop, e.g., clothes changing. This task is defined as Unsupervised Deformable-Instances Image-to-Image Translation (UDIT), which maps multiple foreground instances of a source domain to a target domain, involving significant changes in shape. In this paper, we propose an effective pipeline named Mask-Guided Deformable-instances GAN (MGD-GAN) which first generates target masks in batch and then utilizes them to synthesize corresponding instances on the background image, with all instances efficiently translated and background well preserved. To promote the quality of synthesized images and stabilize the training, we design an elegant training procedure which transforms the unsupervised mask-to-instance process into a supervised way by creating paired examples. To objectively evaluate the performance of UDIT task, we design new evaluation metrics which are based on the object detection. Extensive experiments on four datasets demonstrate the significant advantages of our MGD-GAN over existing methods both quantitatively and qualitatively. Furthermore, our training time consumption is hugely reduced compared to the state-of-the-art. The code could be available at https://github.com/sitongsu/MGD_GAN

1 Introduction

Image-to-Image (I2I) translation aims to learn the mapping between the source and target domain, and begins to emerge as the proposal of Generative Adversarial Networks [Goodfellow *et al.*, 2014]. Since then, increasing attention has been paid to this task because several visual tasks could be transformed into I2I translation such as: style transfer [Liu *et al.*, 2017], super-resolution [Ledig *et al.*, 2017], label-to-image [Park *et al.*, 2019][Gao *et al.*, 2020] and image-inpainting [Yi *et al.*, 2020]. Moreover, great progress has been made in recent years. For example, CycleGAN [Zhu *et al.*, 2017] proposes to exert cycle consistency on the generators. Furthermore, UNIT [Liu *et al.*, 2017] extends the Coupled GAN [Liu and Tuzel, 2016] based on the assumption of a shared latent space. To meet the demand of generating diverse images, MUNIT [Huang *et al.*, 2018], DRIT [Lee *et al.*, 2018], etc. are introduced by recombining the disentangled image representation. The methods above only focus on transferring styles on the whole image without considering characteristics of instances.

Under such condition, Instance-level Image-to-Image Translation is proposed to focus on the specific foreground instances. Generally, it can be classified into two categories: translating both of the background stuff and foreground instances; only translating specific foreground instances while preserving the original background. For the former one, INIT [Shen *et al.*, 2019] firstly raises the idea of translating foreground instances and background areas independently with different styles. Nevertheless, at test time, INIT discards instance information which is contrary with its initial target. To make up for the defect, DUNIT [Bhattacharjee *et al.*, 2020] proposes a unified framework where instances could also be leveraged at test time. As for the latter category, previous methods like AGGAN [Mejjati *et al.*, 2018] and Attention-GAN [Chen *et al.*, 2018] generate attention maps of instances to distinguish the foreground and background.

So far, Instance-level Image-to-Image Translation methods like DUNIT [Bhattacharjee *et al.*, 2020] or AGGAN [Mejjati *et al.*, 2018] are only capable of transferring low-level features like styles. However, in applications like clothes change game, if pants-to-skirt change is required, only transferring the color will be unsatisfactory. To meet the demand, the task of *Unsupervised Deformable-Instances Image-to-Image Translation* (UDIT) is proposed. The task aims to translate foreground instances of a source domain into a target domain, with significant shape deformation in foreground instances and preservation in background. Contrasting-GAN [Liang *et al.*,] firstly achieves the task by cropping and translating instances. However, it could only deal with few objects and the generated images look unnatural. Thus, multiple independent instance masks are incorporated in InstaGAN [Mo *et al.*, 2019]. To guide the instance translation, single mask feature and aggregated mask features are concatenated with image features sequentially.

