Conditional Adversarial Generative Flow for Controllable Image Synthesis

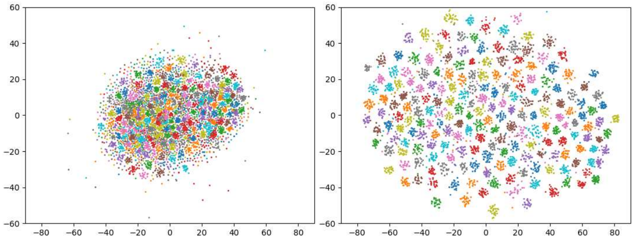
Rui Liu[1] Yu Liu[1] Xinyu Gong[2] Xiaogang Wang[1] Hongsheng Li[1]
[1]CUHK-SenseTime Joint Laboratory, Chinese University of Hong Kong [2]Texas A&M University
ruiliu@cuhk.edu.hk xy_gong@tamu.edu
{yuliu, xgwang, hsli}@ee.cuhk.edu.hk

Abstract

Flow-based generative models show great potential in image synthesis due to its reversible pipeline and exact log-likelihood target, yet it suffers from weak ability for conditional image synthesis, especially for multi-label or unaware conditions. This is because the potential distribution of image conditions is hard to measure precisely from its latent variable z. In this paper, based on modeling a joint probabilistic density of an image and its conditions, we propose a novel flow-based generative model named conditional adversarial generative flow (CAGlow). Instead of disentangling attributes from latent space, we blaze a new trail for learning an encoder to estimate the mapping from condition space to latent space in an adversarial manner. Given a specific condition c, CAGlow can encode it to a sampled z, and then enable robust conditional image synthesis in complex situations like combining person identity with multiple attributes. The proposed CAGlow can be implemented in both supervised and unsupervised manners, thus can synthesize images with conditional information like categories, attributes, and even some unknown properties. Extensive experiments show that CAGlow ensures the independence of different conditions and outperforms regular Glow to a significant extent.

1. Introduction

Generative adversarial networks (GANs) [1, 11, 27, 32] and variational auto-encoders (VAEs) [18] are two types of the most popular generative models due to their solid theoretic foundation and excellent results. Also, the performance of conditional image synthesis by these models improves rapidly with the fast development of deep learning. However, GANs have no explicit encoder to map images into a latent space, which is useful for many downstream tasks while the generated images by VAEs tend to be blurry. These problems remain for conditional versions of these models [30, 31, 36, 37]. Recently flow-based generative models draw increasing attention due to its natural reversibility of mapping between image space and latent space, exact log-likelihood, and its great potential in image synthesis [7, 8, 12, 17]. In this work we focus on conditional image synthesis by flow-based generative model.



(a) Inferred (b) Sampled

Figure 1. Barnes-Hut t-SNE [40] visualization of $6,000$ latent vectors on 200 identities of CGlow [17]. (a) latent vectors inferred by forward CGlow; (b) randomly sampled latent vectors by inverse CGlow. Best viewed in color.

Unfortunately, conditional image synthesis is a challenging task for flow-based generative models, as these models are forced to have a bijective mapping between the distributions of images and latent vectors according to their definitions [7], which means that their latent dimension must match visible dimension [10]. So there is no way to concatenate conditional information with images into the intact model like CGAN [30], CVAE [36] and CVAE-GAN [2]. Another straight-forward idea is to add a discriminative regularization to the optimization objective like [5, 28] with a class dependent prior, as mentioned in the work of original Glow [17]. We name this incremental variant of Glow as *CGlow* in this paper. But it tends to fail when meeting complicated conditions, for example, a face dataset with 200 identities. As shown in Figure 1, the distribution of real latent vectors inferred by forward CGlow has very close clusters, but the clusters of sampled latent vectors keep far apart and have a large divergence from the real distribution, which leads to artifacts in its generated images, as shown in Figure 3. This phenomenon results from that the underlying distribution of image conditions is difficult to measure precisely on the latent space, not to mention some multi-target tasks such as Pose-Invariant Face Recognition [6, 39] and Identity-Attribute Disentanglement [9, 22]. This methods has no way to explore some unknown properties hidden on the latent space either [4].

To tackle the problems of flow-based generative models mentioned above, we propose a novel conditional flow-based generative model, named as conditional adversarial generative flow (CAGlow). Instead of disentangling representation on the latent space directly, which is a difficult task for flow-based models, this approach learns an effective encoder to map the distribution of conditions into a latent space and builds a tight connection between the real and generated distributions in an adversarial manner. The main contributions of this work are summarized as follows:

- We are the first to learn a mapping from conditions to images by using an irreversible encoder to map conditions into the latent space of reversible flow-based models, which can make use of its reversibility to perform controllable image synthesis.
- We also incorporate adversarial networks into the proposed CAGlow, which helps the encoder learn a continuous mapping between condition space and latent space by adversarial training.
- By performing extensive experiments, we testify that CAGlow outperforms the state-of-the-art flow-based model Glow on complex conditions, and this approach can perform image synthesis conditioned on some unknown but interpretable representations learned in an unsupervised fashion.

2. A Review of Flow-based Generative Model and Conditional Image Synthesis

Before going deep into the proposed conditional adversarial generative flow, we take a short review of some state-of-the-art generative models and conditional image synthesis models from a probabilistic viewpoint, which acts as a basic theory of our work.

