Learning Low-Shot Generative Networks for Cross-Domain Data

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)

***Abstract*—We tackle a novel problem of learning generators for cross-domain data under a specific scenario of low-shot learning. Basically, given a source domain with sufficient amount of training data, we aim to transfer the knowledge of its generative process to another target domain, which not only has few data samples but also contains the domain shift with respect to the source domain. This problem has great potential in practical use and is different from the well-known image translation task, as the target-domain data can be generated without requiring any source-domain ones and the large data consumption for learning target-domain generator can be alleviated. Built upon a cross-domain dataset where (1) each of the low shots in the target domain has its correspondence in the source and (2) these two domains share the similar content information but different appearance, two approaches are proposed: a Latent-Disentanglement-Orientated model (LaDo) and a Generative-Hierarchy-Oriented (GenHo) model. Our LaDo and GenHo approaches address the problem from different perspectives, where the former relies on learning the disentangled representation composed of domain-invariant content features and domain-specific appearance ones; while the later decomposes the generative process of a generator into two parts for synthesizing the content and appearance sequentially. We perform extensive experiments under various settings of cross-domain data and show the efficacy of our models for generating target-domain data with the abundant content variance as in the source domain, which lead to the favourable performance in comparison to several baselines.**

I. Introduction

Deep generative models have been one of the most popular research topics nowadays in which the generative process of a data collection is approximated in an unsupervised-learning manner by the powerful capacity of deep neural networks. Without loss of generality, two most prominent models are the variational autoencoder (VAE [1]) and generative adversarial networks (GAN [2]), where the former is capable of learning the mapping from the data space to the latent representation, but suffers from synthesizing data of unsatisfying quality (e.g. blurry images), while the latter can generate synthetic data with better quality but is not able to infer the latent vector of a given data sample. The success of learning generative process largely relies on a huge quantity of training data, for capturing the underlying variation of data and being capable of synthesizing realistic output. The heavy data consumption for training deep generative models particularly limits their applicability when attempting to learn the generator for a novel data collection with little amount of samples. Moreover, the problem would get much harder when we consider to transfer the knowledge of generative process learnt from one data collection (denoted as *source domain*) to another (denoted as *target domain*) where these two domains are closely related but with discrepancy in data distribution, as known as *domain shift*.

The aforementioned difficulties of learning a novel concept from very few samples as well as coping with domain shift [3] are related to two well-known topics: *low-short learning* (also known as few-shot learning) and *domain adaptation*, respectively. Though many works have been proposed to tackle these two topics, they mainly focus on the discriminative tasks (e.g. classification). While we attempt to extend them for generative models, few issues appear: 1) Low-shot learning works for classification typically aim to transfer the features or metrics from the base classes into the novel ones, where these classes are implicitly assumed to be from the same domain, hence the domain shift across classes is not taken into consideration; 2) domain adaptation methods usually aim to match the data or feature distribution across domains, therefore the target domain generally needs massive amount of data and it could be problematic when only few samples are available.

In a nutshell, while given a source domain with rich data, we advance in this paper to discuss the problem of learning a target-domain generator on the condition that there only exists few target samples, together with the domain shift across source and target domains (see Figure 1 for illustration). To the best of our knowledge, we are the first work tackling this challenging problem, i.e. an intersection of generative models, low-shot learning, and domain adaptation. As an initial attempt towards resolving it, we propose to investigate a specific experimental setting for easing the difficulty: each sample in the target domain has a *correspondence* in the source domain, while the data samples of a cross-domain correspondence share the similar content/structure but have difference in appearance/texture. Two approaches are proposed to address the problem from different perspectives, which are Latent-Disentanglement-Oriented (**LaDo**) and Generative-Hierarchy-Oriented (**GenHo**) models.

The LaDo model is motivated by the characteristic of cross-domain dataset, where we learn to disentangle the latent representation of data onto the domain-invariant content space and the domain-specific appearance space. The rich information from source domain captured in the shared content space

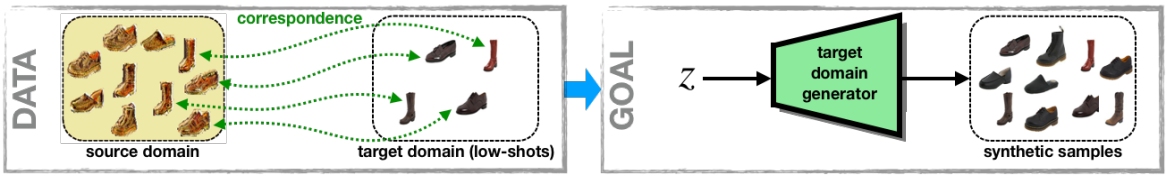

Fig. 1. Illustration of our problem setting. Assume we are given a cross-domain dataset composed a source domain full of training data, and a target domain which includes only low-shots but has correspondences (denoted as green lines) to the source domain, while both domains share the similar content but different appearance. We aims to utilize the knowledge transferred from the source domain for learning a target-domain generator that is able to synthesize samples with diverse content as in the source domain.

can then be propagated to the target domain. Together with the appearance features from the target data, a target-domain generator is learnt and capable of producing the synthetic data with more structure diversity as source samples. As for another GenHo model, an important assumption is further considered: during the generative procedure of a deep generator, the structure of a synthesized sample is mostly outlined in the first few layers of the network, then the detailed appearance and texture is gradually introduced by the later layers till the output space. Similar assumption also exists in other research works of image style transfer and GAN understanding [4]. Based on it, we decompose a generator into two sub-networks, i.e. content generator and appearance generator, and leverage the properties of cross-domain data to build a unique training procedure where the content generator in the source domain is particularly adapted for the target domain. By combining the content generator with the appearance generator, which is learnt upon the target-domain samples, the resultant target-domain generator is also able to synthesize diverse output as LaDo. Experiments with various settings of cross-domain datasets are conducted and the results successfully verify the superiority of our proposed models over several baselines.