Yet, as the state-of-the-art in UDIT, there exists several is-

[*]corresponding author

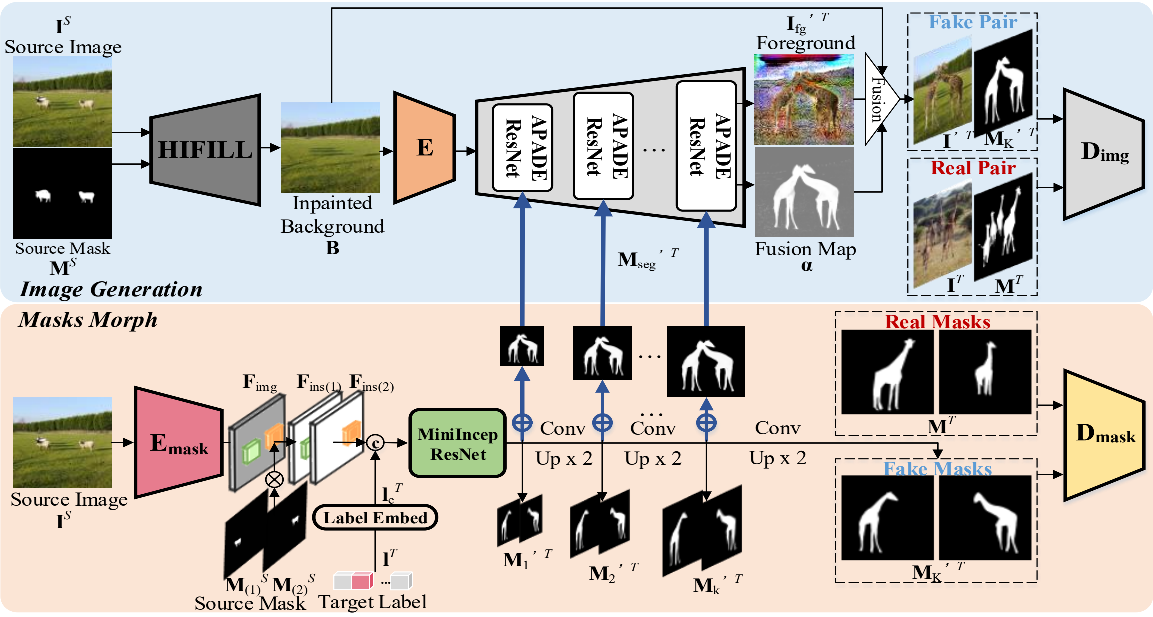

Figure 1: The overall architecture of our MGD-GAN model. The light orange rectangular below represents the Masks Morph part which generates target instance masks efficiently. The light blue rectangular above refers to the Image Generation part which synthesizes vivid instances according to the generated target instance masks while yielding a natural full image.

sues in InstaGAN [Mo *et al.*, 2019]. Firstly, lots of instances fail to translate even if the shapes of generated masks are correct. The simple concatenation of mask and image features leads to the incomplete utilization of shape information. Moreover, sequential training will cause severe time consumption with the increase of instance amount per image. Another defect is that the generated images are unconvincing since the original visual information is partially retained.

To tackle the above issues, we introduce our MGD-GAN to achieve efficient yet accurate multi-instances image-to-image translation with shape deformation. Unlike existing models generating masks and images of the target domain simultaneously, our method decomposes this challenging task into two sequential relatively simpler tasks. The target masks are firstly translated in batch and used to guide the image generation. Thus, the image generation task can fully utilize the shape information, largely relieving the failure cases of inconsistency between generated images and masks. Compared with the sequential training scheme introduced in InstaGAN [Mo *et al.*, 2019], synthesizing the target masks in batch can reduce time-consumption. Besides, we compact all the masks into one map to guide the generation process, thus allowing for multiple instances translation simultaneously without increasing time consumption. We also propose an elegant training scheme which transforms the unsupervised mask-to-instance process into a supervised one by creating paired examples. The designed training scheme not only promotes the generated instance quality, but also contributes to the background quality, since we remove the original instances from the source image and use the inpainted one to be the input of generator.

The major contributions of our work can be summarized as the following manifolds:

1) We propose an effective pipeline for Unsupervised Deformable-instances Image-to-Image Translation (UDIT). The target masks are firstly synthesized to guide the instance generation, thus allowing full utilization of the mask information and avoiding the inconsistency between the generated masks and images.

2) To promote the quality of synthesized images and stabilize the training, we design an elegant training procedure which transforms the unsupervised mask-to-instance process into a supervised way by creating paired examples.

3) We first propose three objective evaluation metrics for UDIT. Extensive experiments are conducted on four datasets constructed from MS COCO [Lin *et al.*, 2014] and Multi-Human Parsing [Zhao *et al.*, 2018]. Quantitative and qualitative results prove that our method surpasses others by a large margin.

2 Method

The overview of our method is illustrated in Fig. 1. Generally, it consists of two major parts: 1) *Masks Morph*, to synthesize target masks; and 2) *Image Generation*, to generate target instances under the guidance of synthesized masks, and render the final image.

2.1 Masks Morph

As depicted in the light orange rectangular in Fig. 1, the source image $\mathbf{I}^{\mathcal{S}}$, source instance masks $\mathbf{M}^{\mathcal{S}}$ and target domain label $\mathbf{l}^{\mathcal{T}}$, which is represented by a one-hot label, are fed into our mask generator G_{mask}. Consequently, we obtain the generated target masks $\mathbf{M}^{'\mathcal{T}}$. Note that the notation $'$ indicates that the image or mask is synthesized. The generation process can be described as follows:

$$\mathbf{M}^{'\mathcal{T}} = G_{mask}(\mathbf{I}^{\mathcal{S}}, \mathbf{M}^{\mathcal{S}}, \mathbf{l}^{\mathcal{T}}). \tag{1}$$

