Adversarial Learning with Local Coordinate Coding

Jiezhang Cao [*1] **Yong Guo** [*1] **Qingyao Wu** [*1] **Chunhua Shen** [2] **Junzhou Huang** [3] **Mingkui Tan** [1]

Abstract

Generative adversarial networks (GANs) aim to generate realistic data from some prior distribution (*e.g.*, Gaussian noises). However, such prior distribution is often independent of real data and thus may lose semantic information (*e.g.*, geometric structure or content in images) of data. In practice, the semantic information might be represented by some latent distribution learned from data, which, however, is hard to be used for sampling in GANs. In this paper, rather than sampling from the pre-defined prior distribution, we propose a Local Coordinate Coding (LCC) based sampling method to improve GANs. We derive a generalization bound for LCC based GANs and prove that a small dimensional input is sufficient to achieve good generalization performance. Extensive experiments on various real-world datasets demonstrate the effectiveness of the proposed method.

1. Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have been successfully applied to many tasks, such as video prediction (Ranzato et al., 2014; Mathieu et al., 2016), image translation (Isola et al., 2017; Kim et al., 2017), etc. Specifically, GANs learn to generate data by playing a two-player game: a generator tries to produce samples from a simple latent distribution, and a discriminator distinguishes between the generated data and real data.

Recently, many attempts have been made to improve GANs (Radford et al., 2015; Arjovsky et al., 2017; Karras et al., 2018). However, existing studies suffer from two limitations. First, many studies employ some simple prior distribution, such as Gaussian distributions (Goodfellow et al., 2014) and uniform distributions (Radford et al., 2015). However, such pre-defined prior distributions are often independent of the data distributions and these methods may produce images with distorted structures without sufficient semantic information. Although such semantic information can be represented by some latent distribution, *e.g.*, extracting embeddings using an AutoEncoder (Hinton & Salakhutdinov, 2006), how to conduct sampling from this distribution still remains an open question in GANs.

Second, the generalization ability of GANs w.r.t. the dimension of the latent distribution is unknown. In practice, we observe that the performance of GANs is sensitive to the dimension of the latent distribution. Unfortunately, it is difficult to analyze the dimensionality of the latent distribution, since the specified prior distribution is independent of the real data. Therefore, it is very necessary and important to explore a new method to study the dimension of latent distribution and its impacts on the generalization ability.

In this paper, relying on the manifold assumption on images (Tenenbaum et al., 2000; Roweis & Saul, 2000), we propose a novel generative model using Local Coordinate Coding (LCC) (Yu et al., 2009) to improve GANs in generating perceptually convincing images. First, we employ an AutoEncoder to learn embeddings lying on the latent manifold to capture the semantic information in data. Then, we develop a new LCC sampling method for training GANs by exploiting the local information on the latent manifold.

The contributions of this paper are summarized as follows.

First, we propose an LCC sampling method for GANs to capture the local information of data. With the LCC sampling, the proposed scheme, called LCC-GANs, is able to sample meaningful points from the latent manifold to generate new data.

Second, we study the generalization bound of LCC-GANs based on the Rademacher complexity of the discriminator set and the error w.r.t. the intrinsic dimensionality of the manifold. In particular, we prove that a small dimensional input is sufficient to achieve good generalization performance. Extensive experiments on real-world datasets demonstrate the superiority of the proposed method over several state-of-the-arts.

[*] Equal contribution [1] School of Software Engineering, South China University of Technology, China [2] School of Computer Science, The University of Adelaide, Australia [3] Tencent AI Lab, China; University of Texas at Arlington, America. Correspondence to: Mingkui Tan <mingkuitan@scut.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

2. Related Studies

Recently, Generative Adversarial Networks have shown promising performance for generating images, such as DCGANs (Radford et al., 2015), WGANs (Arjovsky et al., 2017) and Progressive GANs (Karras et al., 2018). Most existing generative models seek to learn from some simple prior distribution, such as Gaussian distributions and uniform distributions, to generate samples (Goodfellow et al., 2014; Arjovsky et al., 2017; Radford et al., 2015; Karras et al., 2018). However, such prior distributions are independent of the data distributions, which may lose semantic information and lead to difficulties in analyzing the dimension of latent space.