2.1. Three Basic Generative Models

There are three basic types of commonly used generative models: generative adversarial networks (GANs) [11], variational auto-encoders (VAEs) [18] and flow-based generative models (FGMs) [7, 8, 17]. GAN contains a discriminator and a generator model playing a minimax game. Such a two-player game is actually optimized when they reach the Nash-Equilibrium point, that is, the discriminator can not tell whether an image is real or not. Many following works improve generative adversarial networks by better loss, training skills and evaluating metrics [1, 13, 19, 34, 16, 24, 27, 35]. The objective of VAEs is to maximize the variational lower bound of log-likelihood of target data points. This lower bound is composed of a KL divergence, between the distribution modeled by the encoder and a prior distribution of latent vectors, and the reconstruction loss between the output and input data. Since there are different strengths and weaknesses in these two types of models, many works are proposed to take full advantages of both of them for promoting image synthesis [20, 26, 29].

Unlike the former two models, flow-based generative models build up a series of invertible transformations and directly optimize the negative log-likelihood of data distribution.

2.2. Flow-based Generative Models

As mentioned above, flow-based generative models [7, 8, 17] aim to map the distribution of natural image $p^*(x)$ into a latent prior distribution $p^*(z)$ using a bijective function F, that is $z = F(x)$. Because the function F is bijective, $x = F^{-1}(z)$ is valid as well and could be used to generate images. So its objective for maximizing log-likelihood can be formulated by change of variables:

$$\begin{aligned}\log p^*(x) &= \log p^*(z) + \log\left|\det \frac{dF}{dx}\right| \\ &= \log p^*(z) + \sum_{i=1}^{K} \log\left|\det \frac{dh_i}{dh_{i-1}}\right|,\end{aligned} \tag{1}$$

where we define $dh_0 = x$ and $dh_K = z$ for conciseness and the scalar $\log|\det dh_i/dh_{i-1}|$ is the absolute value of the log-determinant of the Jacobian matrix dh_i/dh_{i-1}. Such Jaconbian matrices lie on the design of bijective functions such as affine coupling layers and invertible 1×1 convolutions. Please refer to [8] and [17] for more details.

2.3. Conditional Image Synthesis

Mainstream conditional generative models consist of VAEs and GANs. Along the line of VAEs, CVAE [36] was proposed to extend the traditional VAE to a conditional generative model, which models a conditional distribution and finds the variational lower bound of this distribution following the idea of vanilla VAE. Along the other line of GANs, there exist more conditional models with different forms and applications [3, 14, 15, 33, 41, 42, 43]. To the best of our knowledge, the pioneer work is CGAN [30], which concatenates noise or images with class labels and then feeds them into the generator for conditional image synthesis. This idea is simple but lacks efficiency when dealing with multi-category classification tasks. Then ACGAN [31] was proposed to tackle such problems by simply presenting an auxiliary classifier for the discriminator. Another amazing work is infoGAN [4], which learns interpretable and disentangled representation in a totally unsupervised fashion and provides an elegant theory based on maximizing the mutual information between the input latent codes and their observations.

The Auxiliary Classifier Generative Adversarial Network (ACGAN) is a classical variant of vanilla GAN, whose objective functions are summarized by:

$$\begin{aligned} L_s &= \mathbb{E}_{x\sim p^*(x)}[\log D_\phi(x)] + \mathbb{E}_{x\sim p_\theta(x)}[\log(1 - D_\phi(x))], \\ L_c &= \mathbb{E}_{x\sim p^*(x),c\sim p(c)}[\log p_\phi(c|x)] \\ &\qquad + \mathbb{E}_{x\sim p_\theta(x),c\sim p(c)}[\log p_\phi(c|x)], \end{aligned} \tag{2}$$

where $p^*(x)$ denotes the real distribution of images x, $p_\theta(x)$ denotes the generated one, and the discriminator D_ϕ and the classifier $p_\phi(c|x) = C_\phi(x)$ share the parameters of two-in-one neural networks. The minimax game is optimized by training the generator to maximize $L_c - L_s$ and training the discriminator/classifier to maximize $L_c + L_s$.

As we know, the objective of GAN is actually to minimize the Jensen-Shannon Divergence between the real and fake distribution [1]. So the objective above can also be described as to maximize:

$$\begin{aligned} -JS(p^*(x)||p_\theta(x)) &+ \mathbb{E}_{x\sim p^*(x),c\sim p(c)}[\log p_\phi(c|x)] \\ &+ \mathbb{E}_{x\sim p_\theta(x),c\sim p(c)}[\log p_\phi(c|x)]. \end{aligned} \tag{3}$$

Furthermore, there are some models combining GANs with VAEs to boost the generation performance like [26, 29]. CVAE-GAN [2] is a conditional generative model unifying VAEs with GANs. It first encodes images with labels into latent vectors and then exploits the encoded vectors and same labels to generate images conditionally with the help of a real or fake discriminator and an auxiliary classifier. This model shows great potential in dealing with fine-grained classification problems.

Many empirical studies show the generated images from GANs are sharper than those of VAEs, for both unconditional and conditional models [3, 16]. However, unlike variational auto-encoders and flow-based generative models, classic GANs have no encoder to map natural images into latent space, which is useful for downstream tasks such as image editing, inpainting and attribute morphing. Furthermore, different from VAEs that optimize the lower bound of maximum likelihood and infer the latent variable approximately, the objective of flow-based generative model is to optimize the exact log-likelihood directly and infer the latent variable without sampling.