Feature extraction. The mask translation process is supposed to acquire the size and location information of instances, which is usually implemented by encoding functions. Instead of encoding each instance mask sequentially, we encode the whole source image $\mathbf{I}^{\mathcal{S}}$ by an encoder E_{mask} to obtain the encoded image feature $\mathbf{F}_{img}$. Then, each instance feature $\mathbf{F}_{ins(i)}$ is extracted by multiplying its corresponding resized instance mask with $\mathbf{F}_{img}$. By this means, time consumption significantly decreases, since the repeated mask encoding processes are replaced by a single image encoding. The operations above can be described as:

$$\mathbf{F}_{img} = E_{mask}(\mathbf{I}^{\mathcal{S}}), \;\; \mathbf{F}_{ins(i)} = \mathbf{M}^{\mathcal{S}}_{(i)} \circ \mathbf{F}_{img} \;\; i = 1, ..., N_{ins}, \tag{2}$$

where $\circ$ indicates pixel-wise multiplication. $\mathbf{M}^{\mathcal{S}}_{(i)}$ is the resized mask of i-th instance and N_{ins} means the amount of instances in $\mathbf{I}^{\mathcal{S}}$. By concatenating all the instance features in the first channel, we obtain all the foreground instance features $\mathbf{F}_{ins}$ in $\mathbf{I}\mathcal{S}$.

Masks generation. Given the instance features $\mathbf{F}_{ins}$, we aim to translate them to corresponding target masks. To this end, we firstly inject target domain information into generation by concatenating $\mathbf{F}_{ins}$ with embedded target label feature $\mathbf{l}^{\mathcal{T}}_{e}$. Then, labeled instances features are passed through several cascaded MiniIncep ResNets to be fully detected and fused. Our MiniIncep ResNet, like ResNet [He *et al.*, 2016], consists of a forward and a shortcut branch. In the forward branch, several convolutions of different kernel sizes are arranged in parallel to better capture instance information of different sizes. Further, we propose a coarse-to-fine generation scheme to generate multi-scale masks for fine-grained generation. In details, the features are upsampled level by level. In each level k, generated target masks $\mathbf{M}^{'\mathcal{T}}_{k}$ are outputted, and K denotes the total amount of levels.

Four novel loss functions are proposed to assure the mask generation.

Multi-scale mask adversarial loss. To guide the mask generation, we design a discriminator D_{mask} adapted from PatchGAN [Isola *et al.*, 2017]. Our generated masks $\mathbf{M}^{'\mathcal{T}}_{K}$ are taken as fake inputs and masks $\mathbf{M}^{\mathcal{T}}$ sampled from target domain are real ones. Then we draw the outputs from the last two layers of D_{mask} for multi-scale discrimination. Moreover, the masks are fed into D_{mask} independently so that the overlapped information will not be neglected. The adversarial loss function is designed in hinge version [Lim and Ye, 2017]. The adversarial loss for mask generator is defined as:

$$\mathcal{L}^{G}_{mask} = \sum_{i} \mathbb{E}[l^{R}_{adv}(D^{i}_{mask}(\mathbf{M}^{'\mathcal{T}}_{K})], \tag{3}$$

where D^{i}_{mask} is the output of i-th layer in D_{mask}. Similarly, we calculate the adversarial loss for mask discriminator as:

$$\begin{aligned}\mathcal{L}^{D}_{mask} = \frac{1}{2} \sum_{i} (&\mathbb{E}[l^{F}_{adv}(D^{i}_{mask}(\mathbf{M}^{'\mathcal{T}}_{K})] \\ &+ \mathbb{E}[l^{R}_{adv}(D^{i}_{mask}(\mathbf{M}^{\mathcal{T}})]).\end{aligned} \tag{4}$$

Mask pseudo-cycle loss. We propose *Mask Pseudo-Cycle Loss* $\mathcal{L}_{pc}$ to exert extra supervision since we cut off cycle architecture. Specifically, we inject the source label $\mathbf{l}^{\mathcal{S}}$ into G_{mask} and hope to generate unchanged masks. The loss function is:

$$\mathcal{L}_{pc} = \left\| \mathbf{M}^{\mathcal{S}} - G_{mask}(\mathbf{I}^{\mathcal{S}}, \mathbf{M}^{\mathcal{S}}, \mathbf{l}^{\mathcal{S}}) \right\|_1 . \tag{5}$$

Mask consistency loss. Making sure that generated masks of different sizes are consistent may stabilize the training. Thus, we define *Mask Consistency Loss* $\mathcal{L}_{const}$ as:

$$\mathcal{L}_{const} = \sum_{k=2}^{K} \left\| \mathbf{M}^{'\mathcal{T}}_{1} - d(\mathbf{M}^{'\mathcal{T}}_{k}) \right\|_1 , \tag{6}$$

where $\mathbf{M}^{'\mathcal{T}}_{1}$ are the masks of the smallest size, and $d(\cdot)$ means the downsample function for resizing.