Besides, some generative models do sampling via some learned posterior distribution. For example, Variational AutoEncoder (VAE) (Kingma & Welling, 2014), Wasserstein AutoEncoder (WAE) (Tolstikhin et al., 2018) and Adversarial AutoEncoder (AAE) (Makhzani et al., 2015) enforce the posterior distribution to match the prior distribution. However, it is difficult for these methods to conduct sampling directly on the posterior distribution. Moreover, although these methods help to make inference, overly simplified distributions would also lose semantic information.

3. Preliminaries

3.1. Local Coordinate Coding

We first introduce some definitions about local coordinate coding which will be used to develop our proposed method.

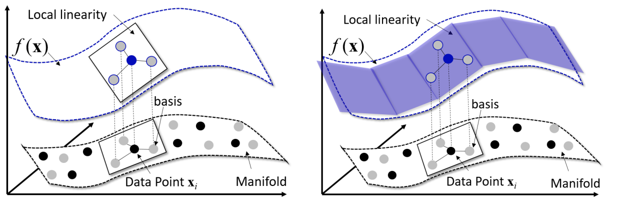
Definition 1 (Lipschitz Smoothness (Yu et al., 2009)) *A function $f_\theta(\mathbf{x})$ in $\mathbb{R}^d$ is $(L_\mathbf{x}, L_f)$-Lipschitz smooth if $\|f(\mathbf{x}') - f(\mathbf{x})\|_2 \leq L_\mathbf{x}\|\mathbf{x} - \mathbf{x}'\|_2$ and $\|f(\mathbf{x}') - f(\mathbf{x}) - \nabla f(\mathbf{x})^\mathsf{T}(\mathbf{x}' - \mathbf{x})\|_2 \leq L_f\|\mathbf{x} - \mathbf{x}'\|_2^2$, where $L_\mathbf{x}, L_f > 0$.*

Definition 2 (Coordinate Coding (Yu et al., 2009)) *A coordinate coding is a pair $(\boldsymbol{\gamma}, \mathcal{C})$, where $\mathcal{C} \subset \mathbb{R}^d$ is a set of anchor points (bases), and γ is a map of $\mathbf{x} \in \mathbb{R}^d$ to $[\gamma_\mathbf{v}(\mathbf{x})]_{\mathbf{v}\in\mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|}$ such that $\sum_\mathbf{v} \gamma_\mathbf{v}(\mathbf{x}) = 1$. Then, the physical approximation of $\mathbf{x} \in \mathbb{R}^d$ is $\mathbf{r}(\mathbf{x}) = \sum_{\mathbf{v}\in\mathcal{C}} \gamma_\mathbf{v}(\mathbf{x})\mathbf{v}$.*

Definition 2 indicates that any point in $\mathbb{R}^d$ can be represented by a linear combination of a set of anchor points.

3.2. Latent Manifold

High dimensional data often lie on some low dimensional manifold (Tenenbaum et al., 2000; Roweis & Saul, 2000). Based on this manifold assumption, we can learn a manifold $\mathcal{M}$ embedded in the latent space $\mathbb{R}^{d_B}$ by some manifold learning method, such as an AutoEncoder (AE) (Hinton & Salakhutdinov, 2006), to capture the semantic information of data. Given N training data $\{\mathbf{x}_i\}_{i=1}^N$, we can use an Encoder to extract the embeddings $\{\mathbf{h}_i\}_{i=1}^N$, where $\mathbf{h}_i = \mathrm{Encoder}(\mathbf{x}_i)$. Formally, the latent manifold can be defined as follows.



(a) Local function approximation based on LCC

(b) Global function approximation based on LCC

Figure 1. A geometric view of Local Coordinate Coding. Given a set of local bases, if data lie on a manifold, a nonlinear function $f(\mathbf{x})$ can be locally approximated by a linear function w.r.t. the coding. Given all bases, $f(\mathbf{x})$ can be globally approximated.