II. Related work

Deep Generative Models. Recent advances in utilizing deep models to learn the generative procedure of a data collection and enable synthesizing new data samples (e.g. VAE and GAN) has spurred a lot of research interests. The generators learnt from both GAN and VAE basically try to find a mapping function between arbitrary points which are drawn from a prior (e.g standard normal distribution) to synthetic data samples that are ideally distributed as real data. Lots of research efforts [5], [6], [7] are devoted to improve both the fidelity and diversity of the generated samples, and discover the disentanglement for the latent factors of variation, while the mapping function built upon deep network generally gets more complicated and requires a great deal of training data during the learning. However, as the number of data samples in the target domain would be only a few in our problem scenario, learning typical generative models from scratch for the target-domain data could be problematic.

Low-Shot Learning. Based on the hypothesis that the knowledge learnt from a known data collection can still benefit making predictions on new dataset which only has few data samples (and annotations) available, lots of research efforts have been devoted to develop few-shot learning algorithms [8], [9] , in which most of them focus on the classification task. Among the recent progress of few-shot learning, some approaches [10], [11] utilize generative models for performing data augmentation in order to cope with the issue of data deficiency and improve classification for novel classes. However, as these approaches are generally based on the dataset with multiple classes and regardless of the domain shift across classes, they are not directly applicable to our problem setting, i.e. source domain constitutes of a single class and has domain shift with respect to the target domain. Recently, Chen *et al.* [12] investigate the progress of few-shot learning algorithms for classification, with a specific interest in evaluating their generalization ability toward cross-domain data (i.e. robustness w.r.t. domain shift). Surprisingly, they find out that a naive approach which is extended from the simple idea of fine-tuning can achieve competitive performance in comparison to several state-of-the-art approaches. Inspired by their finding, we also take this naive method (named as **Baseline++** in the experiments) into comparison, which utilizes the generator learnt from the source domain as a good initialization for the target-domain generator, and then uses the few data samples of target domain to perform fine-tuning.

Domain Adaptation. Domain adaptation basically deals with cross-domain data for the same task and has been widely used in different tasks, such as classification and segmentation. Most of the works on domain adaptation address the domain shift problem by learning the domain-invariant feature and matching the latent distributions across domains, where the recent advance of generative adversarial learning is widely adopted to achieve so [13], [14]. Despite the promising progress in using deep generative models to improve domain adaptation, the problem scenario that we focus in this paper is the other way around, i.e. to adapt the generator from source to target domain. There are several works of image-to-image translation (e.g., CycleGAN [15]) where the cross-domain data is considered but no target-domain generator is obtained as they solely attempt to learn the deterministic mapping between data across domains. Instead, Coupled GAN (CoGAN [16]) extends GAN to generate multi-domain images simultaneously, where both generator and discriminator are with partial weight sharing across domains to better tie the high-level information. Its extension in [17] (as known as UNIT) further includes an encoder to map the data into the latent space (i.e. the input for the generator) thus achieves image-to-image translation. Nevertheless, the learning for target-data generator typically requires a large amount of data, hence could suffer from mode

collapse or unstable training when given a dataset of small size, i.e. low-shot setting in our scenario. A recent work from [18] tackles the image-to-image translation problem with taking the few-shot setting into account. However, its training needs the supervised dataset composed of multiple object classes hence is different from our problem setting. We would like to emphasize here, while image translation models heavily rely on taking source-domain images as inputs/conditions during testing time, the main advantage of having the target-domain generator as our goal in this paper is that it can work as a standalone network and freely synthesize infinite target-domain images. The standalone generator could be even more beneficial for some tasks, such as medical applications, when related source-domain images are expensive to acquire.

III. Proposed Methods

Let $X_{src} = \{x_{src,i}\}_{i=1}^{N_{src}}$ and $X_{tar} = \{x_{tar,i}\}_{i=1}^{N_{tar}}$ denote the source-domain and target-domain data respectively, where $N_{src} \gg N_{tar}$. In our problem setting, we assume that for each data sample $x_{tar,i}$ from the target domain there is a corresponding sample $x_{src,\kappa(i)}$ from the source domain which shares the similar content as $x_{tar,i}$ but has different appearance, where $\kappa(\cdot)$ is a mapping function to obtain the index of the corresponding source sample for $x_{tar,i}$. We then denote the set of all the correspondence pair as $X^{Pair} = \{(x_{tar,i}, x_{src,\kappa(i)})\}_{i=1}^{N_{tar}}$.

As motivated previously, we propose two approaches (i.e. LaDo and GenHo) to address the problem of learning low-shot generators for cross-domain data, where domain-invariant content and domain-specific appearance information is used for (1) learning the disentangled latent space in the LaDo model, and (2) regularizing characteristics on different parts of a generator in the GenHo model. The target-domain generators obtained from both approaches are trained to be capable of synthesizing samples with diverse content as in the source domain. We introduce the details of our models in the following.

