Learning Low-Shot Generative Networks for Cross-Domain Data
Supplementary Materials

Hsuan-Kai Kao*
Academia Sinica
hkkao@iis.sinica.edu.tw

Cheng-Che Lee*
National Chiao Tung University
nctusunnerli.cs06g@nctu.edu.tw

Wei-Chen Chiu
National Chiao Tung University
walon@cs.nctu.edu.tw

(∗ indicates equal contribution.)

I. Clarification on Our Problem Scenario & Possible Confusion with Image-to-Image Translation

The main goal of our task is to learn a generator for the target domain, while there are only few target-domain examples (i.e. few-shot or low-shot) available. As a generator typically is capable of synthesizing data/images from random noise (i.e. a latent vector), it is quite different from the image-to-image translation (I2I) framework which basically learns a mapping function/network between two image domains. Since existing deep generative models in general requires large amount of training data, it would be almost impossible to learn a target-domain generator merely based on few target-domain examples. For instance, the **baseline** in our main manuscript is exactly to train a GAN model only based on the few target-domain examples, where the generated examples from this baseline model is with blurry and poor quality; we also try to train a simple VAE model by only using the target-domain samples. While taking 100 real faces as the target-domain data, the simple VAE model can only produce blurry results and get much poorer FID score according to our evalaution protocol. Some examples for the generated images produced by the simple VAE are given in the Figure 2.

Therefore, in this paper we propose a whole new problem setting, where there is another source-domain that is with rich data and highly related to target-domain (e.g. similar content but different appearance), and we would like to use the correspondence between these two domains to transfer the rich information of the related factor from source domain to the target, such that learning a target-domain generator becomes achievable. We would like to emphasize again here the difficulty of learning a low-shot generator upon cross-domain data and the difference from the image-to-image translation task. Moreover, even we try to use our cross-domain data to train the image-to-image translation model, e.g. cycleGAN, it usually suffers from the mode collapse problem, i.e. randomly-sampled source images will be translated into the same target output, due to little amount of target-domain samples, as the examples that we show in the Figure 3.

Figure 1 demonstrates the generative procedure of our proposed LaDo and GenHo models. After training, only the target-domain generator G_{tar} will be used in the testing time (please note that $G_{tar} = \{G^C, G^A_{tar}\}$ for the GenHo model), where we can randomly draw latent vector z as the input for G_{tar} to generated target-domain images $x_{tar} = G_{tar}(z)$.

II. Training with More Target-Domain Examples

In Figure 4 we show the qualitative results of using 500 target-domain examples. In addition, we experiment with the setting of having 1000 target-domain examples, more than what we use in the main manuscript (e.g. 50, 100 and 500 target-domain samples). The results are shown in the Table I and Figure 5. We can observe that when the number of target example grows increasingly, the performance of **Baseline++** becomes closer to ours, especially in Face data. However, we would like to emphasize again that our main goal in this paper is to tackle the difficult case of having only few-shots in the target-domain. The capability of our proposed methods, especially GenHo, can be well verified by the promising results compared to all the baselines when only a few number of target examples are available (e.g. $N_{tar} \leq 100$), as shown in the main manuscript.

III. Implementation Details

Figure 6 illustrates the building blocks for constructing the networks used in our proposed LaDo and GenHo models. Basically, four different types of residual blocks are utilized, namely *FirstResBlockD*, *ResBlockD*, *ResBlockG*, and *ResBlock*D^C. The postfix of the names represents which network component the residual block is adopted to (e.g., *ResBlockD* is used in the discriminators). Meanwhile, the layers framed with dashed lines are used only if the stride of the block is more than one. To form the networks used in LaDo, we have the same architecture for both Shoes and CelebA datasets, but a slight revision is deployed between the ones applied to Shoes and CelebA datasets under the framework of GenHo. More details of the structure for different networks can be found in Table II, Table III, and Table IV.