Definition 3 (Latent Manifold (Yu et al., 2009)) *A subset $\mathcal{M}$ embedded in the latent space $\mathbb{R}^{d_B}$ is called a smooth manifold with a* ***intrinsic dimension*** *$d := d_\mathcal{M}$, if there exists a constant $c_\mathcal{M}$, such that given any $\mathbf{h} \in \mathcal{M}$, there are d bases $\mathbf{v}_1(\mathbf{h}), \ldots, \mathbf{v}_d(\mathbf{h}) \in \mathbb{R}^{d_B}$ so that $\forall\ \mathbf{h}' \in \mathcal{M}$:*

$$\inf\nolimits_{\boldsymbol{\gamma}\in\mathbb{R}^d} \left\|\mathbf{h}' - \mathbf{h} - \sum\nolimits_{j=1}^d \gamma_j \mathbf{v}_j(\mathbf{h})\right\|_2 \leq c_\mathcal{M}\|\mathbf{h}' - \mathbf{h}\|_2^2.$$

where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_d]^\mathsf{T}$ is the local coding of a latent point $\mathbf{h}$ using the corresponding bases.

3.3. Generative Adversarial Networks

We apply the neural network distance (Arora et al., 2017) to measure the similarity between two distributions.

Definition 4 (Neural Network Distance (Arora et al., 2017)) *Let $\mathcal{F}$ be a set of neural networks from $\mathbb{R}^d$ to $[0, 1]$ and ϕ be a concave measure function, then for $D \in \mathcal{F}$, the neural network distance w.r.t. ϕ between two distributions μ and ν can be defined as*

$$d_{\mathcal{F},\phi}(\mu,\nu) = \sup_{D\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \left[\phi(D(\mathbf{x}))\right] + \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu} \left[\phi(\widetilde{D}(\mathbf{x}))\right] \right| - \phi_c,$$

where $\phi_c = 2\phi(\frac{1}{2})$ is a constant with the given ϕ and $\widetilde{D}(\mathbf{x}) = 1 - D(\mathbf{x})$. For simplicity, we omit the constant ϕ_c.

Objective function of general GANs. Given a Generator G_u and a Discriminator D_v parameterized by $u \in \mathcal{U}$ and $v \in \mathcal{V}$, where $\mathcal{U}$ and $\mathcal{V}$ are parameter spaces. Let $\mathcal{D}_{real}$ be the real distribution of training samples $\mathbf{x} \in \mathbb{R}^d$ and $\mathcal{D}_{G_u}$ be the distribution generated by G_u. The objective function of GANs can be defined as:

$$\min_{u\in\mathcal{U}} \max_{v\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{real}} \left[\phi(D_v(\mathbf{x}))\right] + \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{G_u}} \left[\phi(1 - D_v(\mathbf{x}))\right],$$

where $\phi : [0, 1] \to \mathbb{R}$ is any monotone function.

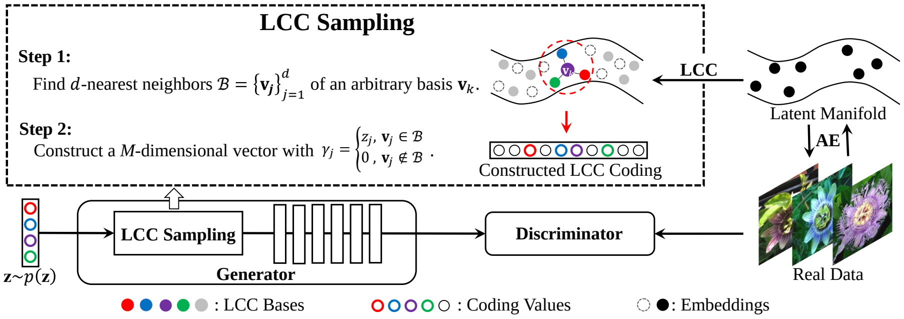

LCC Sampling
Step 1:
Find d-nearest neighbors $\mathcal{B}=\{\mathbf{v}_j\}_{j=1}^{d}$ of an arbitrary basis $\mathbf{v}_k$.
Step 2:
Construct a M-dimensional vector with $\gamma_j=\begin{cases} z_j, \mathbf{v}_j\in\mathcal{B} \\ 0, \mathbf{v}_j\notin\mathcal{B}\end{cases}$.
$\mathbf{v}_k$
Constructed LCC Coding
LCC
Latent Manifold
AE
Real Data
LCC Sampling
Generator
$\mathbf{z}\sim p(\mathbf{z})$
Discriminator
● ● ● ● ● : LCC Bases
○ ○ ○ ○ ○ : Coding Values
◌ ●: Embeddings