A. Latent-Disentanglement-Oriented Model

The basic idea behind our LaDo model is that, if the latent space of the target-domain data can be disentangled into two parts: (1) a domain-invariant subspace where the rich content features from source-domain data are well modelled, and (2) a domain-specific subspace which encodes the appearance provided by low-shot target-domain samples, then our goal of learning low-shot generative models for cross-domain data can be achieved when the target-domain generator learns to synthesize data samples based on the information drawn from these two subspaces. To this end, we adopt the architecture proposed by [19] here into our LaDo model, which consists of appearance encoders $\{E^A_{src}, E^A_{tar}\}$, generators $\{G_{src}, G_{tar}\}$, and domain discriminators $\{D_{src}, D_{tar}\}$ for both source and target domains, and a domain-invariant content encoder E^C. However, unlike the setting of [19] where both domains have a massive amount of training data, directly applying joint learning for all these networks would make it hard to learn the disentangled representation under our low-shot setting of target-domain data. Therefore, we advance to propose a two-step training procedure for our LaDo model (cf. Figure 2) for better handling our problem scenario and improving the overall training stability. We detail these steps below.

Stage-1: Learning source generator & disentanglement.

Given a source-domain data sample x_{src}, the content encoder E^C and source-domain appearance encoder E^A_{src} is used to map it into the content feature z^C_{src} and appearance feature z^A_{src} respectively. As the encoder-generator pair can be built up as a VAE model, where both data variances for content and appearance features are modelled by standard normal distributions $\mathcal{N}(0, I)$ in the latent space, there exists two objective functions, i.e., image reconstruction loss $\mathcal{L}^{IR}_{src}$ and KL-divergence loss $\mathcal{L}^{KL}_{src}$ for the source domain:

$$\begin{aligned}\mathcal{L}^{IR}_{src} &= \sum^{X_{src}} \left\| G_{src}(E^C(x_{src}), E^A_{src}(x_{src})) - x_{src} \right\| \\ \mathcal{L}^{KL}_{src} &= \mathbb{E}\left[D_{KL}(E^C(X_{src}) || \mathcal{N}(0, I)) \right] \\ &\quad + \mathbb{E}\left[D_{KL}(E^A_{src}(X_{src}) || \mathcal{N}(0, I)) \right].\end{aligned} \tag{1}$$

While now both content and appearance features in the latent space are regularized by two Gaussian distributions, denoted as $\mathcal{Z}^C_{src}$ and $\mathcal{Z}^A_{src}$ respectively, we can draw random samples from them as input for the generator G_{src} to produce synthetic source-domain samples $\tilde{X}_{src}$. The adversarial loss $\mathcal{L}^{IA}_{src}$ can be utilized to make $\tilde{X}_{src}$ more realistic:

$$\mathcal{L}^{IA}_{src} = \mathbb{E}[\log(D_{src}(X_{src}))] + \mathbb{E}[\log(1 - D_{src}(\tilde{X}_{src}))] \tag{2}$$

In addition, similar to [19], we have a latent regression loss $\mathcal{L}^{LR}$ for encouraging the invertible mapping between the latent and data spaces: By denoting (z^C_{src}, z^A_{src}) as $z^{C,A}$,

$$\mathcal{L}^{LR}_{src} = \sum \left| (E^C(G_{src}(z^{C,A})), E^A_{src}(G_{src}(z^{C,A}))) - z^{C,A} \right|. \tag{3}$$

However, the aforementioned loss functions (i.e. $\mathcal{L}^{IR}_{src}$, $\mathcal{L}^{KL}_{src}$, $\mathcal{L}^{IA}_{src}$, and $\mathcal{L}^{LR}_{src}$) can only help to learn a good source-domain generator but do not guarantee the disentanglement between the content and appearance features. We therefore leverage the correspondence pair X^{Pair}, where each cross-domain pair shares the similar content but different appearance, to enhance the disentanglement of learning domain-invariant content features as well as the domain-specific appearance features. We propose a paired-content loss $\mathcal{L}^{PC}$ to achieve so. Given a cross-domain pair $(x_{tar,i}, x_{src,\kappa(i)})$, as they share the similar content, the content encoder E^C is constrained to map them into the same content feature, thus $\mathcal{L}^{PC}$ is defined as:

$$\mathcal{L}^{PC} = \sum_{i=1}^{N_{tar}} \left\| E^C(x_{src,\kappa(i)}) - E^C(x_{tar,i}) \right\| \tag{4}$$

The full objective of the Stage-1 in LaDo training is:

$$\begin{aligned}\mathcal{L}_{stage-1} =& \lambda^{IR}\mathcal{L}^{IR}_{src} + \lambda^{KL}\mathcal{L}^{KL}_{src} + \lambda^{IA}\mathcal{L}^{IA}_{src} + \\ & \lambda^{LR}\mathcal{L}^{LR}_{src} + \lambda^{PC}\mathcal{L}^{PC}\end{aligned} \tag{5}$$

where λ hyperparameters control the balance between losses; in which $\{E^C, E^A_{src}\}$, G_{src}, and D_{src} are respectively optimized by $\{\mathcal{L}^{IR}_{src}, \mathcal{L}^{KL}_{src}, \mathcal{L}^{LR}_{src}, \mathcal{L}^{PC}\}$, $\{\mathcal{L}^{IR}_{src}, \mathcal{L}^{IA}_{src}, \mathcal{L}^{LR}_{src}\}$, and $\mathcal{L}^{IA}_{src}$.