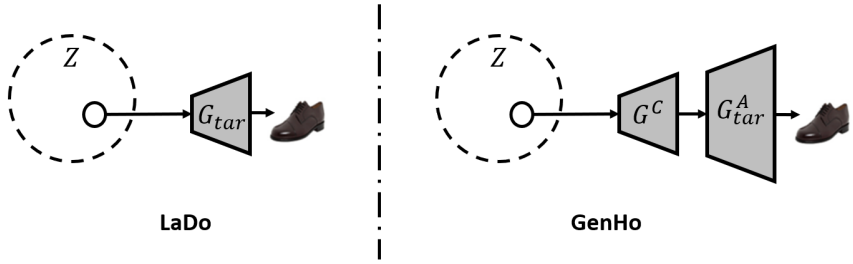


Fig. 1. The generative process of our proposed LaDo and GenHo models.

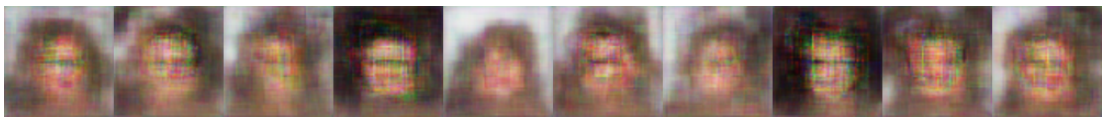


Fig. 2. Examples of the generated images produced by a simple VAE, which is only trained by using the target-domain samples (i.e. 100 real face images here).

Fig. 3. Examples of the generated images produced by a cycleGAN model, which is trained by using the cross-domain samples: 50k edge shoe images as the source-domain data and 100 real shoe images as the target-domain data .

TABLE I
THE FID COMPARISON BETWEEN DIFFERENT APPROACHES WITH 1000 TARGET-DOMAIN SAMPLES. THE BOLD VALUE REPRESENTS THE LOWEST ONE.

	N_{tar} =1000			
Target	Shoes		Faces	
Source	Edge	Style	Sketch	Style
Baseline	117.14		86.42	
Baseline++	45.05	38.75	17.81	**15.63**
CoGAN	49.52	52.31	170.31	198.39
LaDo	32.93	42.46	**16.62**	17.74
GenHo	**29**	**26.58**	16.77	16.74

IV. ABLATION STUDY

To explore the distinction brought by different selections of the content layers in GenHo, we visualize t-SNE scatter plots with 100 paired images on stylized CelebA dataset. As shown in Figure 7, three subfigures correspond to three different selections of content layers for G^C and E^C respectively, where the boundary between the content and appearance layers is chosen to be the one with the output size of 8×8, 16×16, or 32×32. The encoders E^C used in these three settings are trained for 100 epochs with the same training procedure. Each subfigure includes two images, the left one visualizes the fitting between $G^C(\mathcal{Z})$ and $E^C(X_{src})$, which should end up with overlapping distributions if $\mathcal{L}^{CA}_{src}$ is well converged, and the right one visualizes the distance of content information between paired distribution $E^C(x_{src,\kappa(i)})$ and $E^C(x_{tar,i})$, where the points should also be overlapping for each paired image if the encoder extracts the content information properly. As we can see, Figure 7 (a) shows that the distributions of $G^C(\mathcal{Z})$ and $E^C(X_{src})$ overlap each other, and so do the ones of the paired features. However, in Figure 7 (b) and (c), $E^C(x_{src,\kappa(i)})$ and $E^C(x_{tar,i})$ are close to each other while $G^C(\mathcal{Z})$ is far from $E^C(X_{src})$. We find that it could difficult to have both $\mathcal{L}^{CA}_{src}$ and $\mathcal{L}^{PS}$ converge at the same time when the boundary of content layers is closer to the image-space side, since the output of those layers may contain more appearance information than content information, which basically is diverge across different domains. Therefore, we empirically take the layer with output size of 8×8 and all the layers prior to it (i.e. closer to the latent space) to be our content layers G^Cin this paper.