Figure 2. The scheme of the proposed LCC-GANs. We use an AutoEncoder to learn the embeddings on the latent manifold from real data. Relying on LCC, we learn a set of bases such that the LCC sampling can be conducted. As a result, the proposed method is able to take the constructed LCC codings to generate new data.

4. Adversarial Learning with LCC

In this section, we seek to improve GANs by exploiting LCC. The overall structure of the proposed method, called LCC-GANs, is illustrated in Figure 2.

As shown in Figure 2, instead of sampling from some predefined prior distribution, we seek to sample points from a learned latent manifold for training GANs. Specifically, we use an AutoEncoder (AE) to learn embeddings over a latent manifold of real data and then employ LCC to learn a set of bases to form local coordinate systems on the latent manifold. After that, we introduce LCC into GANs by approximating the generator using a linear function w.r.t. a set of codings (see Section 4.1). Relying on such approximation, we then propose an LCC based sampling method to exploit the local information of data on the latent manifold (see Section 4.3). The details of the proposed method are illustrated in following subsections.

4.1. Generator Approximation Based on LCC

According to Definition 3, any point on the latent manifold can be approximated by a linear combination of a set of local bases. Inspired by this, if the bases are sufficiently localized, the generator of GANs can also be approximated by a linear function w.r.t. a set of codings.

Lemma 1 **(Generator Approximation)** *Let $(\boldsymbol{\gamma},\mathcal{C})$ be an arbitrary coordinate coding on $\mathbb{R}^{d_B}$. Given a $(L_{\mathbf{h}},L_G)$-Lipschitz smooth generator $G_u(\mathbf{h})$, for all $\mathbf{h}\in\mathbb{R}^{d_B}$:*

$$\left\| G_u\left(\sum\nolimits_{\mathbf{v}\in\mathcal{C}}\gamma_{\mathbf{v}}(\mathbf{h})\mathbf{v}\right)-\sum\nolimits_{\mathbf{v}\in\mathcal{C}}\gamma_{\mathbf{v}}(\mathbf{h})G_u(\mathbf{v})\right\|_2 \\ \le 2L_{\mathbf{h}}\|\mathbf{h}-\mathbf{r}(\mathbf{h})\|_2+L_G\sum\nolimits_{\mathbf{v}\in\mathcal{C}}|\gamma_{\mathbf{v}}(\mathbf{h})|\cdot\|\mathbf{v}-\mathbf{r}(\mathbf{h})\|_2^2, \qquad (1)$$

where $\mathbf{r}(\mathbf{h})=\sum_{\mathbf{v}\in\mathcal{C}}\gamma_{\mathbf{v}}(\mathbf{h})\mathbf{v}$.

Given the local bases and a Lipschitz smooth generator, the generator w.r.t. the linear combination of the local bases can be approximated by the linear combination of the generator w.r.t. local bases. Since two close latent points often share the same local bases but with different weights (*i.e.*, codings), we can change these weights for generator approximation. Therefore, the pieces of generated data can cover an entire manifold seamlessly (see Figure 1(b)).

Objective function of LCC. We minimize the right-hand term of the inequality in (1) to obtain a set of bases. Given a set of the latent points $\{\mathbf{h}_i\}_{i=1}^{N}$, by assuming $\mathbf{h}\approx\mathbf{r}(\mathbf{h})$ (Yu et al., 2009), we address the following problem:

$$\min_{\boldsymbol{\gamma},\mathcal{C}}\sum_{\mathbf{h}}2L_{\mathbf{h}}\|\mathbf{h}-\mathbf{r}(\mathbf{h})\|_2+L_G\sum_{\mathbf{v}\in\mathcal{C}}|\gamma_{\mathbf{v}}(\mathbf{h})|\cdot\|\mathbf{v}-\mathbf{h}\|_2^2, \\ \text{s.t.}\sum_{\mathbf{v}\in\mathcal{C}}\gamma_{\mathbf{v}}(\mathbf{h})=1,\ \forall\,\mathbf{h}, \qquad (2)$$

where $\mathbf{r}(\mathbf{h})=\sum_{\mathbf{v}\in\mathcal{C}}\gamma_{\mathbf{v}}(\mathbf{h})\mathbf{v}$. In practice, we update $\boldsymbol{\gamma}$ and $\mathcal{C}$ by alternately optimizing a LASSO problem and a least-square regression problem, respectively.