Therefore, in this paper we take full advantages of the latent space of flow-based models by building a continuous mapping from condition space to latent space and capture the targeted distribution precisely by adversarial networks.

3. Conditional Adversarial Generative Flow

In this section, we introduce the formulation and detailed architecture of our proposed model CAGlow, as shown in Figure 2. This model contains a reversible flow, an encoder and a supervision block in general.

3.1. Formulation

First, inspired by Eq.(1), we model an image with its conditions as a joint probabilistic distribution and go one step further to obtain the distribution of latent vectors with conditions by a bijective mapping $z = F(x)$:

$$\log p(x, c_s) = \log p(z, c_s) + \log\left|\det\frac{dF}{dx}\right|, \tag{4}$$

where we let c_s denote the conditions under supervision.

Using Bayesian formula, maximizing equation 4 is equal to:

$$\begin{aligned} \max \quad & \mathbb{E}_{z\sim p^*(z),c_s\sim p(c_s)}[\log p(c_s|z)] \\ & + \mathbb{E}_{z\sim p^*(z)}[\log p^*(z) + \log\left|\det\frac{dF}{dx}\right|], \end{aligned} \tag{5}$$

where the prior $p^*(z)$ is modeled by a standard Gaussian distribution.

We assume there is an unknown other distribution $p(z)$ for latent vectors. According to Gibb's inequality [25], we find a lower bound for $p^*(z)$:

$$\begin{aligned} & \mathbb{E}_{z\sim p^*(z)}[\log p^*(z)] \geq \mathbb{E}_{z\sim p^*(z)}[\log p(z)] \\ = & \mathbb{E}_{z\sim p^*(z)}[\log p^*(z)] - KL(p^*(z)||p(z)). \end{aligned} \tag{6}$$

Second, we model all the conditions as $p(\tilde{c}) = p(c_s, c_u)$ where c_s denotes the supervised conditions and c_u denotes the unsupervised ones. Thus we could use an encoder E to map the conditional information with random noises into a latent distribution $p_\theta(z) = E_\theta(\tilde{c}, \epsilon)$ where ϵ denotes random noise.

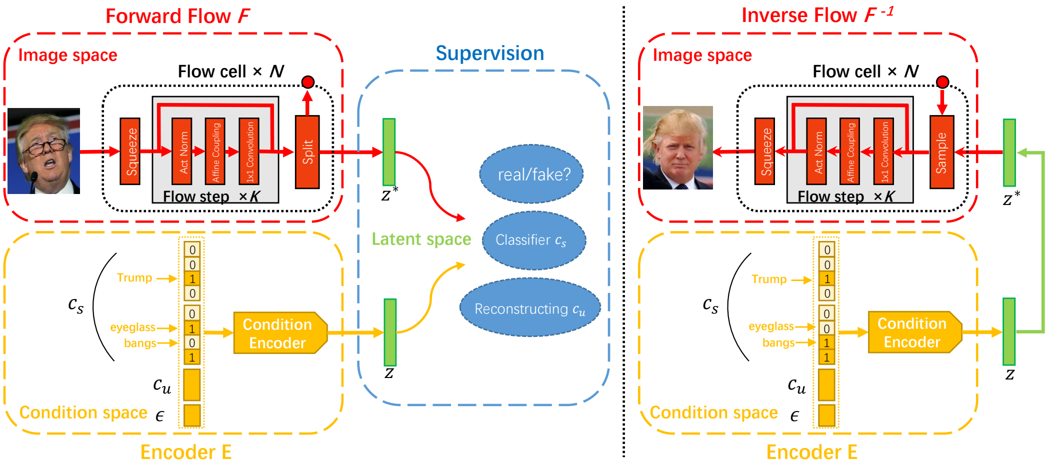
Using the variational lower bound methods from VAEs [18], we can find a lower bound for $p(\tilde{c})$ by

$$\log p(\tilde{c}) \geq -KL(p_\theta(z)||p(z)) + \mathbb{E}_{z\sim p_\theta(z),\tilde{c}\sim p(\tilde{c})}[\log p(\tilde{c}|z)]. \tag{7}$$

Here we define $p(z) = (p_\theta(z) + p^*(z))/2$, so we have $KL(p_\theta(z)||p(z)) + KL(p^*(z)||p(z)) = JS(p_\theta(z)||p^*(z))$. Also, we propose a classifier C to classify z from both real and fake distributions. At last, by bringing together all the Eq.(4-7), we obtain our final objective to maximize:

$$\begin{aligned} & \mathbb{E}_{z\sim p^*(z)}[\log p^*(z) + \log\left|\det\frac{dF}{dx}\right|] \\ & -JS(p_\theta(z)||p^*(z)) + \mathbb{E}_{z\sim p^*(z),c_s\sim p(c_s)}[\log p(c_s|z)] \\ & +\mathbb{E}_{z\sim p_\theta(z),\tilde{c}\sim p(\tilde{c})}[\log p(\tilde{c}|z)]. \end{aligned} \tag{8}$$

This objective function could be decomposed into two parts: the first term is the same as the objective of the reversible flow Eq.(1), and the last three terms are very similar to the objective of ACGAN Eq.(3). The difference is



(a) The training procedure of CAGlow

(b) The image synthesis process of CAGlow

Figure 2. Illustration of the network architecture of the proposed conditional adversarial generative flow. It contains a reversible flow F, an encoder E, and a supervision block including a discriminator D_i distinguishing real vectors from fake ones, a classifier C classifying supervised conditions correctly and a decoder D_e reconstructing unsupervised conditions.

that $p(\tilde{c})$ contains both supervised and unsupervised conditional information. Here we assume that they are independent with each other and implement them with a classifier and a decoder, which are illustrated in following part.