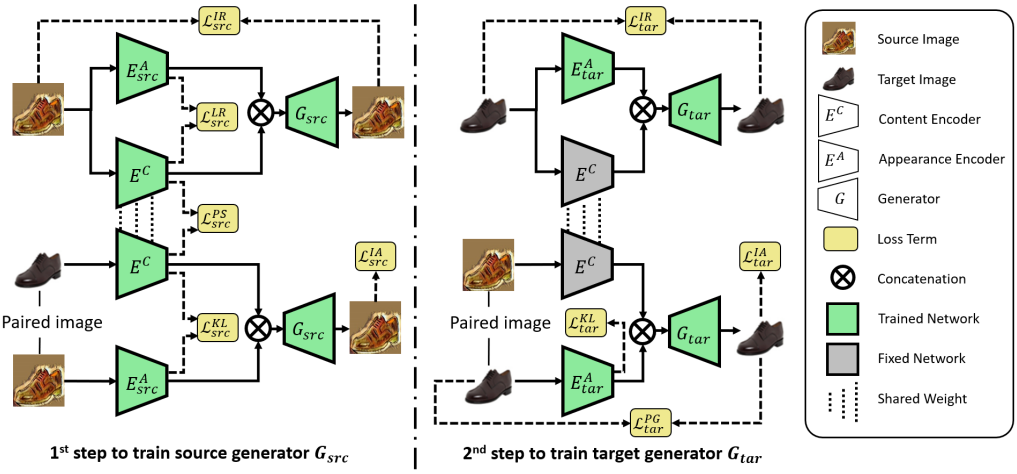

Fig. 2. Overview of our proposed Latent-Disentanglement-Oriented (**LaDo**) Approach.

Stage-2: Learning target-domain generator.
With both E^C and E^A_{src} well trained in the Stage-1, we keep the E^C fixed and use E^A_{src} to initialize the target-domain appearance encoder E^A_{tar} in the Stage-2. Most importantly, we start the learning of our main goal: target-domain generator G_{tar}. Several objectives are hence introduced as follows for encouraging G_{tar} to generate target-domain samples with the content diversity inherited from the source domain.

First, the image adversarial loss $\mathcal{L}^{IA}_{tar}$, image reconstruction loss $\mathcal{L}^{IR}_{tar}$, as well as the KL divergence loss of appearance feature $\mathcal{L}^{KL}_{tar}$ are also applied on the target domain:

$$\begin{aligned}\mathcal{L}^{IA}_{tar} =& \mathbb{E}[\log(D_{tar}(X_{tar}))] + \mathbb{E}[\log(1 - D_{tar}(\tilde{X}_{tar}))] \\ \mathcal{L}^{IR}_{tar} =& \sum^{X_{tar}} \left\| G_{tar}(E^C(x_{tar}), E^A_{tar}(x_{tar})) - x_{tar} \right\| \\ \mathcal{L}^{KL}_{tar} =& \mathbb{E}\left[D_{KL}(E^A_{tar}(X_{tar}) || \mathcal{N}(0, I)) \right]\end{aligned} \tag{6}$$

In particular, please note that the synthetic target-domain samples $\tilde{X}_{tar}$ are obtained by G_{tar} with having the input drawn from latent source-domain content distribution $\mathcal{Z}^C_{src}$ and the latent target-domain appearance distribution $\mathcal{Z}^A_{tar}$.

Second, we use again the correspondence pair X^{Pair} to define a pair generation loss $\mathcal{L}^{PG}$. Basically, given a cross-domain pair $(x_{tar,i}, x_{src,\kappa(i)})$, when we take the content feature $E^C(x_{src,\kappa(i)})$ extracted from $x_{src,\kappa(i)}$ and the appearance feature $E^A_{tar}(x_{tar,i})$ as the input for the target-domain generator G_{tar}, the generated output $\tilde{x}_{tar,i}$ should nicely reconstruct $x_{tar,i}$, since $x_{tar,i}$ and $x_{src,\kappa(i)}$ ideally should have the same content information. $\mathcal{L}^{PG}$ is thus defined as:

$$\mathcal{L}^{PG} = \sum_{i=1}^{N_{tar}} \left\| G_{tar}(E^C(x_{src,\kappa(i)}), E^A_{tar}(x_{tar,i})) - x_{tar,i} \right\| \tag{7}$$

where we can see that both $\mathcal{L}^{PG}$ and $\mathcal{L}^{IA}_{tar}$ motivate G_{tar} to produce target-domain samples of having rich content information transferred from the source-domain.

Finally, the full objective function used in the Stage-2 is:

$$\mathcal{L}_{stage-2} = \lambda^{IR}\mathcal{L}^{IR}_{tar} + \lambda^{KL}\mathcal{L}^{KL}_{tar} + \lambda^{IA}\mathcal{L}^{IA}_{tar} + \lambda^{PG}\mathcal{L}^{PG} \tag{8}$$

where $\{\mathcal{L}^{IR}_{tar}, \mathcal{L}^{KL}_{tar}, \mathcal{L}^{PG}\}$, $\{\mathcal{L}^{IR}_{tar}, \mathcal{L}^{IA}_{tar}, \mathcal{L}^{PG}\}$, and $\mathcal{L}^{IA}_{tar}$ are used to optimize E^A_{tar}, G_{tar}, and D_{tar} respectively. The hyperparamters λ in both stages are simply tuned to let each objective contribute equally (i.e. with similar numerical range), and a unified setting of hyperparameters are adopted across all our experiments. Please note again that the goal of our task is to learn the unconditional image generation, i.e. we only use the target-domain generator G_{tar} during the test time to generate target-domain images based on randomly-sampled latent vectors z, which is quite different to image-to-image translation (where the source-domain images are required during test time to synthesize target-domain ones).