V. CONTENT INFORMATION

To prove that our methods have learned the diverse content information from source domain data, which is never seen in target domain, we generate 50k images with different numbers of paired images N_{tar} for all models and randomly select

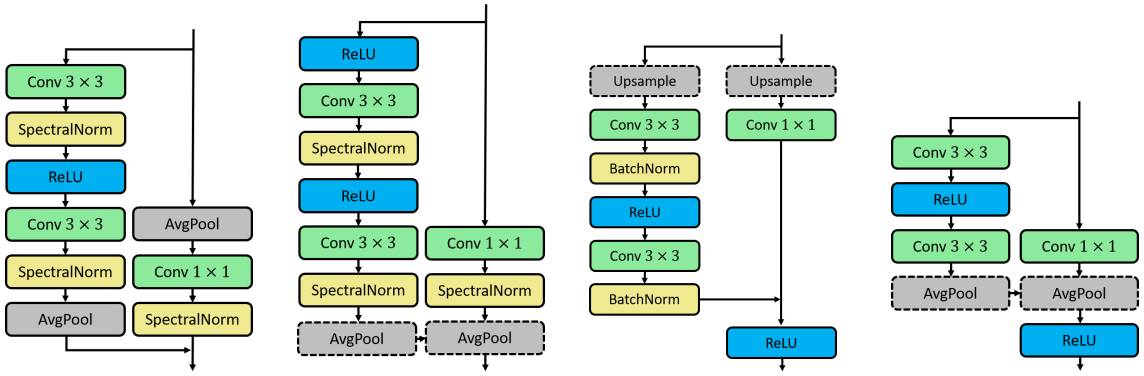
Fig. 4. Generated samples by various methods trained with 500 target-domain samples.

Fig. 5. The generated samples from different approaches trained with 1000 target-domain data.

1000 images. For comparison, we also randomly choose 1000 source-domain images from the dataset, and encode them with the encoder E^C in GenHo. The t-SNE scatter plots is used again to visualize the output of E^C for each model. Figure 8 shows the content mapping of 1000 images generated by different models on Shoes, and Figure 9 shows the one on CelebA. Obviously, **baseline** can not capture the real content, so it is far from the real distribution. Although CoGAN seems being able to capture a good content distribution on Shoes, it fails to do the same on CelebA due to its highly diverse content. Similar results are also shown in qualitative evaluation (please see our main manuscript). As the number of target samples increases, we can see that the distribution of **baseline++** spreads out to cover that of the real images. It demonstrates the improvement on the ability of **baseline++** to capture the real content when more target-domain images are provided for training. Nevertheless, when only a small number of samples are given, (i.e. $N_{tar} = 100$), our methods still cover most area of the real data distribution on both datasets, showing that they are benefited greatly from the rich content knowledge of the source data. In brief, our GenHo model can learn the content distribution which is the most similar to the one of the real/source-domain data.

VI. Examples of Cross-domain Pair

Here in Figure 10 we provide some examples of the cross-domain pairs that used in our experiments (for both Shoes and Faces), where the left two columns present the source images and the rightmost column presents the target images.



(a) FirstResBlockD (b) ResBlockD (c) ResBlockG (d) ResBlockD^C

Fig. 6. The building blocks for constructing the networks used in our proposed LaDo and GenHo models.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$	
Dense, BN, ReLU	$128 \times 4 \times 4$
ResBlockG up	$128 \times 8 \times 8$
ResBlockG up	$128 \times 16 \times 16$
ResBlockG up	$128 \times 32 \times 32$
ResBlockG up	$128 \times 64 \times 64$
3×3 conv, Tanh	$3 \times 64 \times 64$

(a) Generator

Image $x \in \mathbb{R}^{3\times64\times64}$	
FirstResBlockD	$128 \times 32 \times 32$
ResBlockD down	$128 \times 16 \times 16$
ResBlockD down	$128 \times 8 \times 8$
ResBlockD down	$128 \times 4 \times 4$
ResBlockD	$128 \times 4 \times 4$
ResBlockD	$128 \times 4 \times 4$
ReLU, AvgPool2d	$128 \times 1 \times 1$
Dense $\rightarrow$ 1	

