This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# SphericGAN: Semi-supervised Hyper-spherical Generative Adversarial Networks for Fine-grained Image Synthesis

Tianyi Chen[1], Yunfei Zhang[1], Xiaoyang Huo[1], Si Wu[1,2*], Yong Xu[1,2,3], and Hau San Wong[4]
[1]School of Computer Science and Engineering, South China University of Technology
[2]Peng Cheng Laboratory
[3]Communication and Computer Network Laboratory of Guangdong
[4]Department of Computer Science, City University of Hong Kong
{csttychen, cszhangyunfei, csxyhuo}@mail.scut.edu.cn, {cswusi, yxu}@scut.edu.cn,
cshswong@cityu.edu.hk

## Abstract

*Generative Adversarial Network (GAN)-based models have greatly facilitated image synthesis. However, the model performance may be degraded when applied to fine-grained data, due to limited training samples and subtle distinction among categories. Different from generic GANs, we address the issue from a new perspective of discovering and utilizing the underlying structure of real data to explicitly regularize the spatial organization of latent space. To reduce the dependence of generative models on labeled data, we propose a semi-supervised hyper-spherical GAN for class-conditional fine-grained image generation, and our model is referred to as SphericGAN. By projecting random vectors drawn from a prior distribution onto a hyper-sphere, we can model more complex distributions, while at the same time the similarity between the resulting latent vectors depends only on the angle, but not on their magnitudes. On the other hand, we also incorporate a mapping network to map real images onto the hyper-sphere, and match latent vectors with the underlying structure of real data via real-fake cluster alignment. As a result, we obtain a spatially organized latent space, which is useful for capturing class-independent variation factors. The experimental results suggest that our SphericGAN achieves state-of-the-art performance in synthesizing high-fidelity images with precise class semantics.*

## 1. Introduction

Generative learning aims to model complex real-world data distributions, such that high-fidelity data can be synthesized from random vectors drawn from a prior distribution.

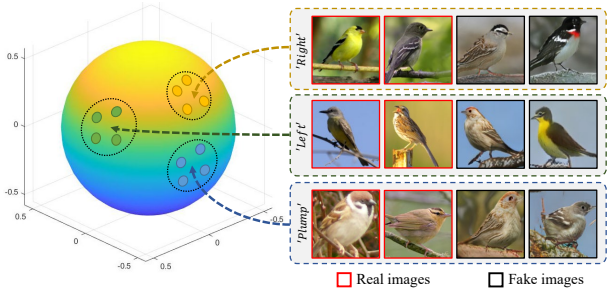



Figure 1. Illustration of our proposed hyper-spherical latent space. By match with the prior clusters of real data, the distribution of latent vectors can be complex, which is useful for capturing the class-independent variation factors.

The existing generative methods can be roughly divided into three groups: autoregressive models [31, 36], Variational Auto-Encoders (VAE) [21] and Generative Adversarial Networks (GANs) [14, 41]. In particular, GAN-based methods have achieved impressive performance in synthesizing high-quality images [9, 10, 17–19, 33]. A variety of conditional GAN architectures have been developed to regularize the data generation process, such as controlling class semantics of synthesized images [5, 6]. To reduce the dependence of class-conditional GANs on labeled training data, semi-supervised generative learning focuses on how to incorporate unlabeled data in the adversarial training process [4, 26, 27, 39]. However, most of the current GAN-based methods are unable to accommodate the fine-grained data scenarios, since generic real-fake data distribution matching is inadequate to capture the subtle distinctions among fine-grained classes.

Enlarging the latent space is an effective way to improve a GAN in capturing the factors of variation in fine-grained data. FineGAN [34] and MixNMatch [25] adopt hierarchi-

*Corresponding author.

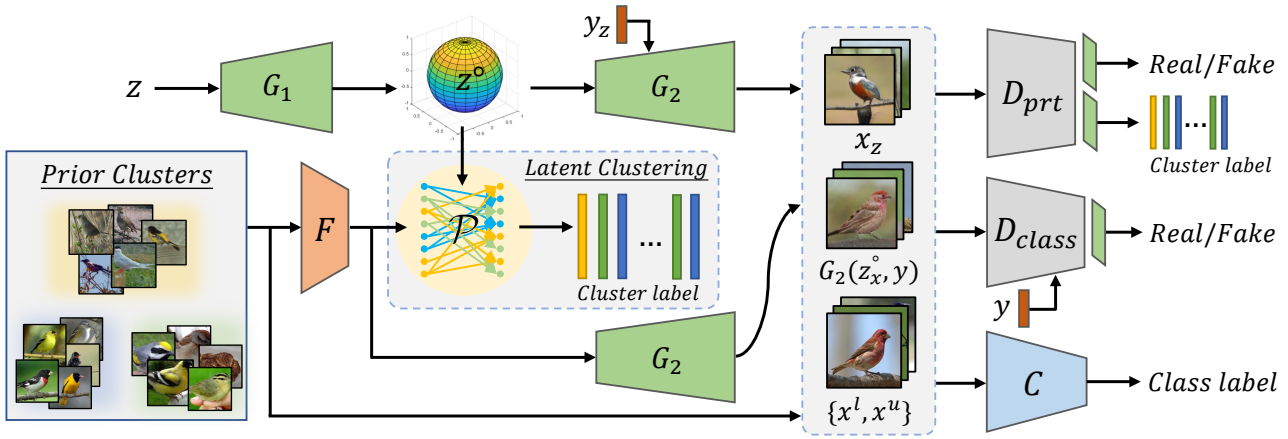



Figure 2. An overview of SphericGAN. The generator $G$ is divided into two parts $\{G_1, G_2\}$. $G_1$ maps a random vector $z$ onto a hyper-sphere, and $G_2$ synthesizes a class-specific image $x_z$ from the resulting latent vector $z^\circ = G_1(z)$ and class label $y_z$. A mapping network $F$ also maps real image $x \in \{x^l, x^u\}$ in the hyper-spherical latent space, and a latent discriminator is incorporated to facilitate the distribution matching between $z^\circ$ and $z_x^\circ = F(x)$ via adversarial training. Further, a set of latent prototypes $\mathcal{P}$ are learnt, and the corresponding latent clusters are aligned with the prior clusters of real data. The unlabeled data can be well utilized in this unsupervised learning process. On the other hand, the classifier $C$ is used to assign pseudo labels to unlabeled data. By competing with a prototypical discriminator $D_{prt}$ and a class conditional discriminator $D_{class}$, the generator is induced to capture class-independent and class-related variation factors.