4.2. Objective Function of LCC-GANs

After solving Problem (2), every latent point $\mathbf{h}\in\mathbb{R}^{d_B}$ is close to its physical approximation $\mathbf{r}(\mathbf{h})$, *i.e.*, $\mathbf{h}\approx\mathbf{r}(\mathbf{h})$, then the generator can be approximated by

$$G_u(\mathbf{h})\approx G_u(\mathbf{r}(\mathbf{h}))\triangleq G_w(\boldsymbol{\gamma}(\mathbf{h})),\mathbf{h}\in\mathcal{H}, \qquad (3)$$

where $\mathbf{r}(\mathbf{h})=\mathbf{V}\boldsymbol{\gamma}(\mathbf{h})$, $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_M]$ and $\boldsymbol{\gamma}(\mathbf{h})=[\gamma_1(\mathbf{h}),\gamma_2(\mathbf{h}),\ldots,\gamma_M(\mathbf{h})]^{\mathsf{T}}$ with $M=|\mathcal{C}|$. Here, $\mathcal{H}$ is the latent distribution and $w\in\mathcal{W}$ is the parameters of the generator w.r.t. u and fixed $\mathbf{V}$ learned from Problem (2).

Using the neural network distance, we consider the following objective function of LCC-GANs between the generated distribution and the empirical distribution:

$$\min_{G_w\in\mathcal{G}}\ d_{\mathcal{F},\phi}\left(\widehat{\mathcal{D}}_{G_w(\boldsymbol{\gamma}(\mathbf{h}))},\widehat{\mathcal{D}}_{real}\right),\mathbf{h}\in\mathcal{H}. \qquad (4)$$

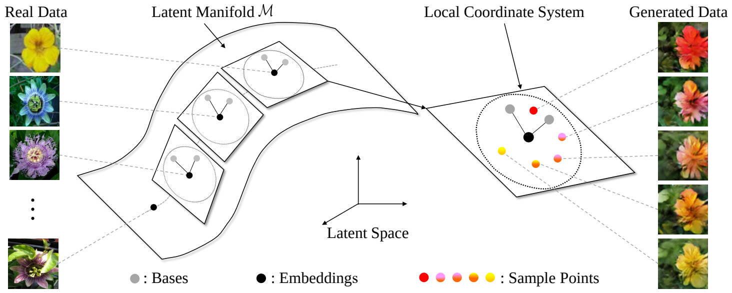


Figure 3. The geometric views on LCC Sampling. By learning embeddings (*i.e.*, black points) which lie on the latent manifold, we use LCC to learn a set of bases (*i.e.*, gray points) to form a local coordinate system such that we can sample different latent points (*i.e.*, coloured points) by LCC sampling. As a result, LCC-GANs can generate new data which have different attributes.

To be more specific, Problem (4) can be rewritten as:

$$\min_{w\in\mathcal{W}}\max_{v\in\mathcal{V}}\mathop{\mathbb{E}}_{\mathbf{x}\sim\widehat{\mathcal{D}}_{real}}\Big[\phi(D_v(\mathbf{x}))\Big]+\mathop{\mathbb{E}}_{\mathbf{h}\sim\mathcal{H}}\Big[\phi\big(\widetilde{D}_v\left(G_w\left(\boldsymbol{\gamma}(\mathbf{h})\right)\right)\big)\Big],$$

where $\phi(\cdot)$ is a monotone function, and $\widetilde{D}_v(\cdot)=1-D_v(\cdot)$. The detailed algorithm is shown in Algorithm 1.