Mask regularization loss. According to the experiments, the generated masks tend to be fragmented. To cope with this, we design *Mask Completeness Loss* $\mathcal{L}_{com}$ to force the aggregation of fragments. The generated masks are downsampled and then upsamlped to the original size. Then, we establish consistent loss between the original one and the operated one. To prevent the generated masks from oversize, *Mask Penalty Loss* $\mathcal{L}_{penalty}$ is proposed. The two functions are summed up as *Mask Regularization Loss*, which can be calculated as:

$$\begin{aligned}&\mathcal{L}_{penalty} = \sum_{H,W} \mathbf{M}^{'\mathcal{T}}_{K}, \;\; \mathcal{L}_{com} = \left\| \mathbf{M}^{'\mathcal{T}}_{K} - u(d(\mathbf{M}^{'\mathcal{T}}_{K})) \right\|_1 \\ &\quad \mathcal{L}_{reg} = \lambda_{com}\mathcal{L}_{com} + \lambda_{penalty}\mathcal{L}_{penalty},\end{aligned} \tag{7}$$

where $d(\cdot)$ and $u(\cdot)$ respectively denote the downsample and upsample functions. λ_{com} and $\lambda_{penalty}$ represent the weights of $\mathcal{L}_{com}$ and $\mathcal{L}_{penalty}$, respectively.

With all the aforementioned losses, the loss functions for *Masks Morph* could be summarized as follows:

$$\begin{aligned}\mathcal{L}_{mask} = &\mathcal{L}^{G}_{mask} + \mathcal{L}^{D}_{mask} + \lambda_{pc}\mathcal{L}_{pc} \\ &+ \lambda_{const}\mathcal{L}_{const} + \lambda_{reg}\mathcal{L}_{reg},\end{aligned} \tag{8}$$

where λ_{pc}, λ_{const}, λ_{reg} indicate the weights of $\mathcal{L}_{pc}$, $\mathcal{L}_{const}$ and $\mathcal{L}_{reg}$, respectively.

2.2 Image Generation with Designed Supervision

In this subsection, we introduce the architecture of our image generator G_{img}, and its training scheme.

Image generation. To fulfill the image generation task above, based on SPADE [Park *et al.*, 2019] which is a mature segmentation-to-image framework, we propose our novel *Adapted-SPADE Generator* (APADE) as G_{img}. As shown in Fig. 1, aside from the original generator part SPADE owns, we add an extra encoder E to help extract background feature.

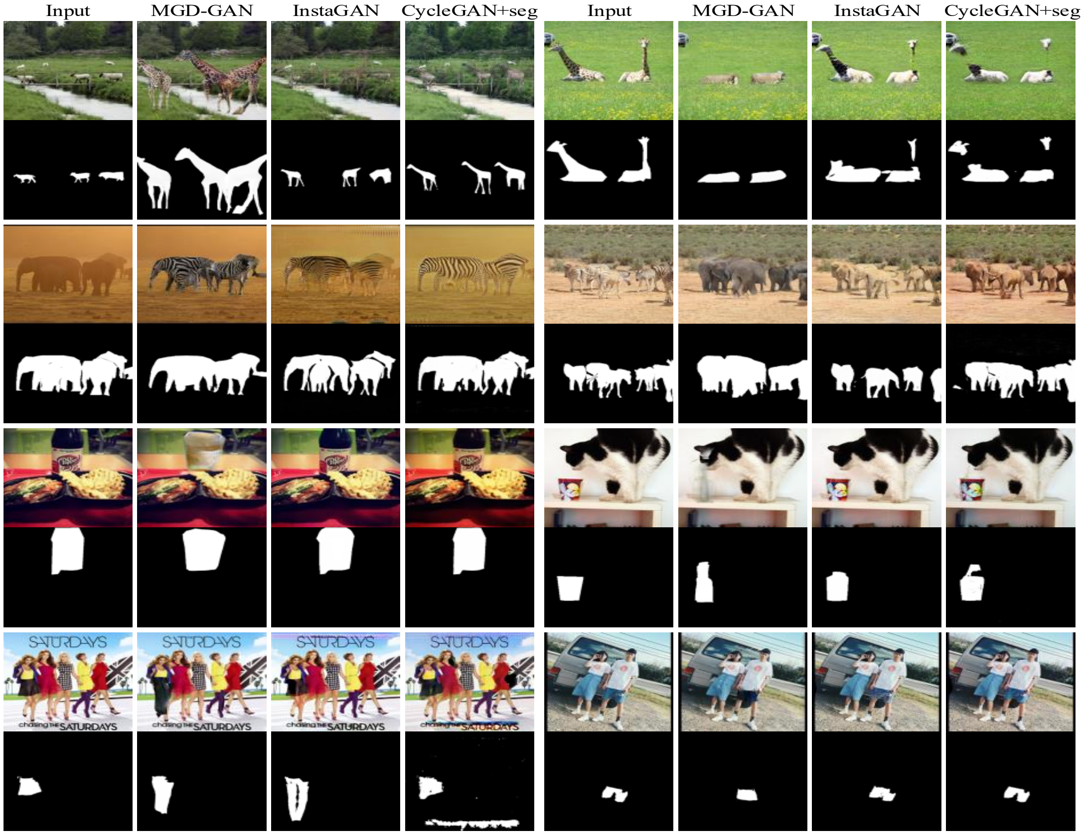

Figure 2: Comparison on *sheep&giraffe*, *elephant&zebra*, *bottle&cup* and *pants&skirt* datasets. Translation is bi-directional.(e.g., The first row shows results of *sheep2giraffe* and *giraffe2sheep*) Our MGD-GAN synthesizes better masks and images than state-of-the-arts.