B. Generative-Hierarchy-Oriented Model

In the LaDo approach, we leverage the cross-domain dataset composed of low-shot target-domain data with source-domain correspondence, and attempt to learn the latent space disentanglement. As the domain-invariant space of content information is discovered, the target-domain generator is able to take the diverse content from source-domain for enriching its generated data distribution. The capability of target-domain generator thus heavily relies on having latent space well disentangled, which could be occasionally hard to achieve or suffer from unstable training due to the challenge of low-shot scenario.

Here we further propose another novel approach, Generative-Hierarchy-Oriented model (GenHo), that aims to directly integrate the characteristics of our cross-domain data into the underlying generative procedure captured by a generator. Particularly, we follow an important assumption as in [20], [7] that the first few layers (closer to the latent space) of an image generator are responsible for producing high-level content representation of the synthetic output (e.g., the shape or rough structure), while the remaining layers sequentially paint fine-grained details in appearance or texture. With comparison to LaDo, our GenHo model decomposes a generator G into a cascade of two sub-networks $\{G^C, G^A\}$, presenting content-generator and appearance-generator respectively. In other words, now the disentanglement between content and appearance information happens within the network architecture of a

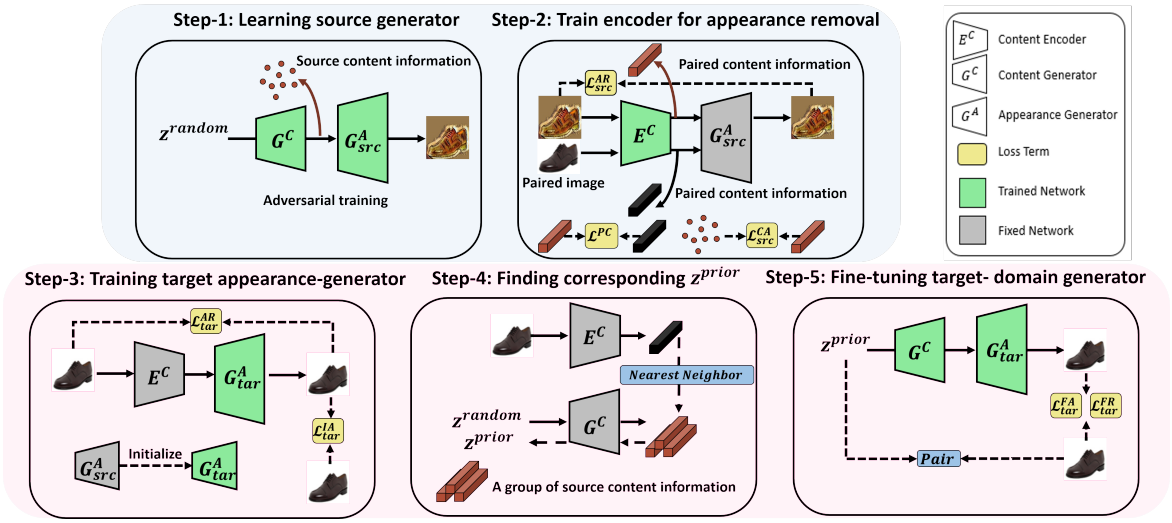

Fig. 3. Overview of our proposed Generative-Hierarchy-Oriented (**GenHo**) approach.

generator, where G^C is learned specifically by using the cross-domain pairs with similar content but distinct appearance.

Training procedure The source-domain and target-domain generators in our proposed GenHo approach attempt to use the same content-generator G^C but have their own appearance-generators, i.e. G^A_{src} and G^A_{tar}. In another words, $G_{src} = \{G^C, G^A_{src}\}$ and $G_{tar} = \{G^C, G^A_{tar}\}$. Instead of training all networks at once, the training procedure of GenHo are carefully designed and decomposed into several steps as illustrated in Figure 3, in which they are able to be categorised by two groups: *content-related steps* and *appearance-related steps* (shaded by light-blue and light-red blocks respectively). For the former group of content-related steps, the cross-domain data is used for training; while in the appearance-related steps only the target-domain data is utilized.

Content-related steps This part including two steps is mainly responsible to capture the distribution of diverse content information from the source domain, for the future use in the target-domain generator. Basically, Step-1 ensures G^C to produce the content distribution; while Step-2 learns an encoder E^C which aims to not only map the images into latent vectors of content information but also match the content latent vectors between source-target correspondences.

Step-1: Learning source generator. The source-domain generator $G_{src} = \{G^C, G^A_{src}\}$ is learnt as the typical GANs. In other words, we directly train the source-domain generator G_{src} and simply split it into two parts (i.e. G^C and G^A_{src}) after training. With a large amount of source-domain training data X_{src} available, G_{src} ideally is able to produce images of high fidelity. Here we denote the feature space produced by $G^C(z^{random})$ as $\mathcal{Z}^C$, where $z^{random} \sim \mathcal{N}(0, I)$.