3.2. Network Structure

Considering our target is to maximize Eq.(8), we would introduce the proposed network structure carefully for achieving this goal. As can be seen in Figure 2, the proposed model contains three parts: 1) a reversible multi-scale flow F; 2) an encoder E_θ; and 3) a supervision block which is a three-in-one neural network including a discriminator $D_{i\phi}$, a classifier C_ϕ and a decoder $D_{e\phi}$.

Reversible flow F builds a bijective mapping between the distributions of natural images and latent vectors using reversible networks formulated as $z = F(x)$ where z has a prior distribution $p^*(z)$. Here we take a standard Gaussian distribution for modeling z and optimize it using maximum likelihood estimation. Specifically, we take the structure of Glow $N \times K$ as our baseline as shown in Figure 2. So the loss for reversible flow is

$$\mathcal{L}_F = -\mathbb{E}_{z \sim p^*(z)}[\log p^*(z) + \log \left| \det \frac{dF}{dx} \right|]. \tag{9}$$

Note that samples from $p^*(z)$ are taken as the real data which are fed into the supervision block for further adversarial training, so we take multi-stage training strategy and the first stage is to train a regular Glow model for the following efficient sampling of the latent vectors. An extra advantage of this strategy is that after training, the pretrained Glow model could be used for different tasks by adding different supervised signals on the small supervision blocks, getting rid of the large computation consumption for training many different conditional Glow models on different tasks.

Encoder E_θ helps to model the conditional distribution of latent vectors z on conditions $\tilde{c}$. That is, $p_\theta(z) = E_\theta(\tilde{c}, \epsilon)$ where ϵ is from an underlying distribution $p(\epsilon)$ modeled by a standard Gaussian distribution to help E generate diverse samples of latent vectors. $p(\tilde{c})$ is actually modelling the joint distribution for both supervised conditions c_s and unsupervised conditions c_u. Take Figure 2 as an example, when a face image is fed into the forward flow F, its supervised conditions c_s containing identity number and attributes like eyeglasses and bangs are fed into the encoder E as one-hot vectors, and simultaneously offer a supervised signal from the top of the classifier C. Meanwhile, an unsupervised condition c_u and a random noise ϵ are sampled from their specific distribution and concatenated with the supervised conditions. c_u will be decoded from the latent vectors by a decoder D_e to enhance its mutual information with z. According to the objective Eq.(8), We would like to minimize the JS Divergence between this conditional distribution $p_\theta(z)$ and the distribution of real latent vectors $p^*(z)$ inferred by the forward flow, with the help of discriminator $D_{i\phi}$. So the loss for the encoder E_θ is:

$$\mathcal{L}_E = -\mathbb{E}_{\epsilon \sim p(\epsilon), \tilde{c} \sim p(\tilde{c})}[\log D_{i\phi}(E_\theta(\tilde{c}, \epsilon))]. \tag{10}$$

Discriminator $D_{i\phi}$ aims to distinguish generated latent vectors from real ones inferred by reversible flow correspondingly:

$$\mathcal{L}_{D_i} = -\mathbb{E}_{z \sim p^*(z)}[\log D_{i\phi}(z)] - \mathbb{E}_{z \sim p_\theta(z)}[1 - \log D_{i\phi}(z)]. \tag{11}$$

Classifier C_ϕ partly shares the parameters with the discriminator D_ϕ and outputs different class probabilities by softmax or sigmoid functions. We supervise its training by a cross entropy loss or binary cross entropy loss for different specific tasks. By such a neural network parameterized classifier, we can obtain a class posterior probabilities $q_\phi(c_s|z)$ of both labeled real vectors and generated ones. The loss could be formulated as:

$$\begin{aligned} \mathcal{L}_C = &- \mathbb{E}_{z\sim p^*(z), c_s\sim p(c_s)}[\log q_\phi(c_s|z)] \\ &- \mathbb{E}_{z\sim p_\theta(z), c_s\sim p(c_s)}[\log q_\phi(c_s|z)]. \end{aligned} \tag{12}$$

Decoder $D_{e\phi}$ partly shares the network parameters with the discriminator and classifier, and it aims to decode the unsupervised conditions from the generated latent vectors for reconstructing them. So the loss for the decoder is:

$$\mathcal{L}_{D_e} = -\mathbb{E}_{z\sim p_\theta(z), c_u\sim p(c_u)}[\log q_\phi(c_u|z)], \tag{13}$$

where $p(c_u)$ could be modeled by uniform distribution for continuous codes and binomial distribution for discrete codes. Correspondingly, the loss could be set to mean square error and binary cross entropy loss.