4.3. LCC Sampling Method

To address Problem (4), one of the key issues is on how to conduct sampling from the learned latent manifold. Although the latent manifold can be learned by AutoEncoder, it is very hard to sample valid points on it to train GANs. To address this, we propose an LCC sampling method to capture the latent distribution on the learned latent manifold (see Figure 3). The proposed sampling method contains the following two steps.

Step 1: Given a local coordinate system, we randomly select a latent point (specifically, it can be a basis), and then find its d-nearest neighbors $\mathcal{B}=\{\mathbf{v}_j\}_{j=1}^d$.

Step 2: We construct an M-dimensional vector $\boldsymbol{\gamma}(\mathbf{h})=[\gamma_1(\mathbf{h}),\gamma_2(\mathbf{h}),\ldots,\gamma_M(\mathbf{h})]^{\mathsf{T}}$ as the LCC coding for sampling. Here, each element of $\boldsymbol{\gamma}(\mathbf{h})$ is corresponding to the weight of the basis. To conduct local sampling, we construct the coding of the neighbors $\mathcal{B}$ as follows:

$$\gamma_j(\mathbf{h})=\begin{cases} z_j, & \mathbf{v}_j\in\mathcal{B} \\ 0, & \mathbf{v}_j\notin\mathcal{B}\end{cases},$$

where z_j is the j-th element of $\mathbf{z}\in\mathbb{R}^d$ from the prior distribution $p(\mathbf{z})$. Here, we set $p(\mathbf{z})$ to be the standard Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$. Finally, we obtain a new latent point $\mathbf{V}\boldsymbol{\gamma}(\mathbf{h})$.

Based on Definition 3, the intrinsic dimensionality is determined by the number of bases in a local region. Thus, we turn the determination of intrinsic dimension into an easier problem of selecting sufficient number of local bases.

Algorithm 1 LCC-GANs Training Method.

```
Initialize: Training data $\{\mathbf{x}_i\}_{i=1}^N$; a prior distribution $p(\mathbf{z})$,
    where $\mathbf{z}\in\mathbb{R}^d$; minibatch size $n$.
Learn the latent manifold $\mathcal{M}$ using an AutoEncoder
Construct LCC bases $\{\mathbf{v}_i\}_{i=1}^M$ on $\mathcal{H}$ by optimizing:
    $\min_{\boldsymbol{\gamma},C}\sum_{\mathbf{h}}2L_{\mathbf{h}}\|\mathbf{h}-\mathbf{r}(\mathbf{h})\|_2+L_G\sum_{\mathbf{v}\in C}|\gamma_{\mathbf{v}}(\mathbf{h})|\cdot\|\mathbf{v}-\mathbf{h}\|_2^2$
for number of training iterations do
    Do LCC Sampling to obtain a minibatch $\{\gamma(\mathbf{h}_i)\}_{i=1}^n$
    Sample a minibatch $\{\mathbf{x}_i\}_{i=1}^n$ from the data distribution
    Update the discriminator by ascending the gradient:
        $\nabla_v\frac{1}{n}\sum_{i=1}^n\phi(D_v(\mathbf{x}_i))+\phi((1-D_v(G_w(\gamma(\mathbf{h}_i)))))$
    Do LCC Sampling to obtain a minibatch $\{\gamma(\mathbf{h}_i)\}_{i=1}^n$
    Update the generator by descending the gradient:
        $\nabla_w\frac{1}{n}\sum_{i=1}^n\phi(1-D_v(G_w(\gamma(\mathbf{h}_i))))$
end for
```

5. Theoretical Analysis

We first give some necessary notations to develop our theoretical analysis for LCC based GANs. Let $\{\mathbf{x}_i\}_{i=1}^N$ be a set of observed training samples drawn from the real distribution $\mathcal{D}_{real}$, and let $\widehat{\mathcal{D}}_{real}$ denote the empirical distribution over $\{\mathbf{x}_i\}_{i=1}^N$. Given a generator G_u and a set of the latent points $\{\mathbf{h}_i\}_{i=1}^r$, $\{G_u(\mathbf{h}_i)\}_{i=1}^r$ denotes a set of r generated samples from the generated distribution $\mathcal{D}_{G_u}$, and $\widehat{\mathcal{D}}_{G_w}$ is an empirical generated distribution. Motivated by (Arora et al., 2017; Zhang et al., 2018), we define the generalization of LCC-GANs as follows:

Definition 5 (Generalization) *The neural network distance $d_{\mathcal{F},\phi}(\cdot,\cdot)$ between distributions generalizes with N training samples and error ϵ, if for a learned distribution $\mathcal{D}_{G_u}$, the following holds with high probability,*

$$\left|d_{\mathcal{F},\phi}\left(\widehat{\mathcal{D}}_{G_w},\mathcal{D}_{real}\right)-\inf_{\mathcal{G}}d_{\mathcal{F},\phi}\left(\mathcal{D}_{G_u},\mathcal{D}_{real}\right)\right|\leq\epsilon.$$

In Definition 5, the generalization of GANs means that the population distance $d_{\mathcal{F},\phi}(\mathcal{D}_{G_u},\mathcal{D}_{real})$ is close to the distance $d_{\mathcal{F},\phi}(\widehat{\mathcal{D}}_{G_w},\mathcal{D}_{real})$. In theory, we hope to obtain a small $d_{\mathcal{F},\phi}(\mathcal{D}_{G_u}, \mathcal{D}_{real})$. In practice, we can minimize the empirical loss $d_{\mathcal{F},\phi}(\widehat{\mathcal{D}}_{G_w}, \widehat{\mathcal{D}}_{real})$ to approximate $d_{\mathcal{F},\phi}(\widehat{\mathcal{D}}_{G_w}, \mathcal{D}_{real})$. First, we have the following generalization bound on $\widehat{\mathcal{D}}_{real}$ to develop the generalization analysis of LCC-GANs.

Theorem 1 *Suppose $\phi(\cdot)$ is Lipschitz smooth: $|\phi'(\cdot)| \le L_\phi$, and bounded in $[-\Delta, \Delta]$. Given the coordinate coding $(\gamma, \mathcal{C})$, an example set $\mathcal{H}$ in latent space and the empirical distribution $\widehat{\mathcal{D}}_{real}$, if the generator is Lipschitz smooth, then the expected generalization error satisfies:*

$$\begin{aligned}&\mathbb{E}_{\mathcal{H}}\left[d_{\mathcal{F},\phi}\left(\widehat{\mathcal{D}}_{G_{\widehat{w}}(\gamma(\mathbf{h}))}, \widehat{\mathcal{D}}_{real}\right)\right]\\ \le&\inf_{\mathcal{G}} \mathbb{E}_{\mathcal{H}}\left[d_{\mathcal{F},\phi}\left(\mathcal{D}_{G_u(\mathbf{h})}, \widehat{\mathcal{D}}_{real}\right)\right] + \epsilon(d_{\mathcal{M}}),\end{aligned}$$

where $\epsilon(d_{\mathcal{M}}) = L_\phi Q_{L_{\mathbf{h}},L_G}(\gamma, \mathcal{C}) + 2\Delta$, and generative quality $Q_{L_{\mathbf{h}},L_G}(\gamma, \mathcal{C})$ has an upper bound w.r.t. $d_{\mathcal{M}}$ in Lemma 3 which is given in supplementary materials.

See supplementary materials for the proof.

Theorem 1 shows that the generalization bound for $\widehat{\mathcal{D}}_{real}$ is related to the dimension of the latent manifold (*i.e.*, $d_{\mathcal{M}}$) rather than the dimension of the latent space (*i.e.*, d_B). Based on Theorem 1 and the Rademacher complexity (Bartlett & Mendelson, 2002), we then accomplish the generalization bound on an unknown real distribution $\mathcal{D}_{real}$.