Step-2: Training encoder for appearance removal. As we assume that source and target domains share the content information, it is necessary to have a function mapping back from data space to $\mathcal{Z}^C$ in order to discover the rich content provided by source domain, and bridge it with the target domain. This mapping function can then be treated as an encoder of appearance removal, denoted as E^C, and it builds up an encoder-decoder pair with source appearance-generator G^A_{src}. With keeping G^A_{src} fixed, we propose three objectives to learn E^C, i.e. appearance reconstruction loss $\mathcal{L}^{AR}_{src}$, content adversarial loss $\mathcal{L}^{CA}_{src}$, and paired-content loss $\mathcal{L}^{PC}$. First, for a source-domain sample x_{src}, through the process of removing its appearance by E^C then generating the removed appearance back again by G^A_{src}, the output $\tilde{x}_{src}$ should nicely reconstruct x_{src}. $\mathcal{L}^{AR}_{src}$ is then defined as:

$$\mathcal{L}^{AR}_{src} = \sum^{X_{src}} \left\| G^A_{src}(E^C(x_{src})) - x_{src} \right\| \tag{9}$$

Second, we encourage the distribution-matching between of $E^C(X_{src})$ and $G^C(z^{random})$, where we adopt adversarial learning technique to achieve so. The $\mathcal{L}^{CA}_{src}$ is written as:

$$\mathcal{L}^{CA}_{src} = \mathbb{E}[\log D^C(G^C(\mathcal{Z}))] + \mathbb{E}[\log(1 - D^C(E^C(X_{src})))] \tag{10}$$

where $\mathcal{Z}$ presents samples drawn from $\mathcal{N}(0, I)$ and D^C is the discriminator used in the adversarial learning here. Finally, we bridge the shared content information in $\mathcal{Z}^C$ for each cross-domain pair $(x_{tar,i}, x_{src,\kappa(i)})$, by encouraging $E^C(x_{tar,i})$ and $E^C(x_{src,\kappa(i)})$ as close as possible. Hence

$$\mathcal{L}^{PC} = \sum_{i=1}^{N_{tar}} \left\| E^C(x_{src,\kappa(i)}) - E^C(x_{tar,i}) \right\| \tag{11}$$

Appearance-related steps The remaining steps (i.e., Step-3, Step-4, and Step-5) in the training procedure focus on integrating the well-trained content space with the target-domain appearance-generator G^A_{tar} to build up the final target-domain generator G_{tar}. Step-3 trains G^A_{tar} to generate target-domain images by taking content latent vectors as input. To further boost the training of G_{tar}, Step-4 finds the matches between the low-shot target-domain samples and the latent prior z^{prior} in which these matches are used by Step-5 to fine-tune the whole target-domain generator $G_{tar} = \{G^C, G^A_{tar}\}$.

Step-3: Training target appearance-generator. After training E^C in Step-2, it is now able to remove the appearance of target-domain data X_{tar}. We then keep E^C fixed and utilize it to

help learning of target appearance-generator G^A_{tar}, which is empirically initialized by the weights from G^A_{src}. As now E^C and G^A_{tar} together become an encoder-decoder pair, we adopt an image adversarial loss $\mathcal{L}^{IA}_{tar}$ and an appearance reconstruction loss $\mathcal{L}^{AR}_{tar}$ for training G^A_{tar}:

$$\begin{aligned}\mathcal{L}^{IA}_{tar} &= \mathbb{E}[\log(D_{tar}(X_{tar}))] + \mathbb{E}[\log(1 - D_{tar}(\tilde{X}_{tar}))] \\ \mathcal{L}^{AR}_{tar} &= \sum^{X_{tar}} \left\| G^A_{tar}(E^C(x_{tar})) - x_{tar} \right\|\end{aligned} \tag{12}$$

where D_{tar} is the target-domain discriminator and $\tilde{X}_{tar}$ is computed by $G^A_{tar}(E^C(X_{tar}))$.

Step-4: Finding corresponding z^{prior}. Till now, we have trained once all the components of target generator G_{tar}, i.e. $\{G^C, G^A_{tar}\}$. However, as they are trained in different steps, there potentially exists discrepancy between them. For addressing this concern, we propose to find the corresponding z^{prior} in the latent space for all target-domain data X_{tar}. Basically, we first sample a large number of z^{random} and feed them into G^C to get plenty of corresponding feature vectors $G^C(z^{random})$ in $\mathcal{Z}^C$. We then search from these $G^C(z^{random})$ to get the nearest neighbors for each of the $E^C(x_{tar})$. Derived from the matches between $G^C(z^{random})$ and $E^C(x_{tar})$, we get each $x_{tar,i}$ its corresponding z^{random} which is denoted as $z^{prior,i}$ for clarity.

Step-5: Fine-tuning target-domain generator. As the training of previous steps could be still imperfect, here we fine-tune the whole the target-domain generator which is able to provide feedback to update the previous components in the early stages. Based on the pairs of $\{z^{prior,i}, x_{tar,i}\}$ we found from Step-4, the holistic fine-tuning on $G_{tar} = \{G^C, G^A_{tar}\}$ is performed by adopting the two loss functions: $\mathcal{L}^{FA}_{tar}$ and $\mathcal{L}^{FR}_{tar}$. In which $\mathcal{L}^{FA}_{tar}$ is defined in a similar way as $\mathcal{L}^{IA}_{tar}$ but now $\tilde{X}_{tar}$ is obtained from $G_{tar}(z)$, where $z \sim \mathcal{N}(0, I)$; while another $\mathcal{L}^{FR}_{tar}$ follows the similar idea as [21] to train G^C by minimizing the objective: $\sum_{i=1}^{N_{tar}} \left\| G_{tar}(z^{prior,i}) - x_{tar,i} \right\|$.