Additionally, we design a novel input segmentation part $\mathbf{M}'^{\mathcal{T}}_{seg}$ to make the SPADE adaptable with our task. $\mathbf{M}'^{\mathcal{T}}_{seg}$ is composed of summed masks, label information, and aggregated edges for every mask. Furthermore, in order to compensate for the defected generated region in the inpainting process, we also incorporate the background masks into $\mathbf{M}'^{\mathcal{T}}_{seg}$. At inference time, under the guidance of $\mathbf{M}'^{\mathcal{T}}_{seg}$, our G_{img} takes the background image $\mathbf{B}^{\mathcal{S}}$ of the source image as input, and produces the target foreground image $\mathbf{I}'^{\mathcal{T}}_{fg}$ as well as a fusion map $\boldsymbol{\alpha}'^{\mathcal{T}}$. Since directly pasting foregrounds to backgrounds would cause sharp and unnatural margins, we use the fusion map to blend $\mathbf{I}'^{\mathcal{T}}_{fg}$ with $\mathbf{B}^{\mathcal{S}}$ in a natural way, and obtain the final synthesized image $\mathbf{I}'^{\mathcal{T}}$. The generation and blending above can be expressed as:

$$\mathbf{I}'^{\mathcal{T}}_{fg}, \boldsymbol{\alpha}'^{\mathcal{T}} = G_{img}(\mathbf{B}^{\mathcal{S}}, \mathbf{M}'^{\mathcal{T}}_{seg}), \mathbf{I}'^{\mathcal{T}} = \mathbf{I}'^{\mathcal{T}}_{fg} \cdot \boldsymbol{\alpha}'^{\mathcal{T}} + \mathbf{B}^{\mathcal{S}} \cdot (1-\boldsymbol{\alpha}'^{\mathcal{T}}). \tag{9}$$

Designed supervision. Even with the guidance of generated segmentation $\mathbf{M}'^{\mathcal{T}}_{seg}$, training G_{img} is still challenging seeing that there is no ground truth image corresponding to $\mathbf{M}'^{\mathcal{T}}_{seg}$. The direct way to fix the problem is to "create" pairwise training samples for G_{img}. In the created pair, the input of G_{img} is the background $\mathbf{B}^{\mathcal{T}}$ of the target image $\mathbf{I}^{\mathcal{T}}$. While the ground truth is $\mathbf{I}^{\mathcal{T}}$ itself. As depicted in Fig. 1, we obtain the background by adopting the pretrained image inpainting network called HiFill [Yi *et al.*, 2020]. Given the inpainted background $\mathbf{B}^{\mathcal{T}}$, we aim to restore the removed foreground instances according to the instance masks $\mathbf{M}^{\mathcal{T}}$ of $\mathbf{I}^{\mathcal{T}}$. The restoration is trained in a supervised manner, since the ground truth $\mathbf{I}^{\mathcal{T}}$ and $\mathbf{M}^{\mathcal{T}}$ are provided. In this way, we can establish the *Designed Supervision* following the Eq. 9. In the training phase, the inputs of G_{img} in Eq. 9 are $\mathbf{B}^{\mathcal{T}}$ and $\mathbf{M}^{\mathcal{T}}_{seg}$ made from $\mathbf{M}^{\mathcal{T}}$. After the supervised training, G_{img} will be able to synthesize instances on a background image according to given masks. We set several losses for the supervised training process. First, to assure that the fusion map $\boldsymbol{\alpha}'^{\mathcal{T}}$ mostly indicates the foreground part, we use a binary cross entropy loss:

$$\mathcal{L}_{fmap} = -\mathbf{M}^{\mathcal{T}} \cdot \log \boldsymbol{\alpha}'^{\mathcal{T}} - (1-\mathbf{M}^{\mathcal{T}}) \cdot \log(1-\boldsymbol{\alpha}'^{\mathcal{T}}). \tag{10}$$

Following SPADE [Park *et al.*, 2019], we adopt the multi-scale discriminator D_{img}. The adversarial losses for generator and discriminator are defined as $\mathcal{L}^{G}_{img}$ and $\mathcal{L}^{D}_{img}$. Note that VGG similarity loss $\mathcal{L}_{vgg}$ and feature matching loss $\mathcal{L}_{feat}$ in SPADE [Park *et al.*, 2019] are also adopted to promote the performance. Consequently, the overall loss for image gener-