cal generator architectures to individually synthesize background, mask (defines object shape) and object appearance, which are associated with different latent codes. Due to the limited expressiveness of the latent codes that are sampled from a prior distribution, StyleGAN [18, 19] adopts a nonlinear mapping network to map a random vector to a higher dimensional latent space $\mathcal{W}$, and the resulting vector is injected into different blocks of a generator to control the data generation process. In [1], an extended latent space $\mathcal{W}^+$ is used to further enhance the generalization capability of StyleGAN. Different from the above unconditional GANs, we adopt an unsupervised strategy to learn a hyper-spherical latent space, which facilitates the learning of class-independent variation factors as shown in Figure 1.

More specifically, we propose a semi-supervised GAN with a hyper-spherical latent space for class-conditional generative learning on fine-grained data, and our model is referred to as SphericGAN. Compared to the typical generator-classifier-discriminator model of Triple-GAN [24], there are two additional components in SphericGAN: a mapping network and a prototypical discriminator as shown in Figure 2. In contrast to the existing semi-supervised GANs that sample latent codes from a pre-defined distribution, such as Gaussian, we learn a hyper-sphere from the pre-defined space. In our hyper-spherical latent space, the statistics of resulting latent vectors can be complex, which is useful for capturing the class-independent variation factors in real data. Toward this end, we divide the real instances into a set of groups via clustering. After adopting the mapping network to map real images on the sphere, a set of prototypes are learnt to associated with the prior clusters. By pushing latent vectors toward the nearest prototypes, the underlying structure of real data can be captured. On the other hand, to align the clusters of real and fake data in semantics, the prototypical discriminator is incorporated to maximize the mutual information between the synthesized images and the ground-truth cluster code. Extensive experiments are conducted to verify the effectiveness of our hyper-spherical latent space in improving the class-conditional synthesis quality of fine-grained data.

We summarize the contributions of this work as follows: (1) We explore semi-supervised class-conditional fine-grained image generation from a new perspective: learning a hyper-spherical latent space to enhance the capability of a generator in capturing class-independent variation factors. (2) By incorporating a mapping network to map real data in the hyper-spherical latent space, the underlying data structure is learnt by aligning the latent vectors with the prior clusters of real instances. (3) A prototypical discriminator is designed and incorporated in the adversarial training process to further match the clusters of real and synthesized instances in semantics. (4) We judiciously design the optimization formulation of all the constituent networks to achieve superior performance over previous state-of-the-arts on multiple standard benchmarks.

## 2. Related Work

### 2.1. Semi-Supervised GANs

Different from generic class-conditional GANs that focus on supervised generative learning on sufficient labeled training data, semi-supervised GANs [11, 22] are developed

for the case where training data are only partially labeled. The amount of unlabeled data is typically much larger than that of labeled data. To leverage unlabeled data, a number of methods extended a discriminator for class label prediction. Springenberg proposed a Categorical GAN (CatGAN) [35], in which a discriminator played two roles: distinguishing real samples from fake ones, while at the same time predicting the class labels of real samples. Based on CatGAN, Salimans et al. [32] explored a variety of GAN training techniques to stabilize the optimization process, which leads to better synthesis quality. To improve the distribution matching between real data and synthesized data, Wei et al [38] adopted the Wasserstein distance [2], and imposed a consistency regularization on the discriminator to ensure the property of local Lipschitz continuity. Due to the fact that the two roles of the discriminator may conflict with each other to a certain extent, another strategy is to incorporate an additional classifier in the adversarial training process. Li et al. [24] proposed a Triple GAN (TripleGAN) to jointly optimize a generator, a classifier and a discriminator. The classifier aims to deceive the discriminator by inferring the class labels of unlabeled data to as accurately an extent as possible. Further, Gan et al. [13] proposed a triangle GAN ($\Delta$-GAN) to incorporate an additional discriminator to identify unlabeled data and synthesized data. Wu et al. [39] enhanced TripleGAN by performing feature-semantics matching. Liu et al. [26,27] utilized the operation of random regional replacement to construct difficult real-fake instances to enhance the capability of both discriminator and classifier. To improve the downstream classification task, Dong et al. [12] modified TripleGAN by encouraging the generator to synthesize difficult samples, such that decision boundaries were regularized in low-density regions.

### 2.2. Fine-Grained Generative Models

Compared to generic image generation, synthesizing fine-grained images is more challenging due to subtle inter-class distinctions, and has not been extensively explored so far [4]. The scale of fine-grained datasets is typically small, due to the reason that the cost of data collection is high and extensive expertise is required to annotate the data. For training a generator in this case, Bao et al. [3] combined VAE and GAN to stabilize model training. To capture the factors of variation, Yang et al. [40] proposed a Layered Recursive GAN (LR-GAN) to generate the parts of background and foreground independently. To control the semantics of synthesized images, Chen et al. [8] developed an Information maximizing GAN (InfoGAN) to associate latent codes with semantic attributes in an unsupervised manner. Based on InfoGAN, Singh et al. [34] designed a hierarchical generator architecture to disentangle background, object shape and appearance, and the resulting model was referred to as FineGAN. Li et al. [25] further extended FineGAN for encoding the attributes of reference images and transferring them to synthesized images. To simplify the model design and training procedure, Chen et al. [7] proposed a Single-Stage Controllable GAN (SSC-GAN) to associate the variation factors with different latent codes.