We emphasize here, our motivation of proposing two methods is to provide insights and initial attempts on resolving this challenging problem from different perspectives: LaDo aims to disentangle the latent space while GenHo decomposes the generative procedure. Basically these methods have their own pros and cons: LaDo has less complexity in training (only two steps needed) while GenHo usually can produce results with better quality as shown later in experiments.

IV. Experiment

A. Datasets and Baselines.

50K images are randomly sampled from UT Zappos50K [22] and CelebA [23] respectively to build up two datasets for our experiments, where the image size is set to 64×64. These experimental datasets are transformed with edge detection [15] and style transfer [24] to form the source-domain data, where the style is randomly chosen from WikiArt dataset. The target domain only consists of a limited number of randomly sampled real/original images in which each of them is paired with its corresponding image from the source domain. That is, we are aiming to transfer the knowledge from the source domain composed of sketchy or stylized images to learn a target-domain generator for the real data. For exploring the difference in capacity between various approaches, the number of target-domain samples is set to be 50, 100, and 500. The implementation details are provided in the supplement.

We compare our proposed methods with three different baseline models, including **Baseline**, **Baseline++**, and **CoGAN** [16]: Baseline is a GAN model trained from scratch with adversarial learning based on target-domain samples only; Baseline++ takes the well-learnt source generator G_{src} as its initialization then fine-tunes on the target-domain data via adversarial learning; CoGAN [16] aims to learn generators for cross-domain data but does not tackle the low-shot setting. The network architecture for the generators used in these baseline models are similar to the ones in our proposed approaches thus we have fair comparison in terms of network capacity. Note that we exclude other baselines (e.g. CycleGAN) from image translation works since they generally do not have the standalone target-domain generator and also could suffer from the low-shot setting of our problem scenario (cf. supplement).

B. Quantitative Evaluation

We adopt Fréchet Inception distance (FID), which is commonly used in GAN-related works to measure the diversity and the quality of generated images, for our quantitative evaluation (FID values lower the better) to compare the generative capability on target-domain data. FID measures the similarity between two groups of images (e.g. the generated target-domain images and the real-world ones in this paper). Note that here we do not adopt another popular Inception score (IS) as our metric, since it has issues on the usage beyond ImageNet dataset , thus being unsuitable for our case (as human faces and shoes are not included in ImageNet).

The quantitative results under various settings of cross-domain data and number of target-domain samples (i.e. N_{tar}) are shown in Table I. We draw several observations here: (1) **Baseline** performs the worst and suffers severely from the lack of diversity and potential overfitting since it only relies on little amount of target-domain data and is trained from scratch. We can see that even when N_{tar} grows up to 500, its performance is still far below the others thus significantly requiring more data to train; (2) **CoGAN** performs occasionally fine on some cases such as shoe-dataset, but has problems on the face-dataset which has a larger data diversity. Also, its performance is unstable across different settings of N_{tar} or different types of source data (e.g. the huge gap between the performance of having stylized and sketchy face images as source-data, $N_{tar} = 100$); (3) **Baseline++** performs surprisingly well in average and continues improving while N_{tar} increases, this is analogous to the finding in the task of low-shot classification pointed out by [12], where the transfer-learning-based approach is able to handle (up to a certain degree) low-shot learning and cross-domain data. (4) Both our LaDo and GenHo generally obtain the best or competitive performance in comparison to all the baselines. It verifies that our models can well capture the content diversity from the source-domain and are able to

TABLE I
THE FID COMPARISON BETWEEN DIFFERENT APPROACHES WITH 50, 100, AND 500 TARGET-DOMAIN SAMPLES UNDER VARIOUS EXPERIMENTAL SETTINGS.

	N_{tar} =50				N_{tar} =100				N_{tar} =500			
Target	Shoes		Faces		Shoes		Faces		Shoes		Faces	
Source	Edge	Style	Sketch	Style	Edge	Style	Sketch	Style	Edge	Style	Sketch	Style
Baseline	199.46		233.15		169.45		237.73		125.74		150.94	
Baseline++	114.72	166.87	54.83	28.78	79.13	89.24	30.81	24.6	55.04	45.66	20.57	**15.65**
CoGAN	187.40	166.50	186.14	179.53	50.42	46.85	184.39	57.64	62.78	44.19	211.48	188.83
LaDo	**66.16**	**84.89**	51.44	57.03	**42.73**	49.95	28.63	30.42	35.05	44.36	18.21	17.74
GenHo	76.67	100.78	**32.78**	**24.34**	43.77	**43.19**	**19.68**	**17.07**	**28.4**	**23.06**	**16.25**	17.51

generate high-quality target-domain samples. Also, we find that GenHo in most cases is better than LaDo, we believe it is due to the careful design of having the cross-domain data well integrated into the generative process of generator. Moreover, it is worth noting that our models work much better than the others under the case of having small N_{tar} (e.g. 100); in other words, our models can provide a simple way to learn target-domain generator with good quality while only requiring small amount of effort to collect low-shot cross-domain pairs, which is of great potential in practical usages.

C. Qualitative Evaluation

We compare the performance of different models visually on both datasets (shoe images and face ones, with stylized data as the source domain). Figure 4 and Figure 5 demonstrate the results produced by training with 50 and 100 target-domain images respectively (Results with $N_{tar} = 500$ are in the supplement). In the first two rows of every figure, we show the examples of cross-domain pairs used for model training.