(b) Discriminator

Image $x \in \mathbb{R}^{3\times64\times64}$	
3×3 conv ReLU 4×4 conv ReLU	$128 \times 32 \times 32$
3×3 conv ReLU 4×4 conv ReLU	$128 \times 16 \times 16$
3×3 conv ReLU 4×4 conv ReLU	$128 \times 8 \times 8$
3×3 conv ReLU 4×4 conv ReLU	$128 \times 4 \times 4$
(mean) 4×4 conv (logvar) 4×4 conv	$128 \times 1 \times 1$ $128 \times 1 \times 1$

(c) Encoder

TABLE II
THE NETWORK ARCHITECTURE OF LADO ON SHOES AND CELEBA.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$	
Dense, BN, ReLU	$128 \times 4 \times 4$
ResBlockG up	$128 \times 8 \times 8$
ResBlockG up	$128 \times 16 \times 16$
ResBlockG up	$128 \times 32 \times 32$
ResBlockG up	$128 \times 64 \times 64$
3×3 conv, Tanh	$3 \times 64 \times 64$

(a) Generator

Image $x \in \mathbb{R}^{3\times64\times64}$	
FirstResBlockD	$128 \times 32 \times 32$
ResBlockD down	$128 \times 16 \times 16$
ResBlockD down	$128 \times 8 \times 8$
ResBlockD down	$128 \times 4 \times 4$
ResBlockD	$128 \times 4 \times 4$
ResBlockD	$128 \times 4 \times 4$
ReLU, AvgPool2d	$128 \times 1 \times 1$
Dense $\rightarrow$ 1	

(b) Discriminator

Image $x \in \mathbb{R}^{3\times64\times64}$	
3×3 conv ReLU 4×4 conv ReLU	$128 \times 32 \times 32$
3×3 conv ReLU 4×4 conv ReLU	$128 \times 16 \times 16$
3×3 conv ReLU 4×4 conv ReLU	$128 \times 8 \times 8$

(c) Content Encoder

$E^C(x^{src}) \in \mathbb{R}^{128\times8\times8}$	
ResBlockD^C down	$128 \times 4 \times 4$
ResBlockD^C	$128 \times 4 \times 4$
AvgPool2d	$128 \times 1 \times 1$
Dense $\rightarrow$ 1	

(d) Content Discriminator

TABLE III
THE NETWORK ARCHITECTURE OF GENHO ON SHOES.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$	
Dense, BN, ReLU	$512 \times 4 \times 4$
ResBlockG up	$512 \times 8 \times 8$
ResBlockG up	$256 \times 16 \times 16$
ResBlockG up	$128 \times 32 \times 32$
ResBlockG up	$64 \times 64 \times 64$
3×3 conv, Tanh	$3 \times 64 \times 64$

(a) Generator

Image $x \in \mathbb{R}^{3 \times 64 \times 64}$	
FirstResBlockD	$64 \times 32 \times 32$
ResBlockD down	$128 \times 16 \times 16$
ResBlockD down	$256 \times 8 \times 8$
ResBlockD down	$512 \times 4 \times 4$
ResBlockD	$512 \times 4 \times 4$
ResBlockD	$512 \times 4 \times 4$
ReLU, AvgPool2d	$512 \times 1 \times 1$
Dense $\rightarrow$ 1	