We focus on class-conditional fine-grained image synthesis in semi-supervised scenarios. There are fundamental differences between our proposed SphericGAN and the above works. To improve the expressiveness of latent codes, StyleGAN adopts a nonlinear mapping to obtain a high-dimensional latent space, which is a black box during training, while we learn a hyper-spherical latent space, and explicitly organize latent vectors by matching with the real data structure. By aligning with the prior clusters of real data, the latent clusters are associated with well-defined semantics. This is the first attempt to discover class-independent variation factors in class-conditional data synthesis.

## 3. Methodology

In semi-supervised scenarios, the training data is composed of a small amount of labeled data $X^{\mathbb{L}} = \{(x^l, y^l)\}$ and a large amount of unlabeled data $X^{\mathbb{U}} = \{x^u\}$, i.e. $|X^{\mathbb{L}}| \ll |X^{\mathbb{U}}|$. The proposed SphericGAN consists of a classifier $C$, a generator $G$, three discriminators $\{D_{lat}, D_{prt}, D_{class}\}$ and a mapping network $F$. $C$ learns to assign pseudo label $\tilde{y}^u = \texttt{one-hot}(C(x^u))$ to $x^u$ as accurately as possible. $G$ is divided into two parts $\{G_1, G_2\}$: $G_1$ maps a random vector $z$ to a hyper-spherical latent space, and $G_2$ synthesizes images $x_z$ from the latent code $z^\circ = G_1(z)$ together with a random class label $y_z$, i.e. $x_z = G_2(z^\circ, y_z)$. $F$ maps real images in the hyper-spherical latent space, and $D_{lat}$ distinguishes the resulting latent vectors $z_x^\circ = F(x)$ from $z^\circ$, where $x \in X^{\mathbb{L}} \bigcup X^{\mathbb{U}}$. In addition, $D_{prt}$ and $D_{class}$ are trained to identify real and synthesized images unconditionally and conditioned on class label, respectively. We provide the design of the components and the optimization formulation in the following subsections.

### 3.1. Hyper-spherical Latent Space

#### 3.1.1 Latent Clusters

Latent space typically encodes rich semantic information. To model a complex distribution in our hyper-spherical latent space, we aim to learn $K$ latent prototypes $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$ in an unsupervised manner. For each latent vector, the nearest prototype is determined only by their angle, and the cosine function is thus used as the similarity measure. We compute the probability distribution $\varphi(z^\circ) = [\varphi(z^\circ)^{(1)}, \varphi(z^\circ)^{(2)}, \ldots, \varphi(z^\circ)^{(K)}]$ of $z^\circ$ belonging to latent clusters as follows:

$$\varphi(z^\circ)^{(i)} = \frac{\exp(\cos(z^\circ, \mathcal{P}_i))}{\sum_k \exp(\cos(z^\circ, \mathcal{P}_k))}. \quad (1)$$

To perform clustering over latent vectors, we formulate the corresponding training loss as follows:

$$L_{lat}^{clust} = \mathbb{E}_z[-\varphi(z^\circ)\log\varphi(z^\circ)] - \lambda\varphi_I \log\overline{\varphi}_z, \quad (2)$$

where $\lambda$ is a weighting factor, $\varphi_I$ denotes a uniform distribution, and $\overline{\varphi}_z$ represents the mean prediction of latent vectors. Minimizing $L_{lat}^{clust}$ encourages the prototypes to move to the high-density regions, while at the same time maintaining the balance of the clusters.

To ensure that the images synthesized from the latent vectors belonging to the same cluster hold similar semantics, we incorporate a prototypical discriminator $D_{prt}$, which consists of a representation learning backbone and two heads: One head infers cluster labels, and the other identifies real and synthesized instances. We define an evaluation loss over the cluster predictions as follows:

$$L_{lat}^{eval} = \mathbb{E}_z[-\texttt{one-hot}(\varphi(z^\circ))\log D_{prt}^{\mathcal{P}}(x_z)], \quad (3)$$

where $D_{prt}^{\mathcal{P}}(\cdot)$ denotes the predicted cluster probability distribution. The mutual information between cluster label and synthesized data is maximized by minimizing $L_{lat}^{eval}$.

#### 3.1.2 Aligning Real-fake Clusters

To discover the class-independent variation factors, we employ a deep clustering algorithm [16] to identify groups of similar real instances and obtain prior clusters. Let $\rho_x$ denote the prior cluster label of real instance $x$. To induce the generator to capture the real data structure, we adopt the mapping network $F$ to map real instances in the hyper-spherical latent space, and $z_x^\circ$ denotes the resulting latent vectors. A latent discriminator $D_{lat}$ is incorporated to identify the latent vectors produced by $F$ and $G_1$. On the other hand, $F$ and $G_1$ cooperate to deceive $D_{lat}$, and the adversarial training loss function is defined as follows:

$$L_{lat}^{adv} = \mathbb{E}_x[\log(1 - D_{lat}(z_x^\circ))] + \mathbb{E}_z[\log D_{lat}(z^\circ)], \quad (4)$$

where $D_{lat}(\cdot)$ denotes the predicted probability of a latent vector being generated by $G_1$. The competition with $D_{lat}$ enforces $F$ and $G_1$ to match the distributions of the two types of latent vectors.

It is non-trivial to induce $F$ to focus on the class-independent attributes. To address this issue, we impose a cycle consistency regularization on $F$, and the corresponding loss function is formulated as follows:

$$L_{lat}^{cons} = \mathbb{E}_x[\|z_x^\circ - F(G_2(z_x^\circ, y))\|_1], \quad (5)$$

where $y$ denotes the class label of $x$. We encourage $F$ to encode the class-independent information.