3.3. Objective of CAGlow

We show the designed networks for maximizing equation 8, but in practice, the distribution of real latent vectors and generated ones may not overlap with each other, especially during the early stage of training process, and thus the discriminator can separate them accurately. This phenomenon makes the training process unstable and easy to mode collapse. To overcome this typical but important problem, we propose a pair-wise feature matching regularization strategy, which uses an $L2$ loss between the representation of real and fake data points with same conditions. Let $f(z)$ denote the features of the latent vectors z on the intermediate layer of the network of supervision block, so this pairwise feature matching loss is formulated as:

$$\mathcal{L}_{FM} = \frac{1}{2}||f(z) - f(z')||_2^2. \tag{14}$$

So the final goal of our proposed CAGlow is to minimize the loss:

$$\mathcal{L} = \sum_{S\in\{F,E,D_i,C,D_e,FM\}} (\lambda_S \mathcal{L}_S), \tag{15}$$

where the exact loss functions are presented in Eq.(9-14). Note that the discriminator $D_{i\phi}$, the classifier C_ϕ and the decoder $D_{e\phi}$ share most of parameters in the supervision networks except for their output layers. $\mathcal{L}_{D_i}$ measures how well the discriminator separates the real and fake vectors and $\mathcal{L}_C$ measures how good the classifier at classifying different categories, which can be used directly in downstream tasks like semi-supervised learning. $\mathcal{L}_{D_e}$ measures how well the decoder reconstructing the input unsupervised codes, which could be used for unknown properties exploration.

4. Experiments

In this section, we would empirically demonstrate the advantage of our proposed approach over some leading baselines to a significant extent.

4.1. Implementation Details

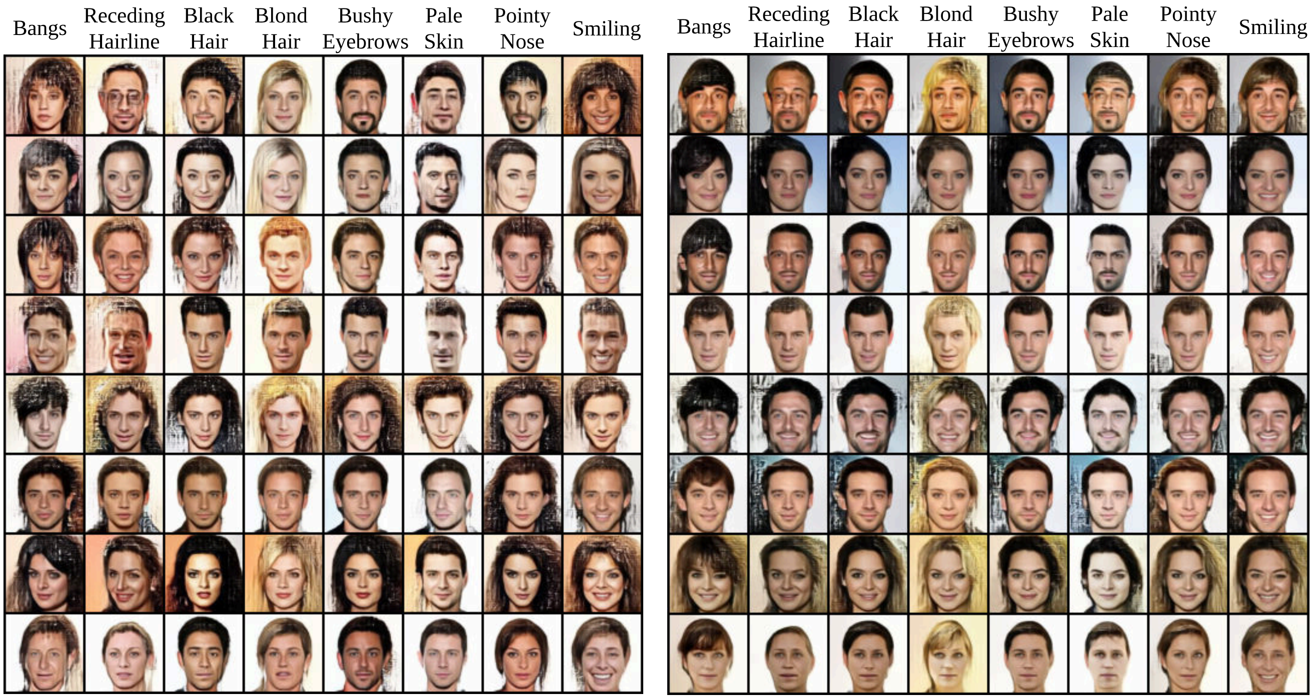
Datasets. We validate the effectiveness of our proposed model on some publicly accessible datasets. The first dataset is MNIST digits dataset [21] containing $50,000$ training data and $10,000$ test data with classes from number 0 to 9. The second one is the large-scale face dataset CelebA [23] which contains $202,599$ number of face images with $10,177$ number of identities and 40 binary attributes annotations per image. For CelebA dataset, we choose a relatively small image size 64 for evaluation due to the large computation consumption of Glow [17]. But the notion is the same for larger image size.