(b) Discriminator

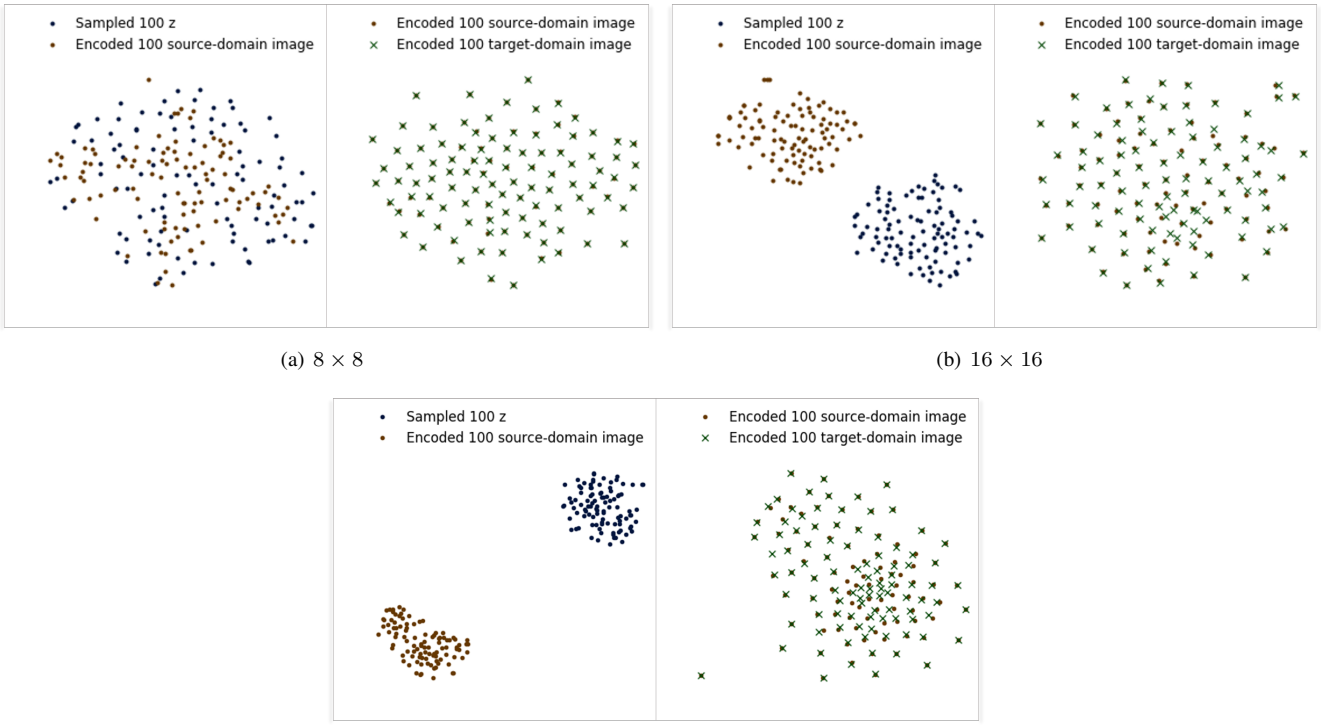
Image $x \in \mathbb{R}^{3 \times 64 \times 64}$	
3×3 conv ReLU 4×4 conv ReLU	$128 \times 32 \times 32$
3×3 conv ReLU 4×4 conv ReLU	$256 \times 16 \times 16$
3×3 conv ReLU 4×4 conv ReLU	$512 \times 8 \times 8$

(c) Content Encoder

$E^C(x^{src}) \in \mathbb{R}^{128 \times 8 \times 8}$	
ResBlockD^C down	$512 \times 4 \times 4$
ResBlockD^C	$512 \times 4 \times 4$
AvgPool2d	$512 \times 1 \times 1$
Dense $\rightarrow$ 1	

(d) Content Discriminator

TABLE IV
THE NETWORK ARCHITECTURE OF GENHO ON CELEBA.