To match the underlying structure of real data, we align the latent clusters with the prior clusters of real data, and thus require the prototypical discriminator $D_{prt}$ to correctly predict $\rho_x$, conditioned on the instance synthesized from $z_x^\circ$. We evaluate the prediction results on both original real instances and synthesized instances as follows:

$$L_{lat}^{align} = \mathbb{E}_x[-\rho_x \log D_{prt}^{\mathcal{P}}(x) - \rho_x \log D_{prt}^{\mathcal{P}}(G_2(z_x^\circ, y))]. \quad (6)$$

As a result, we not only explicitly model the real data structure in the hyper-spherical latent space, but also associate the latent clusters with the semantics that are held by the prior clusters of real data.

### 3.2. Model Optimization

To induce $G$ to synthesize high-fidelity images, we incorporate the prototypical discriminator $D_{prt}$ and the class-conditional discriminator $D_{class}$. Both of them are trained to distinguish real instances from fake ones, and the adversarial training loss function is formulated as follows:

$$\begin{aligned} L_{data}^{adv} &= \mathbb{E}_x[\log D_{prt}^{idt}(x) + \log D_{class}(x, y)] \\ &+ \mathbb{E}_z[\log(1 - D_{prt}^{idt}(x_z)) + \log(1 - D_{class}(x_z, y_z))], \end{aligned} \quad (7)$$

where $D_{prt}^{idt}(\cdot)$ ($D_{class}(\cdot,\cdot)$) represents the predicted probability of an input being from real data (conditioned on class label). In the competition with $G$, $D_{class}$ plays an important role in matching class-conditional distributions of real and synthesized data. However, there are limited labeled data per class in our case. To utilize unlabeled data, $D_{prt}$ is incorporated for marginal distribution matching.

On the other hand, $D_{prt}$ and $D_{class}$ aim at real-fake instance identification, and thus overlook the class separability of synthesized data. Considering that the classifier $C$ is able to extract the most discriminative features to infer class label, our motivation is to employ $C$ to verify the synthesized images in terms of class semantics. In addition to the synthesized data, $C$ is also trained on real data, and the training loss function is defined as follows:

$$\begin{aligned} L_{data}^{eval} &= \mathbb{E}_{x^l}[-y^l \log C(x^l)] + \mathbb{E}_{x^u}[-C(x^u)\log C(x^u)] \\ &+ \mu\mathbb{E}_z[-y_z \log C(x_z)], \end{aligned} \quad (8)$$

where $C(\cdot)$ denotes the predicted class probability distribution, and $\mu$ represents a weighting factor. For the unlabeled data, $C$ is encouraged to provide high-confident predictions via posterior entropy minimization.

By integrating the training loss functions in the above three aspects: latent space regularization, data generation and semantic verification, the optimization formulation of our SphericGAN is expressed as follows:

$$\begin{aligned} &\min_{\mathcal{P},G,F}\max_{D_{lat},D_{prt}} L_{lat}^{clust} + L_{lat}^{eval} + L_{lat}^{adv} + L_{lat}^{align} + \nu L_{lat}^{cons}, \\ &\min_{G,C}\max_{D_{prt},D_{class}} L_{data}^{adv} + L_{data}^{eval}, \end{aligned} \quad (9)$$

where $\nu$ denotes a weighting factor. We consider that incorporating $\mathcal{P}$ is helpful for modeling a complex distribution, aligning real and fake clusters aims to capture class-independent variation factors, competing with $D_{prt}$ and $D_{class}$ leads to better synthesized image fidelity, and the guidance of $C$ is useful for capturing the subtle distinctions among fine-grained classes. All the components of SphericGAN are jointly optimized, and we summarize the training process in Algorithm 1.

**Algorithm 1** Pseudo-code of training our SphericGAN.

**Input:** Labeled data $X^{\mathbb{L}}$ and unlabeled data $X^{\mathbb{U}}$.
**Initialize:** Generator $G$, mapping network $F$, discriminators $\{D_{lat}, D_{prt}, D_{class}\}$, classifier $C$, latent prototypes $\mathcal{P}$, learning rates $\{\gamma, \varepsilon\}$, and number of training epochs $T$.
**for** $t = 1$ to $T$ **do**
  **for** each mini-batch **do**
    Sample vectors $z \sim p_0$, and synthesize latent codes $z^{\circ}$ and instances $(x_z, y_z)$.
    Sample labeled instances $(x_l, y_l)$ from $X^{\mathbb{L}}$, unlabeled instances $x_u$ from $X^{\mathbb{U}}$.
    Optimize $\mathcal{P}$ by using `Adam` [20]:
    $\mathcal{P} \leftarrow \texttt{Adam}(\nabla(L_{lat}^{clust} + L_{lat}^{eval}), \mathcal{P}, \gamma)$.
    Optimize $F$ by using `Adam`:
    $\theta_F \leftarrow \texttt{Adam}(\nabla(L_{lat}^{adv} + L_{lat}^{align} + \nu L_{lat}^{cons}), \theta_F, \gamma)$.
    Optimize $\{D_{lat}, D_{prt}, D_{class}\}$ by using `Adam`:
    $\theta_D^{lat} \leftarrow \texttt{Adam}(\nabla L_{lat}^{adv}, \theta_D^{lat}, \gamma)$,
    $\theta_D^{prt} \leftarrow \texttt{Adam}(\nabla(L_{lat}^{eval} + L_{lat}^{align} + L_{data}^{adv}), \theta_D^{prt}, \gamma)$,
    $\theta_D^{class} \leftarrow \texttt{Adam}(\nabla L_{data}^{adv}, \theta_D^{class}, \gamma)$,
    Optimize $G$ by using `Adam`:
    $\theta_G \leftarrow \texttt{Adam}(\nabla(L_{lat}^{adv} + L_{lat}^{align} + L_{data}^{adv} + L_{data}^{eval}), \theta_G, \gamma)$.
    Optimize $C$ by using stochastic gradient descent (`SGD`):
    $\theta_C \leftarrow \texttt{SGD}(\nabla L_{data}^{eval}, \theta_C, \varepsilon)$,
  **end for**
**end for**
**Return** $\theta_G, \theta_F, \{\theta_D^{lat}, \theta_D^{prt}, \theta_D^{class}\}, \theta_C$ and $\mathcal{P}$.