Networks. In our experiment, we set the reversible flow network to be a typical setting of Glow $N \times K$. N is the number of cells which contains a *Squeeze* and a *Split* operation for downsampling and dimension reduction. K is the number of steps which contains an affine coupling layer and an invertible 1×1 convolution. Please refer to [8, 17] for details. We set Glow 3×10 for MNIST and 3×32 for CelebA. In the experiments of MNIST, the encoder and supervision block contain a two fully-connected layers with 64 hidden neurons. The discriminator, classifier and decoder only share the first layer and output different vectors for calculating their own losses. In the experiments of CelebA, the encoder first embeds identities into a fixed-dimensional latent vector and concatenate it with one-hot vectors of attributes and random noise. Then the vectors pass through one fully-connected layer and three deconvolutional layers with upsampling scale 2, 2, 1 and channnel size 128, 512, 48 respectively. The supervision block contains two stride 2 convolutional layers with channel size 64, 128, followed by four specific fully-connected layers for outputting the probabilities for real or fake, different identities, different attributes and reconstructing unsupervised conditions.

Baselines. Since the proposed model is an extension from the state-of-the-art flow-based generative model Glow, we mainly testify the superiority of the proposed model CAGlow to the prior work Glow and its incremental variant CGlow [17].

4.2. Controllable Image Synthesis

Conditional images synthesis results on different identities and attributes by different approaches are demonstrated in Figure 3. We set same identity for each row and same attribute for each column. From Figure 3a, we could see that the generated images by CGlow are disturbed severely by



(a) CGlow [17]

(b) CAGlow

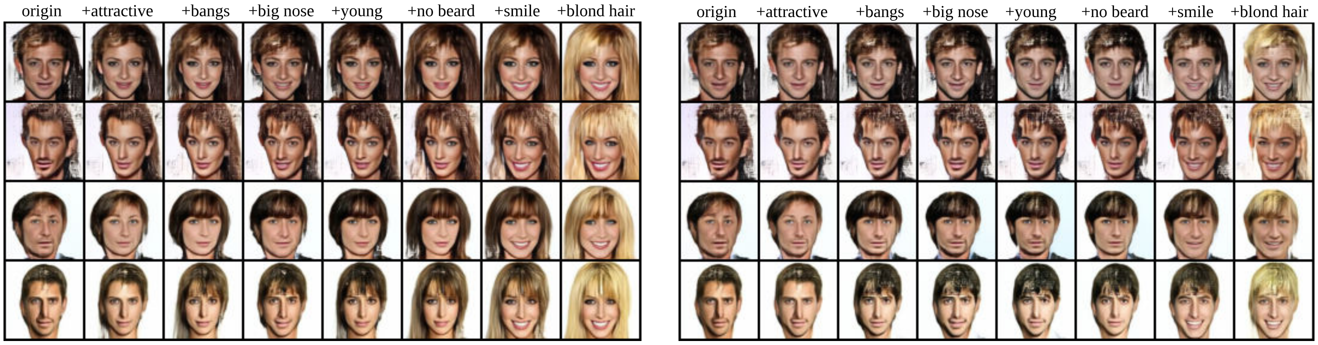
Figure 3. Conditional image synthesis demonstration. From top to bottom: different people. From left to right: different attributes (specific attribute is annotated above the first row). (a) Images generated by CGlow. Identities and attributes interfere with each other heavily; (b) Images generated by CAGlow with better controllability.

different identities and attributes. The change of attributes has an adverse impact on the identities and vice versa. In addition, the change of attributes also influences the appearance or disappearance of other attributes in CGlow. Besides, we could see some artifacts in the images generated by CGlow in that the sampling distribution deviates from the real one, as mentioned in Section 1. While the images synthesized by CAGlow avoid such negative effects and show excellent performance under this setting, as shown in Figure 3b.

Image synthesis under cumulative conditions. To further validate the controllability of our approach, we demonstrate the generation results of changing multiple attributes step by step. Because it is difficult for CGlow to change multiple attributes with identities persistent, we compare our approach with the regular Glow with pre-storing features. To maintain identities while changing attributes, the regular Glow must first parse all the latent vectors of original images and store a mean feature for each specific attribute. Then it infers the latent vector of an arbitrary image and changes this vector by adding pre-stored attribute feature, and thus it could generate targeted images with identity unchanged. This strategy works well when manipulating just one attribute. But the results are not ideal enough when manipulating multiple attributes. As shown in Figure 4, we add one more attribute to the original face images step by step. In regular Glow, adding the attributes ‘young’ and ‘no beard’ causes the appearance of the attributes ‘makeup’ and ‘long hair’ and adding the attribute ‘blond hair’ even results in the change of identity. In contrast, our model performs well by controlling the change of attributes independently under cumulative conditions.

Besides, our approach has two extra advantages over the regular Glow: 1) By a condition-latent-condition encoding-decoding strategy, we disentangle the feature of identities and attributes well, so it could produce non-interfered images; 2) We only feed some one-hot vectors of conditions into the encoder followed by the inverse flow to generate images, which does not need pre-storing attribute features and inference process for obtaining a specific latent vector, so CAGlow has a obvious improvement on time and space consumption.

Smooth Interpolation. We also demonstrate an interpolation generation results on two different identities and attributes simultaneously in Figure 5. The operation is completed by simply changing the input one-hot vector of two specific targets from $[0, 1]$ to $[1, 0]$. As one can see from the figure, the interpolation of one specific condition demonstrates continuous changes of generated images and has no negative impact on another condition.



(a) Glow [17]

(b) CAGlow

Figure 4. Image synthesis under cumulative conditions demonstration. From left to right: adding different attributes step by step (specific attribute is annotated above the first row). (a) Images generated by regular Glow with pre-storing features. Identities and other attributes are interfered heavily; (b) Images generated by CAGlow with better controllability.