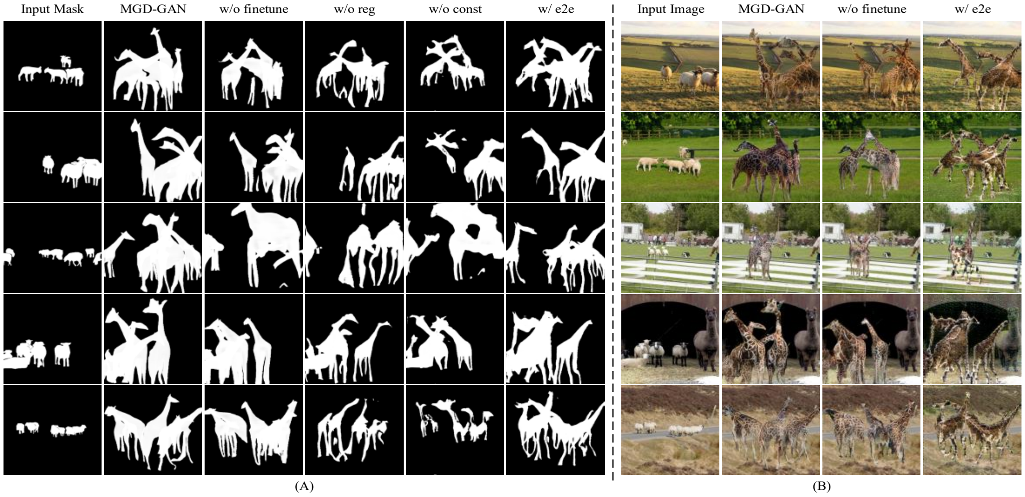

Figure 3: Qualitative ablation study on *sheep&giraffe* dataset. In (A), we illustrate the ablation study results of mask generation. In (B), we show the image results. Our model with all components performs the best in terms of mask and image generation.

ation $\mathcal{L}_{img}$ could be defined as:

$$\begin{aligned}\mathcal{L}_{img} =& \mathcal{L}^{G}_{img} + \mathcal{L}^{D}_{img} + \lambda_{fmap}\mathcal{L}_{fmap} \\ & + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_{feat}\mathcal{L}_{feat}.\end{aligned} \tag{11}$$

λ_{fmap}, λ_{vgg}, λ_{feat} are weights of $\mathcal{L}_{fmap}$, $\mathcal{L}_{vgg}$ and $\mathcal{L}_{feat}$.

2.3 Reversal Fine-tuning of Masks

After G_{img} and G_{mask} are both well trained, we use G_{img} to generate $\mathbf{I}^{'\mathcal{T}}$ based on the predicted masks from G_{mask}. Once the generated masks from G_{mask} do not follow the required data distribution, the generated foreground instances from G_{img} would be potentially judged as "fake" by D_{img}. Thus, G_{mask} would be encouraged to provide better mask generation. We name this fine-tuning rule after *Reversal Fine-tuning of Masks*, because the guidance from mask to image is forward directional. Note that, we fix all parameters in G_{img} and D_{img} when we conduct the fine-tuning training.

3 Experiments and Analysis

3.1 Implementation Details

We set batch size $N = 2$ for training. The mask, image and the fine-tuning part are trained for 100, 200 and 50 epochs, respectively. For hyper-parameters, we set $\lambda_{penalty}$ as 0.1, λ_{const} and λ_{reg} as 1, λ_{pc}, λ_{com}, λ_{fmap}, λ_{vgg} and λ_{feat} as 10. The Adam optimizer is adopted with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ and the learning rate $lr = 0.002$. All experiments are conducted on a NVIDIA Titan Xp GPU.

3.2 Datasets

MS COCO [Lin *et al.*, 2014]: Three domain pairs, *sheep&giraffe*, *elephant&zebra* and *bottle&cup*, are selected from MS COCO. Masks of each instance are provided.

Multi-Human Parsing [Zhao *et al.*, 2018]: Each image in MHP contains at least two persons (average 3) in crowd scenes. For each person, 18 semantic categories are defined and annotated, e.g. "skirt". Each annotated part corresponds to a binary mask. We select pair *pants&skirt* from MHP.

3.3 Evaluation Metrics

Existing evaluation metrics are not suitable for UDIT since UDIT focuses on the instance-level translation with no specific corresponding real guidance. Obviously, the more realistic the generated instances are, the more easily they would be detected. Inspired by this, we propose three novel metrics: *Mean Match Rate* (MMR), *Mean Object Detection Score* (MODS) and *Mean Valid IoU Score* (MVIS).