# 4. Experiments

We assess the generation performance of our proposed SphericGAN on multiple standard benchmarks. In addition to the comparison with state-of-the-art generic GANs and semi-supervised GANs, we also conduct extensive experiments for model analysis and explanatory visualization.

## 4.1. Datasets and Settings

**Benchmarks.** The experiments are conducted on diverse fine-grained benchmarks: CUB-200 [37] contains about 6K training images and 6K testing images from 200 fine-grained bird categories. FaceScrub-100 [39] is a human facial image dataset, and contains 13K training images and 2K test images from the largest 100 classes of FaceScrub [30]. In Stanford-Cars [23], there are 8K/8K training/testing images from 196 categories.

**Semi-supervised Settings.** To comply with the semi-supervised setting of competing methods, there are 2.8K/2K/4K randomly selected labeled data for CUB-200/FaceScrub-100/Stanford-Cars. The amount of labeled data is limited by the smallest classes when maintaining the balance among the classes. For CUB-200/FaceScrub-100/Stanford-Cars, 14/20/20 images per class are labeled, and the value of the labeled image is close to 0.5/0.2/0.5.

**Implementation Details.** All constituent networks are trained from scratch, and the number of training epochs is set to 500. We adopt the `Adam` optimizer [20] with the learning rate $\gamma = 0.0002$ and the momentum parameters of $(\beta_1 = 0.5, \beta_2 = 0.999)$ (`SGD` with $\varepsilon = 0.001$ for the classifier). The weighting factors $\lambda$ in Eq.(2), $\mu$ in Eq.(8) and $\nu$ in Eq.(9) are set to 4.0, 5.0 and 0.01, respectively.

**Baseline Model.** To illustrate the effectiveness of our improvement strategies, we build a baseline model by disabling the prototype-related components. Note that the baseline model is still comparable with SphericGAN in terms of model capacity.

**Evaluation Metrics.** The synthesized images are evaluated by the following four metrics: Fréchet Inception Distance (**FID**) [15], class-wise FID (**cFID**), Inception Score (**IS**) [32] and Recognition Accuracy (**RA**). Both FID and IS have been widely used to evaluate the performance of GAN-based methods. cFID is used to indicate how closely the real and synthesized data match with each other in each class. RA is assessed by an independent classifier, which is pre-trained with full supervision on each test dataset.

## 4.2. Quantitative Results

We conduct a number of quantitative experiments to assess the proposed SphericGAN, followed by comparison with a number of state-of-the-art GAN-based models in view of the availability of open-source codes. For a fair comparison, we implement our SphericGAN without any advanced GAN training strategies.

**Comparison with Baseline.** To highlight the effectiveness of our hyper-spherical latent space together with the corresponding regularizers in improving the quality of image synthesis, we first compare SphericGAN with the baseline model at different semi-supervised scenarios. Let $\tau$ denote the ratio of labeled data to all the training data. We conduct a series of experiments on CUB-200 and FaceScrub-100 by setting $\tau$ to 0.2, 0.3, 0.4, 0.5 and 1.0. The results shown in Figure 3 suggest that our strategies can lead to consistent improvement in all the cases.

**Relative Contributions.** We investigate the relative contributions of our adopted improvement strategies to the synthesis quality. The experiment is conducted on the representative dataset of CUB-200. We progressively enhance the baseline model by incorporating the following strategies one by one: latent clustering, semantic verification of

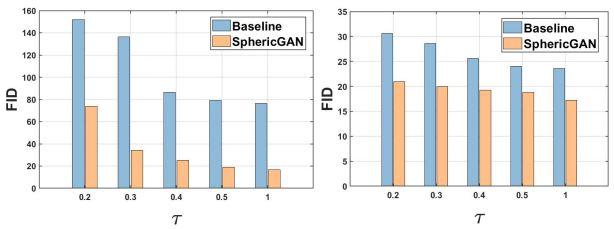



Figure 3. Comparison with the baseline model at different levels of supervision on CUB-200 (*left*) and FaceScrub-100 (*right*).

Table 1. Improvement over the baseline model on CUB-200. '*LC*' indicates latent clustering, '*SV*' indicates semantic verification of synthesized data, and '*CA*' indicates real-fake cluster alignment.

| Method | *LC* | *SV* | *CA* | FID↓ | cFID↓ | IS↑ | RA↑ |
|---|---|---|---|---|---|---|---|
| Baseline | | | | 79.12 | 186.48 | 4.34±.05 | 9.40 |
| | ✓ | | | 50.20 | 165.23 | 4.39±.03 | 9.88 |
| | ✓ | ✓ | | 30.15 | 115.35 | 4.60±.03 | 92.04 |
| | ✓ | ✓ | ✓ | 18.87 | 103.35 | 5.03±.05 | 98.46 |
| ***Improvement*** | - | - | - | ***-60.25*** | ***-83.13*** | ***+0.69*** | ***+89.06*** |

synthesized data, and real-fake cluster alignment. We assess the synthesis quality of the resulting models in terms of FID/cFID/IS/RA in Table 1. The results confirm the effectiveness of latent clustering and real-fake cluster alignment in improving the synthesized image fidelity. On the other hand, semantic verification of synthesized data plays an important role in inducing the generator to capture precise class semantics, since the improvement reaches about 50/82 (percentage) points in cFID/RA. Figure 4 shows representative synthesized images of the baseline model and variants. We can observe that the synthesis quality is improved progressively as the improvement strategies applied.