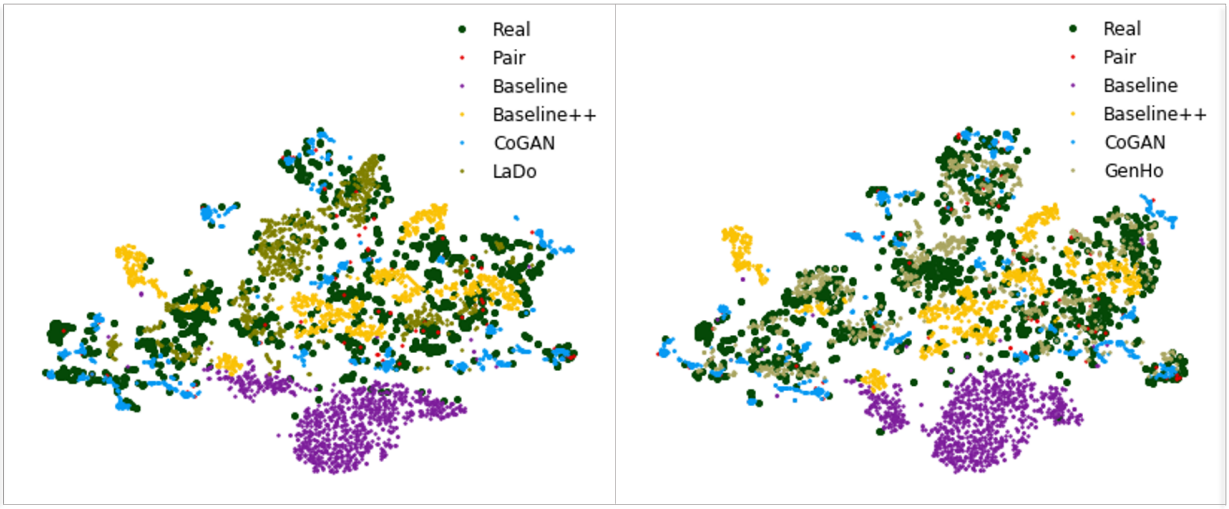


(a) 8×8

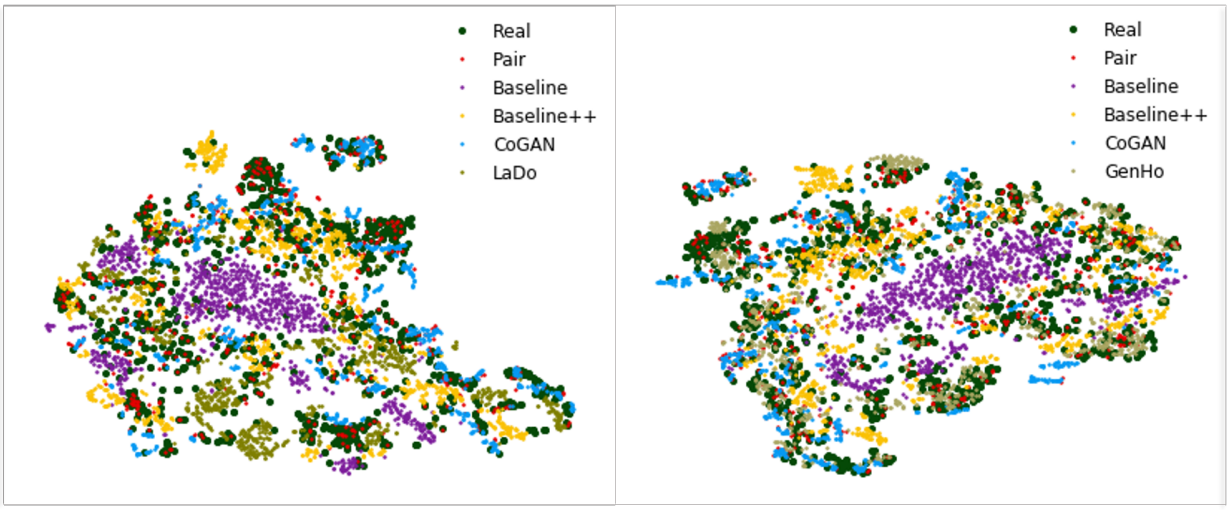
(b) 16×16

(c) 32×32

Fig. 7. The latent distribution generated with different selections of content layers.