Specifically, we feed the generated images into the pretrained Mask-RCNN [He *et al.*, 2020] to get predicted labels, scores and masks. Then we use the generated masks to match the predicted ones as a retrieval process. MMR measures the ratio of the matched masks amount to the total masks amount. Since the predicted scores represent the confidence of being classified into the specific category, we use MODS to calculate average scores of being classified into the target domain. Besides, we design MVIS to evaluate the average IoU between the predicted masks and the generated ones. The three metrics measure the distance between the generated instances and the real ones comprehensively. The higher they are, the more realistic the generated instances are. Details of the metrics are illustrated in the appendix.

3.4 Comparison with State-of-the-arts

We choose two state-of-the-arts : CycleGAN [Zhu *et al.*, 2017] and InstaGAN [Mo *et al.*, 2019] as our competitors. For fairness, we augment CycleGAN with segmentation masks which is named as CycleGAN+seg. The quantitative and qualitative results are demonstrated in Tab. 1 and Fig. 2. For *giraffe2sheep* translation, as we can observe in Tab. 1, our model significantly surpasses InstaGAN by 31.0, 26.5 and 24.9 in MMR, MODS and MVIS metrics. The gaps become 34.7, 19.4 and 16.7 when it comes to *sheep2giraffe*

	sheep/giraffe			elephant/zebra			bottle/cup		
Method	MMR	MODS	MVIS	MMR	MODS	MVIS	MMR	MODS	MVIS
GT	63.3/88.8	72.9/94.7	62.5/77.6	80.5/87.1	87.2/94.9	77.3/79.4	36.8/46.1	37.2/46.4	34.3/44.7
InstaGAN	16.7/30.3	19.5/50.0	17.8/37.2	53.8/61.3	67.7/80.8	60.0/59.6	8.3/**22.7**	11.3/**20.8**	9.9/**20.6**
MGD-GAN	**47.7/65.0**	**46.0/69.4**	**42.7/53.9**	**76.0/83.7**	**81.2/85.9**	**71.4/72.0**	**17.4**/20.0	**17.0**/18.0	**15.3**/19.0

Table 1: Quantitative results of different methods on *sheep&giraffe*, *elephant&zebra* and *bottle&cup* datasets. For all metrics, higher is better. Note that 'GT' indicates 'Ground Truth'. 'sheep' means the results of generated sheep image from *giraffe2sheep* translation.

Method	sheep/giraffe		
	MMR	MODS	MVIS
A) MGD-GAN	**47.7**/65.0	**46.1**/69.4	**42.7**/53.9
B) A w/o reg	14.7/65.1	14.0/71.5	12.2/54.0
C) A w/o const	38.1/60.0	37.5/66.9	35.6/51.4
D) A w/o finetune	46.7/**67.1**	45.48/**74.7**	41.07/**59.8**
E) A w/ e2e	45.3/18.1	38.2/23.8	38.9/18.6

Table 2: Quantitative ablation results on *sheep&giraffe* dataset.

translation. Averagely, our MGD-GAN obtains nearly double scores than our best competitor InstaGAN. In the results of *zebra2elephant* translation, we win InstaGAN by a margin of 22.2, 13.5, 11.4 in MMR, MODS and MVIS metrics. As for *elephant2zebra* translation, the gaps are 22.4, 5.1, 12.4. Especially, our results approach the scores of ground truth which shows our high image quality. Though the scores of InstaGAN on *bottle2cup* translation are slightly higher than ours, the visual results still prove the effectiveness of ours as shown in the third row of Fig. 2. Moreover, the scores of InstaGAN on *bottle2cup* and *cup2bottle* are unbalanced, while our model achieves balanced results on all the datasets which proves the stability of our model.

The qualitative results in Fig. 2 show that, the visual results our model yields are more compelling. For the *sheep2giraffe*, our generated giraffes are more vivid. Contrary to the other two competitors, our generated sheep image owns better visual results without any sign of original instances. This proves the effectiveness of the inpainting operation. Comparatively, as demonstrated in Fig. 2, our generated elephants and zebras are still more lifelike though translation between the two domains is pretty easy. Since the bottles and cups are pretty similar in shape, InstaGAN and CycleGAN both fail to morph the masks. In contrast, as depicted in the third row of Fig. 2, our model successfully translates both the masks and the images. As for the clothes change shown in the last row of Fig. 2, though the InstaGAN morphs the skirt mask to the pants mask, it still fails to generate corresponding instance. While ours translates both which argues that the shape information is fully utilized in our model.

In particular, our model hugely cuts off the training time budget compared to our best competitor InstaGAN. Quantitatively, for the training of *sheep&giraffe*, our model consumes 57 hours totally, while InstaGAN takes about 150 hours.

3.5 Ablation Study

To demonstrate the effectiveness of our proposed functions and components, we conduct ablation study on *sheep&giraffe* dataset. We build four baseline models (B, C, D, E) totally. The quantitative and qualitative results of our ablation study are shown in Tab. 2 and Fig. 3, respectively.