**Comparison with Unconditional GANs.** We compare our SphericGAN with representative unconditional GANs: SN-GAN [29], StyleGAN2 [19], FineGAN [34] and MixNMatch [25]. StyleGAN2 serves as a state-of-the-art generic image synthesis model. FineGAN and MixNMatch are developed for modeling fine-grained data. Note that the training images of the unconditional GANs are the same as the proposed approach, but the class labels of labeled samples are not included. The results are summarized in Table 2. We can make the following observations: SN-GAN significantly underperforms other competing methods. We consider that the GAN architecture is one of the important factors. The performance of FineGAN is comparable with that of MixNMatch, and StyleGAN2 performs better than both of them on CUB-200. Our proposed SphericGAN is able to outperform StyleGAN2 by about 7, 3 and 7 points in FID on the three datasets, respectively.

**Comparison with Class-conditional GANs.** Furthermore, we compare our SphericGAN with the existing semi-supervised generative models, including TripleGAN [24], EnhancedTGAN [39], Δ-GAN [13], R$^3$-CGAN [26] and SSC-GAN [7]. All the competing models are trained in the same semi-supervised setting. Table 2 shows that SSC-GAN achieves the previous state-of-the-art results, and our SphericGAN improves the results from 20.03/20.65/39.02 to 18.87/18.84/35.69 on CUB-200/FaceScrub-100/Stanford-Cars in FID. We believe that our hyper-spherical latent space and the cluster-based regularization contribute to the superior performance, since both SphericalGAN and SSC-GAN are based on the network architectures of FineGAN. In addition, we report the results of the fully supervised BigGAN [5], which serves as a state-of-the-art generic class-conditional GAN. We find that the generation performance of our SphericGAN is comparable with that of BigGAN on CUB-200 and FaceScrub-100.


Figure 4. Representative images synthesized by the baseline model and variants on CUB-200.

### 4.3. Qualitative Results

**Real-fake Clusters.** In the training process of SphericGAN, we encourage latent vectors to move toward the nearest prototypes, and further match the resulting latent clusters with the clusters of real images. It is interesting to investigate whether a latent cluster is consistent with the matched cluster in semantics. In Figure 5, we show a number of real and synthesized images in three representative clusters on each test dataset. On both CUB-200 and Stanford-Cars, the clusters capture the factors of object shape. Since the facial images in FaceScrub-100 are cropped and aligned well, the clusters are associated with the factor of face pose. We apply SphericGAN to generate images with the same class label and different latent code $z^\circ$ (belong to the three latent clusters, respectively). The synthesized images hold class-independent semantics similar to the real ones. In addition, we visualize the distributions of the synthesized images from $z^\circ$ and real images associated with 5 latent clusters via the t-SNE embedding [28] of the prototypical discriminator features, and Figure 6 shows that the statistics of the two types of data are matched.

Table 2. Comparison between SphericGAN and state-of-the-art unconditional/class-conditional GANs in fine-grained image synthesis. $*$ indicates that an unconditional GAN is trained on the same data as semi-supervised GANs, without using the class labels of labeled data.

| | CUB-200 | | | FaceScrub-100 | | | Stanford-Cars | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | FID↓ | IS↑ | RA↑ | FID↓ | IS↑ | RA↑ | FID↓ | IS↑ | RA↑ |
| ***Unconditional GANs*** | | | | | | | | | |
| SN-GAN* [29] | 160.09 | 4.21±0.05 | - | 41.26 | 1.66±0.05 | - | 53.20 | 2.80±0.05 | - |
| FineGAN* [34] | 46.68 | 4.62±0.03 | - | 24.63 | 1.76±0.02 | - | 45.72 | 2.85±0.04 | - |
| MixNMatch* [25] | 45.59 | 4.78±0.08 | - | 25.63 | 1.71±0.05 | - | 45.94 | 2.60±0.05 | - |
| StyleGAN2* [19] | 25.58 | 4.12±0.07 | - | 22.14 | 1.84±0.04 | - | 42.35 | 2.98±0.06 | - |
| ***Semi-supervised GANs*** | | | | | | | | | |
| Triple-GAN [24] | 140.94 | 3.94±0.06 | 9.35 | 91.05 | 1.45±0.03 | 36.21 | 114.12 | 2.45±0.06 | 4.43 |
| EnhancedTGAN [39] | 133.57 | 4.17±0.03 | 9.16 | 57.58 | 1.57±0.02 | 62.69 | 105.20 | 2.43±0.05 | 3.48 |
| Δ-GAN [13] | 96.42 | 4.36±0.05 | 9.01 | 35.49 | 1.71±0.04 | 94.99 | 61.44 | 2.77±0.10 | 4.74 |
| $R^3$-CGAN [26] | 88.62 | 4.43±0.06 | 8.60 | 25.28 | 1.73±0.02 | 74.30 | 44.57 | 3.05±0.04 | 5.48 |
| SSC-GAN [7] | 20.03 | 4.68±0.04 | 97.85 | 20.65 | 1.82±0.03 | 96.86 | 39.02 | 3.10±0.03 | 87.45 |
| Baseline | 79.12 | 4.34±0.05 | 9.40 | 31.60 | 1.71±0.03 | 86.23 | 53.34 | 2.89±0.05 | 7.32 |
| SphericGAN | **18.87** | **5.03±0.05** | **98.46** | **18.84** | **1.93±0.04** | **96.99** | **35.69** | **3.22±0.03** | **88.80** |
| ***Full supervised GAN*** | | | | | | | | | |
| BigGAN [5] | 22.57 | 5.36±0.07 | - | 15.51 | 1.84±0.01 | - | 32.45 | 3.26±0.02 | - |