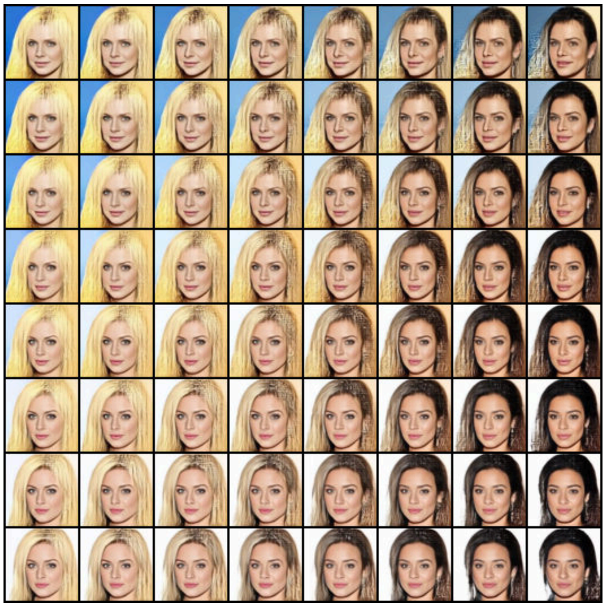


Figure 5. Interpolation both on ID and attribute. From top to bottom: interpolation on two different people. From left to right: interpolation on two different attributes (blond hair to black hair).

	Glow	CGlow	CAGlow
Acc (MNIST)	-	98.89%	99.55%
Acc (CelebA)	-	87.43%	95.16%
FID (MNIST)	25.78	29.64	26.34
FID (CelebA)	103.67	126.52	104.91

Table 1. Accuracy and FID results on MNIST and CelebA.

Model	CGlow	CAGlow
Accuracy	87.36%	93.75%
Variance	0.0245	0.0016

Table 2. Accuracy of Attributes and Variance of AMP with different identities on CelebA.

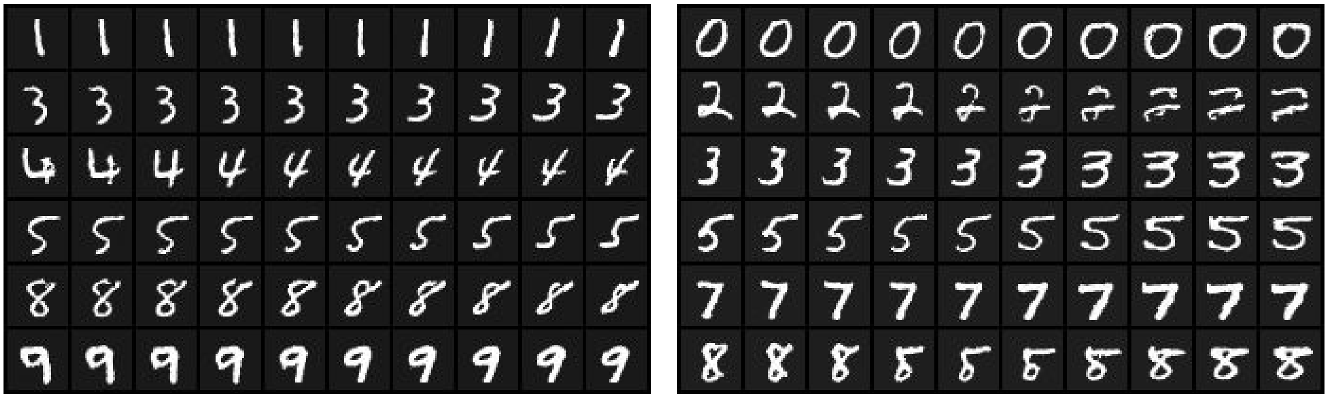
4.3. Quantitative Comparisons

In this part we perform some experiments to verify the superiority of our approach using some quantitative results.

Category preserving test. We would take Fréchet Inception Distance (FID) [24] and top-1 accuracy as our metric, to evaluate the realism, diversity and discriminability respectively. We pretrain the GoogLeNet [38] on MNIST and CelebA dataset and then calculate the top-1 accuracy of the generated samples by different approaches. Following the method in [13], we calculate the FID score of the generated samples on a pretrained GoogLeNet. As shown in Table 1, our approach achieves better performance on both MNIST and CelebA dataset. The FID score of CAGlow is pretty close to the original Glow, which means that our approach could learn a good conditional distribution of latent vectors without losing diversity.

Attribute preserving test. Here we propose a novel evaluation metric Attribute Mean Probability (AMP) for testing the stability of the attributes. We first train L different classifiers for L different attributes based on CelebA dataset to obtain above 99% precision. For any image x_i with identity i, these classifiers could output the probabilities $p_l(x_i)$ for different attributes $l \in \{1, ..., L\}$. The value of AMP is calculated by $AMP_i = \frac{1}{L}\sum_{l=1}^{L} p_l(x_i)$. Based on these classifiers, we could calculate the mean value of predicted accuracy for all generated samples and the variance of AMP along the identities. Our approach has better accuracy and lower variance, as reported in Table 2.