First, the *Mask Regression Loss* $\mathcal{L}_{reg}$ is discarded in baseline B. In Tab. 2, we can observe the huge gap between B and A in sheep domain, though B is slightly higher than A in giraffe domain. That proves $\mathcal{L}_{reg}$ is key to the training balance. Besides, less constraints lead to failed mask generation, which can be verified in the fourth column in Fig. 3(A). Second, we train our model without *Mask Consistency Loss* $\mathcal{L}_{const}$ as baseline C. The scores of C are far behind A in both domains, which inversely proves the effectiveness of $\mathcal{L}_{const}$. The generated masks of C shown in fifth column of Fig. 3(A) become unrecognized, as the training process becomes unstable. Third, when we remove the mask fine-tuning process, as shown in Fig. 3(A), the mask generator fails to perform well on each instance. Besides, in Fig. 3(B), model without fine-tuning generates instances with obvious holes. Although in Tab. 2, baseline D surpasses our full model A slightly on the giraffe domain, visualization still proves the effectiveness of the fine-tuning. Fourth, we train our baseline model E in the end-to-end manner, which means we abandon the *Designed Supervision* and train the G_{mask} and G_{img} together. The scores of E in Tab. 2 decrease significantly in all metrics. The generated masks and images in Fig. 3(A) and Fig. 3(B) both demonstrate the bad performance of E. This argues the importance of our proposed training scheme. Combining Fig. 3 and Tab. 2, we can conclude that, our MGD-GAN with all components performs the best in mask and image generation.

4 Conclusion

In this paper, we propose an effective pipeline named MGD-GAN for Unsupervised Deformable-Instances Image-to-Image Translation (UDIT), which first generates target masks in batch and then utilizes them to guide the instance synthesis while rendering the whole image in a natural way. An elegant training procedure named Designed Supervision is proposed to transform the unsupervised mask-to-instance to a supervised one thus greatly promoting the image quality and the training stability. Experiments on four datasets argue that our method outperforms the state-of-the-art qualitatively and quantitatively.

Acknowledgements

This work is supported by National Key Research and Development Program of China (No. 2018AAA0102200), the National Natural Science Foundation of China (Grant No. 61772116, No. 61872064, No. 62020106008), Sichuan Science and Technology Program (Grant No.2019JDTD0005).

References

[Bhattacharjee *et al.*, 2020] Deblina Bhattacharjee, Seungryong Kim, Guillaume Vizier, and Mathieu Salzmann. DUNIT: detection-based unsupervised image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4786–4795, 2020.

[Chen *et al.*, 2018] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Eur. Conf. Comput. Vis.*, pages 167–184, 2018.

[Gao *et al.*, 2020] Lianli Gao, Junchen Zhu, Jingkuan Song, Feng Zheng, and Heng Tao Shen. Lab2pix: Label-adaptive generative adversarial network for unsupervised image synthesis. In *ACM MM*, 2020.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[He *et al.*, 2020] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):386–397, 2020.

[Huang *et al.*, 2018] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Eur. Conf. Comput. Vis.*, volume 11207, pages 179–196, 2018.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5967–5976, 2017.

[Ledig *et al.*, 2017] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 105–114, 2017.

[Lee *et al.*, 2018] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Eur. Conf. Comput. Vis.*, volume 11205, pages 36–52, 2018.

[Liang *et al.*,] Xiaodan Liang, Hao Zhang, Liang Lin, and Eric P. Xing. Generative semantic manipulation with mask-contrasting GAN. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Eur. Conf. Comput. Vis.*, Lecture Notes in Computer Science, pages 574–590.

[Lim and Ye, 2017] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *CoRR*, abs/1705.02894, 2017.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, volume 8693, pages 740–755, 2014.

[Liu and Tuzel, 2016] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS*, pages 469–477, 2016.

[Liu *et al.*, 2017] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, pages 700–708, 2017.

[Mejjati *et al.*, 2018] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NIPS*, pages 3697–3707, 2018.

[Mo *et al.*, 2019] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. In *Int. Conf. Learn. Represent.*, pages 2242–2251, 2019.

[Park *et al.*, 2019] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2337–2346, 2019.

[Shen *et al.*, 2019] Zhiqiang Shen, Mingyang Huang, Jianping Shi, Xiangyang Xue, and Thomas S. Huang. Towards instance-level image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3683–3692, 2019.

[Yi *et al.*, 2020] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7505–7514, 2020.

[Zhao *et al.*, 2018] Jian Zhao, Jianshu Li, Yu Cheng, Terence Sim, Shuicheng Yan, and Jiashi Feng. Understanding humans in crowded scenes: Deep nested adversarial learning and A new benchmark for multi-human parsing. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *ACM Int. Conf. Multimedia*, pages 792–800, 2018.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Int. Conf. Comput. Vis.*, pages 2242–2251, 2017.