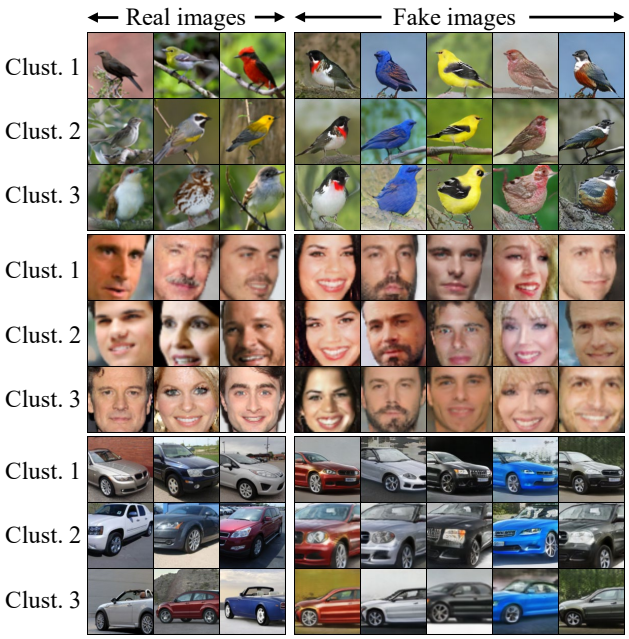
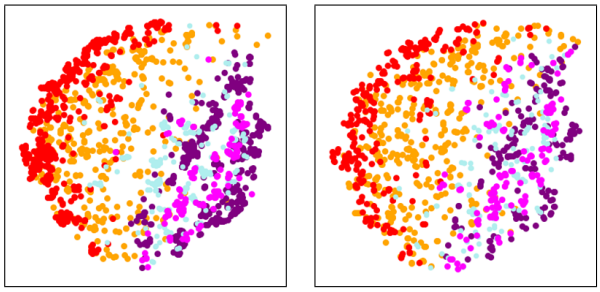



Figure 5. Visualization of three representative clusters on real and fake images. Fake images are synthesized by varying class label $y$ in each column.



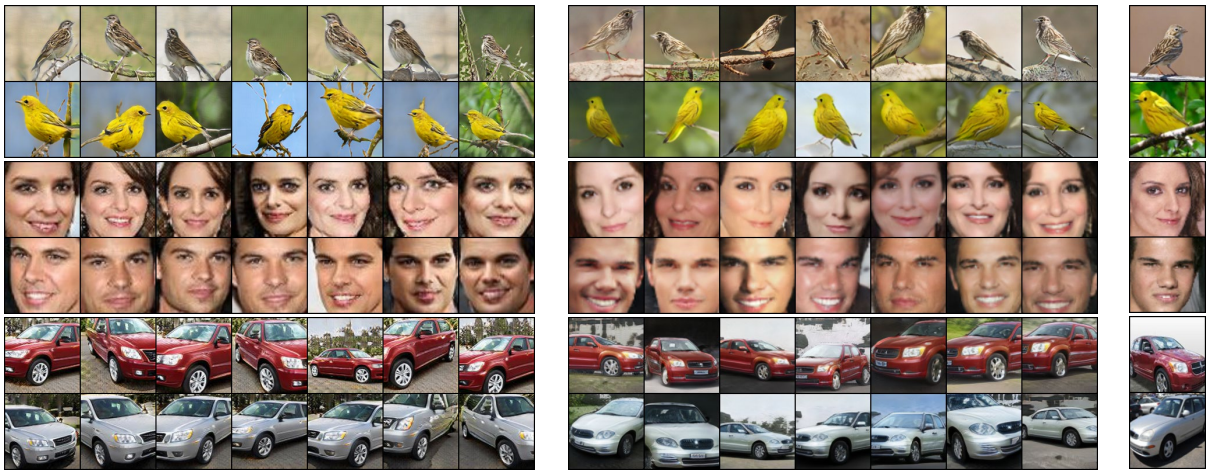
(a) Real instances (b) Fake instances

Figure 6. The t-SNE plot of synthesized and real instances associated with 5 latent clusters, which are marked in different colors.

**Spherical Interpolation.** To explore the continuity and smoothness of the hyper-spherical latent space, we apply spherical interpolation between paired latent vectors and synthesize new images with the interpolated vectors. In the experiment, we sample a latent prototype and two different latent vectors, and determine a spherical interpolation path, from the first latent vector through the latent prototype and finally to the other latent vector. By fixing class label, the synthesized images along the path hold the same class semantics. Each example shown in Figure 7 reveals a smooth transformation, which suggests that the hyper-spherical latent space encodes rich information on class-independent semantics.

**Visual Comparison.** In addition to the quantitative evaluation in the previous subsection, we also visually compare our SphericGAN with two popular GANs: BigGAN and StyleGAN2. In Figure 8, we show the images synthesized by SphericGAN and BigGAN on a number of classes. By comparing with the given real images, we find that SphericGAN is able to capture precise class semantics as well as BigGAN. Since StyleGAN2 is an unconditional generative

Figure 7. Visualization of interpolation paths in the hyper-spherical latent space. The red boxes indicate the images synthesized from a starting latent vector, a latent prototype and an end latent vector.



(a) Fully supervised BigGAN (b) Semi-supervised SphericGAN (c) Real

Figure 8. Representative images synthesized by BigGAN and SphericGAN on CUB-200, FaceScrub-100 and Stanford-Car.

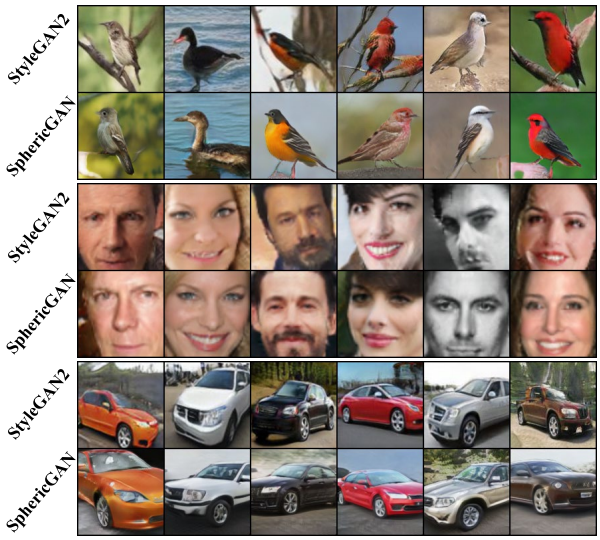



Figure 9. Visual comparison between StyleGAN2 and our proposed SphericGAN.

model, the class labels of the synthesized images cannot be specified, and we thus select the most similar results for comparison. The representative synthesized images shown in Figure 9 demonstrate that the image fidelity of SphericGAN is higher than that of StyleGAN2.