Cumulative conditions interfering test. Same as the operation in 4.2, we change multiple attributes step by step. Based on L pretrained classifiers mentioned above, we calculate the last-step and this-step mean probability of $L-1$ attributes except for the changed one for the generated images. Then we take the absolute value of the difference be-



(a) Varying latent code for rotation.

(b) Varying latent code for width.

Figure 6. Unknown properties exploration on MNIST [21].

# Mpl	CGlow	Glow + pre-store	CAGlow
1	0.004350	0.002245	0.000774
2	0.023215	0.007213	0.002608
3	0.047055	0.014352	0.005825
4	0.077767	0.023767	0.009939

Table 3. The absolute value of the difference of AMP *w.r.t.* the times of manipulation. Lower value means more stability.

tween the AMP of last step and this step as the metric. This evaluating metric describes the disturbing extent to the result. Smaller value means a more stable generating system and illustrates a better disentanglement between different attributes. We show that our results achieve the best performance, as summarized in Table 3.

4.4. Interpretable Properties Exploration with Unsupervised Learning

In the part we will explore some underlying properties in MNIST and CelebA dataset. Besides the conditional information these datasets provide, there are some unknown conditional information hidden. This experiment aims to demonstrate that our model could generate images conditioned on some unknown but interpretable properties. Note that these properties are found in an unsupervised manner. We do not add any supervision signals on the loss and only use an auto-encoding reconstruction loss for the input codes sampled from a prior distribution. We assume uniform distribution for unsupervised codes and take a mean square error loss for reconstruction.

The results on MNIST are shown in Figure 6. From this figure, we can see that the rotation direction and width of the generated digits changed continuously with the varying of the latent conditional codes respectively.

We also show the exploration results on CelebA in Figure 7. As we can see, our approach could capture the underlying distribution for different yaw angles and brightness, which are not annotated in CelebA dataset.

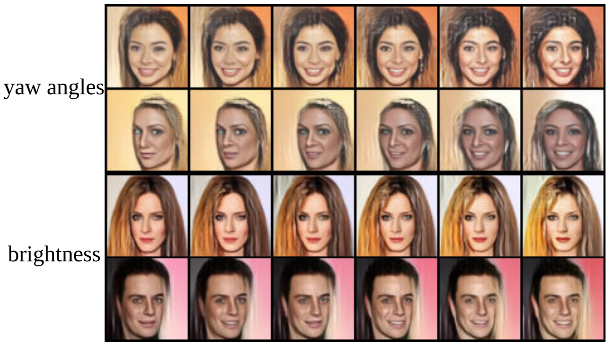


Figure 7. Unknown properties exploration on CelebA.

5. Conclusion

In this paper we proposed a novel generative model CAGlow that seamlessly unifies three sub-blocks: a reversible flow, an encoder and a supervision block and takes advantage of an adversarial training strategy. This framework provides great controllability and flexibility for synthesizing images conditioned on multiple annotations. Both qualitative and quantitative experimental results testified the superiority of the proposed approach to the vanilla version of Glow. In the future we plan to further investigate the impact of a more complex prior distribution instead of a simple Gaussian distribution on flow-based generative models.

Acknowledgement. This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14202217, CUHK14203118, CUHK14205615, CUHK14207814, CUHK14213616, CUHK14208417, CUHK14239816, and in part by CUHK Direct Grant.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 1, 2, 3

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. CVAE-GAN: fine-grained image generation through asymmetric training. In *ICCV*, 2017. 1, 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *ICLR*, 2019. 2, 3

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016. 2

[5] Brian Cheung, Jesse A. Livezey, Arjun K. Bansal, and Bruno A. Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014. 1

[6] Changxing Ding and Dacheng Tao. A comprehensive survey on pose-invariant face recognition. *ACM Trans. Intell. Syst. Technol.*, pages 37:1–37:42, 2016. 2

[7] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *ICLR workshops*, 2015. 1, 2

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017. 1, 2, 5

[9] Yixiao Ge, Zhuowan Li, Haiyu Zhao, Guojun Yin, Shuai Yi, Xiaogang Wang, and Hongsheng Li. FD-GAN: Pose-guided Feature Distilling GAN for Robust Person Re-identification. In *NeurIPS*, 2018. 2

[10] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2

[12] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. *arXiv preprint arXiv:1810.01367*, 2018. 1

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 2, 7

[14] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *CVPR*, 2018. 2

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 2, 3

[17] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 1, 2, 5, 6, 7

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 1, 2, 3

[19] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018. 2

[20] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 2

[21] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998. 5, 8

[22] Yu Liu, Fangyin Wei, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Exploring disentangled feature representation beyond face identification. In *CVPR*, 2018. 2

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 5

[24] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *NeurIPS*, 2018. 2, 7

[25] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. 3

[26] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 2, 3

[27] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint arXiv:1611.04076*, 2016. 1, 2

[28] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *NeurIPS*, 2016. 1

[29] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *ICML*, 2017. 2, 3

[30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1, 2

[31] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017. 1, 2

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1

[33] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 2

[34] Tim Sainburg, Marvin Thielk, Brad Theilman, Benjamin Migliori, and Timothy Gentner. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. *arXiv preprint arXiv:1807.06650*, 2018. 2

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 2

[36] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NeurIPS*, 2015. 1, 2

[37] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR*, 2016. 1

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[39] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017. 2

[40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, pages 2579–2605, 2008. 1

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2

[42] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2