As we can see from the results, the **Baseline** model generates blurry and poor synthetic images since it is only trained from little amount of the target-domain data where the adversarial learning is hard and problematic; **CoGAN** can produce more realistic results for shoe images than the baseline model but it is suspected of trying to memorize the samples provided in the training process. Such issue can be observed in Figure 5 where the first five shoes images generated by CoGAN are almost the same as the real images. Moreover, this property of potentially memorizing training samples under the low-shot setting would lead to severe problem of mode collpase for the dataset with larger diversity such as CelebA. As also pointed out in the quantitative performance, in comparison to the Baseline and CoGAN models, **Baseline++** can generate samples of better quality and diversity for both shoes and face images. However, we can still observe from the shoe images synthezied by Baseline++, there are defective shape and the lack of details. In contrast, our method can simultaneously give results of good quality, capture diverse content from source-domain data, and better keep the content details. In particular, our GenHo model performs the best and we contribute this to the design choice that the characteristics of the cross-domain data are directly integrated into the generative procedure captured by the generator. Please refer to our supplement for more experiments, ablation study, and discussion.

D. Automatic Content Matching

For exploring more realistic scenario, we now attempt to alleviate the requirement of manually annotating the cross-domain correspondences. Following our assumption that the cross-domain pairs share the similar content but different appearance, for each of the few-shot target-domain samples, we adopt the *content similarity* to find its best match from the source domain and build up the cross-domain pairs automatically. For instance, here we randomly select 50k male faces and 100 female faces from CelebA dataset as the source-domain and target-domain data respectively. By utilizing the facial landmark detection algorithm, for each of the female images we find from the source-domain data (i.e male faces) a best match with the most similar configuration of facial landmarks (as the examples shown in Figure 6(a)), thus the cross-domain correspondences are constructed and used for learning our proposed methods. Even when the cross-domain pairs found in such manner are not perfectly aligned, our proposed models can still produce prominent results in comparison to other baselines without suffering from mode collapse, as shown in Figure 6(b). Regarding other data types in addition to the face images, different measurements of content similarity can be adopted for achieving the cross-domain matching, e.g. the high-level features extracted from the ImageNet-pretrained VGG network, which are actually widely used to obtain the structural content of an image.

V. CONCLUSION

We propose a novel problem scenario of learning generative models from cross-domain data under the low-shot learning scheme. We propose two different models, LaDo and GenHo, which are capable of generating results in the target domain with having diverse content obtained from the source domain. The efficacy of both the proposed models are verified through throughout experiments. Our project page is at https://github.com/SunnerLi/Low-Shot-GAN

Acknowledgement This project is supported by the Ministry of Science and Technology of Taiwan under grant MOST-109-2634-F-009-015, MOST-109-2634-F-009-020, and MOST-109-2636-E-009-018, and we are grateful to the National Center for High-performance Computing of Taiwan for computer time and facilities.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.
[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.
[3] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, 2015.

Fig. 4. Generated samples by various methods trained with 50 target-domain samples.

Fig. 5. Generated samples by various methods trained with 100 target-domain samples.

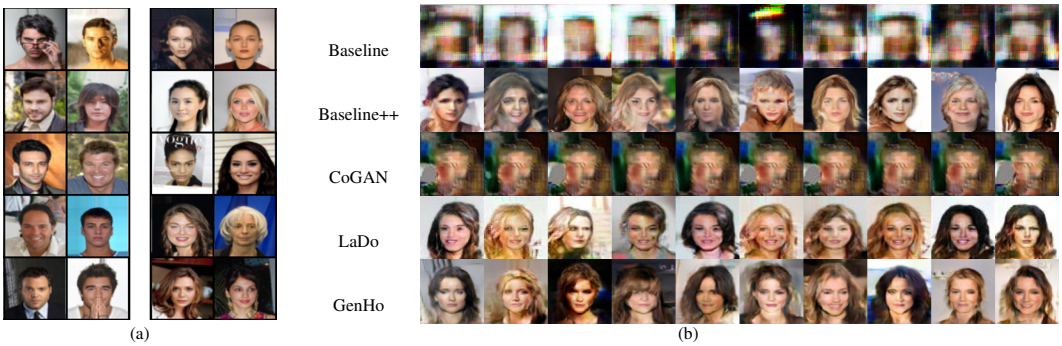

Fig. 6. (a) visualizes the example pairs of male and female faces, found by using the similarity of spatial configuration of facial landmarks. Left: male. Right: female. (b) shows the generated samples from different approaches (target-domain data: 100 female).

[4] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," *ArXiv:1811.10597*, 2018.

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *ArXiv:1710.10196*, 2017.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *ArXiv:1805.08318*, 2018.

[7] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *ArXiv:1812.04948*, 2018.

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018.

[9] M. Ye and Y. Guo, "Deep triplet ranking networks for one-shot recognition," *ArXiv:1804.07275*, 2018.

[10] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," in *International Conference on Learning Representations (ICLR) Workshops*, 2017.

[11] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *CVPR*, 2018.

[12] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015.

[14] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

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

[16] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *NeurIPS*, 2016.

[17] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NeurIPS*, 2017.

[18] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *ICCV*, 2019.

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

[20] N. Bodla, G. Hua, and R. Chellappa, "Semi-supervised fusedgan for conditional image generation," in *ECCV*, 2018.

[21] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *CVPR*, 2018.

[22] A. Yu and K. Grauman, "Fine-Grained Visual Comparisons with Local Learning," in *CVPR*, 2014.

[23] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[24] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.