## 5. Conclusion

Compared to generic image synthesis, it is more challenging to synthesize class-specific fine-grained data, due to limited training data and subtle inter-class distinctions. In this paper, we present SphericGAN, a semi-supervised hyper-spherical GAN for this task. To better capture class-independent variation factors, we learn a hyper-spherical latent space, in which we group latent vectors to match the underlying structure of real data. The latent clusters are further associated with semantics by aligning with the prior clusters of real instances. Since the label information is not required in this process, the unlabeled data can be well utilized, which eventually benefits the synthesis quality. In the experiments, we demonstrate the superior performance of SphericGAN in fine-grained image synthesis.

## Acknowledgments

This work was supported in part by the China Scholarship Council, in part by the National Natural Science Foundation of China (Project No. 62072188, 62072189), in part by the Research Grants Council of the Hong Kong Special Administration Region (Project No. CityU 11201220), and in part by the Natural Science Foundation of Guangdong Province (Project No. 2020A1515010484).

## References


[1] Rammen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: how to embed images into the StyleGAN latent space? In *Proc. IEEE International Conference on Computer Vision*, 2019. 2

[2] Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In *Proc. International Conference on Machine Learning*, 2017. 3

[3] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. CVAE-GAN: fine-grained image generation through asymmetric training. In *Proc. International Conference on Computer Vision*, 2017. 3

[4] Yaniv Benny and Lior Wolf. OneGAN: simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *Proc. European Conference on Computer Vision*, 2020. 1, 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. International Conference on Learning Representation*, 2018. 1, 6, 7

[6] Ting-Yun Chang and Chi-Jen Lu. TinyGAN: distilling biggan for conditional image generation. In *Proc. Asian Conference on Computer Vision*, 2020. 1

[7] Tianyi Chen, Yi Liu, Yunfei Zhang, Si Wu, Yong Xu, Feng Liangbing, and Hau San Wong. Semi-supervised single-stage controllable gans for conditional fine-grained image generation. In *Proc. International Conference on Computer Vision*, pages 9264–9273, October 2021. 3, 6, 7

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In *Proc. Neural Information Processing Systmes*, 2016. 3

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[10] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: diverse image synthesis for multiple domains. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[11] Zhijie Deng, Hao Zhang, Xiaodan Liang, Luona Yang, Shizhen Xu, Jun Zhu, and Eric P. Xing. Strctured generative adversarial networks. In *Proc. Neural Information Processing Systmes*, 2017. 2

[12] Jinhao Dong and Tong Lin. MarginGAN: adversarial training in semi-supervised learning. In *Proc. Neural Information Processing Systmes*, 2019. 3

[13] Zhe Gan, Liqun Chen, Weiyao Wang, Yunchen Pu, Yizhe Zhang, Hao Liu, Chunyuan Li, and Lawrence Carin. Triangle generative adversarial networks. In *Proc. Neural Information Processing Systmes*, 2017. 3, 6, 7

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Neural Information Processing Systmes*, 2014. 1

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, and Bernhard Nessler. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Neural Information Processing Systmes*, 2017. 5

[16] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proc. International Conference on Machine Learning*, pages 1558–1567. PMLR, 2017. 4

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. International Conference on Learning Representation*, 2018. 1

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 6, 7

[20] Diederik Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representations*, 2015. 5

[21] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proc. International Conference on Learning Representation*, 2014. 1

[22] Diederik P. Kingma, Shakir Mohamed, Danilo J. Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Proc. Neural Information Processing Systmes*, pages 3581 – 3589, 2017. 2

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proc. IEEE Workshop on 3D Representation and Recognition*, 2013. 5

[24] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Proc. Neural Information Processing Systmes*, 2017. 2, 3, 6, 7

[25] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. MixNMatch: multifactor disentanglement and encoding for conditional image generation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3, 6, 7

[26] Yi Liu, Guangchang Deng, Xiangping Zeng, Si Wu, Zhiwen Yu, and Hau-San Wong. Regularizing discriminative capability of CGANs for semi-supervised generative learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3, 6, 7

[27] Yi Liu, Xiaoyang Huo, Tianyi Chen, Xiangping Zeng, Si Wu, Zhiwen Yu, and Hau-San Wong. Mask-embedded discriminator with region-based semantic regularization for semi-supervised class-conditional image synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5506–5515, June 2021. 1, 3

[28] L. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579 – 2605, 2008. 6

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. International Conference on Learning Representation*, 2018. 6, 7

[30] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *Proc. IEEE International Conference on Image Processing*, 2014. 5

[31] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016. 1

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proc. Neural Information Processing Systmes*, 2016. 3, 5

[33] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: learning a generative model from a single natural image. In *Proc. International Conference on Computer Vision*, 2019. 1

[34] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3, 6, 7

[35] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *Proc. International Conference on Learning Representation*, 2016. 3

[36] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016. 1

[37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Technical Report CNS-TR-2011-001, California Institute of Technology*, 2011. 5

[38] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of Wasserstein GANs: a consistency term and its dual effect. In *Proc. International Conference on Learning Representation*, 2018. 3

[39] Si Wu, Guangchang Deng, Jichang Li, Rui Li, Zhiwen Yu, and Hau-San Wong. Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3, 5, 6, 7

[40] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. LR-GAN: layered recursive generative adversarial networks for image generation. In *Proc. International Conference on Learning Representation*, 2017. 3

[41] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proc. International conference on machine learning*, pages 7354–7363, 2019. 1