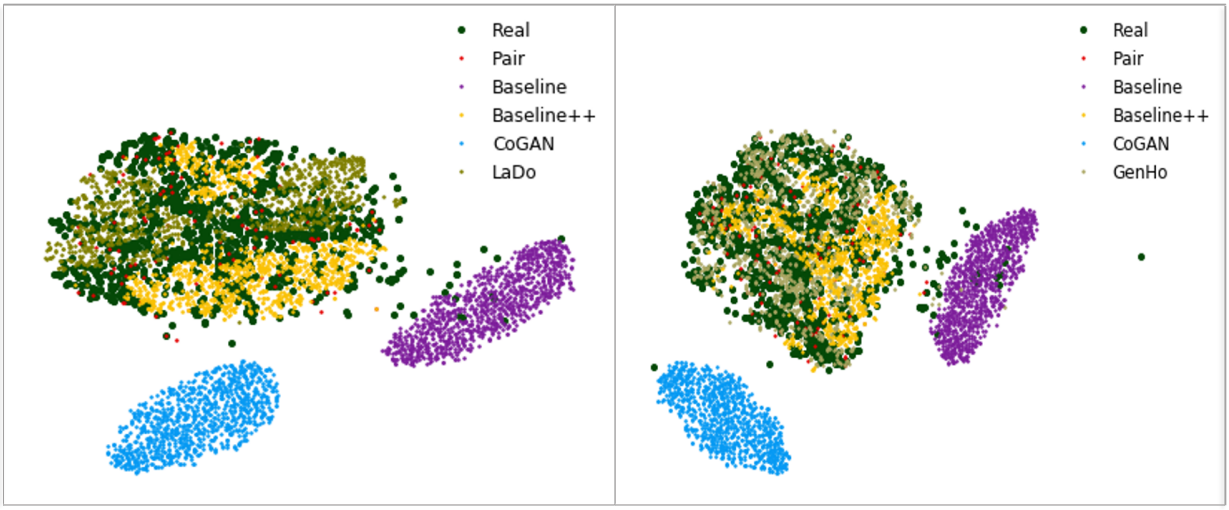


(a) Models are trained with 100 paired images. Left: comparing different models with respect to our LaDo framework. Right: comparing different models with respect to our GenHo framework.

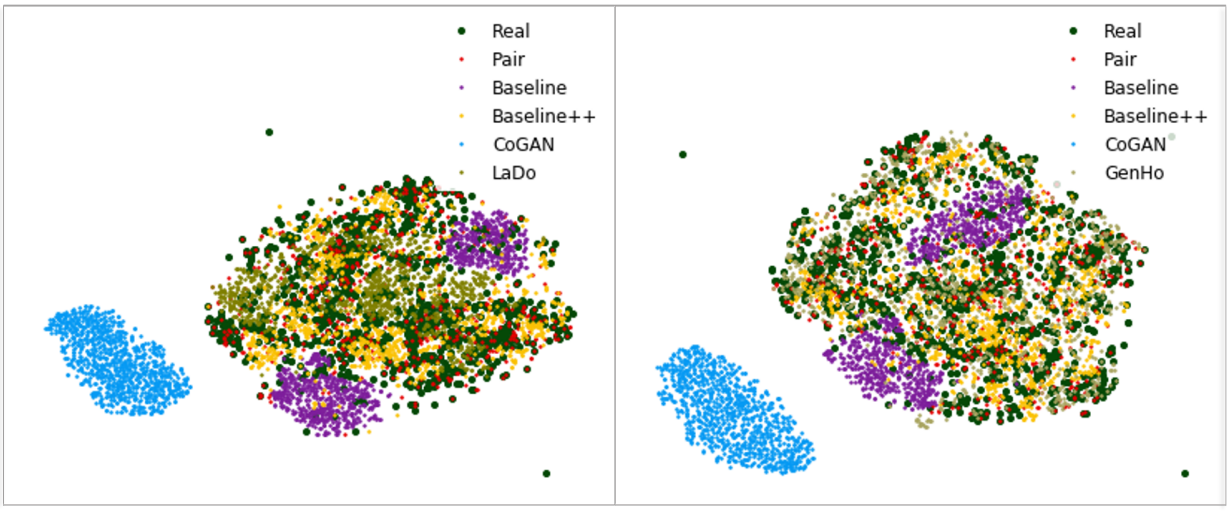


(b) Models are trained with 500 paired images. Left: comparing different models with respect to our LaDo framework. Right: comparing different models with respect to our GenHo framework.

Fig. 8. Content information generated with different models on Shoes.



(a) Models are trained with 100 paired images. Left: comparing different models with respect to our LaDo framework. Right: comparing different models with respect to our GenHo framework.



(b) Models are trained with 500 paired images. Left: comparing different models with respect to our LaDo framework. Right: comparing different models with respect to our GenHo framework.

Fig. 9. Content information generated with different models on CelebA.

Fig. 10. Top: Example cross-domain pairs of Shoes. Bottom: Example cross-domain pairs of Faces