Theorem 2 *Under the condition of Theorem 1, given an empirical distribution $\widehat{\mathcal{D}}_{real}$ with N samples drawn from $\mathcal{D}_{real}$, the following holds with probability at least $1-\delta$,*

$$\begin{aligned}&\left|\mathbb{E}_{\mathcal{H}}\left[d_{\mathcal{F},\phi}\left(\widehat{\mathcal{D}}_{G_{\widehat{w}}}, \mathcal{D}_{real}\right)\right] - \inf_{\mathcal{G}}\mathbb{E}_{\mathcal{H}}\left[d_{\mathcal{F},\phi}\left(\mathcal{D}_{G_u}, \mathcal{D}_{real}\right)\right]\right|\\ \le&2R_{\mathcal{X}}(\mathcal{F}) + 2\Delta\sqrt{\frac{2}{N}\log(\frac{1}{\delta})} + 2\epsilon(d_{\mathcal{M}}),\end{aligned}$$

where $R_{\mathcal{X}}(\mathcal{F})$ is the Rademacher complexity of $\mathcal{F}$.

See supplementary materials for the proof.

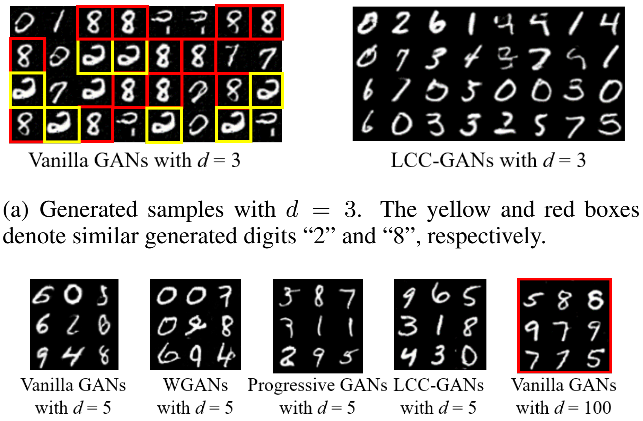
Theorem 2 shows that the generalization error of LCC-GANs can be bounded by Rademacher complexity of $\mathcal{F}$ and an error term $\epsilon(d_{\mathcal{M}})$. Specifically, the former term $R_{\mathcal{X}}(\mathcal{F})$ implies that the set of discriminator should be smaller to have better generalization ability, and also be large enough to be able to identify the data distribution, which is consistent with (Zhang et al., 2018). The latter term $\epsilon(d_{\mathcal{M}})$ indicates that a small dimensional input is sufficient to achieve good generalization. In practice, every dataset has its own dimension of the latent manifold. Nevertheless, experiments show that the proposed method is able to generate perceptually convincing images with small dimensional inputs.

6. Experiments

We compare LCC-GANs with several state-of-the-arts, *i.e.*, Vanilla GANs (Goodfellow et al., 2014), WGANs (Arjovsky et al., 2017) and Progressive GANs (Karras et al., 2018). Here, Vanilla GANs and Progressive GANs are used to implement our LCC-GANs. For all considered GAN methods, the inputs are sampled from a d-dimensional prior distribution, and we train the generative models to produce 64×64 images. All experiments are conducted on a single Nvidia Titan X GPU.

Implementation details. We implement LCC-GANs based on PyTorch.[1] We follow the experimental settings in DCGANs (Radford et al., 2015). Specifically, for the optimization, we use Adam (Kingma & Ba, 2015) with a mini-batch size of 64 and a learning rate of 0.0002 to train the generator and the discriminator. We initialize the parameters of both the generator and the discriminator following the strategy in (He et al., 2015).

Datasets and evaluation metrics. To thoroughly evaluate the proposed method, we conduct experiments on a wide variety of benchmark datasets, including MNIST (LeCun et al., 1998), Oxford-102 (Nilsback & Zisserman, 2008), LSUN (Yu et al., 2015) and CelebA (Liu et al., 2015). For quantitative comparisons, we adopt the *Inception Score* (IS) (Salimans et al., 2016) and *Multi-Scale Structural Similarity* (MS-SSIM) (Karras et al., 2018) as the performance metrics, which are highly consistent with human evaluations. Inception Score measures both the single image quality and the diversity over a large number of samples (*i.e.*, 50k). In general, a larger IS value corresponds to the better performance of the method, and a smaller MS-SSIM value corresponds to images with more diversity.


Vanilla GANs with $d = 3$
LCC-GANs with $d = 3$

(a) Generated samples with $d = 3$. The yellow and red boxes denote similar generated digits “2” and “8”, respectively.

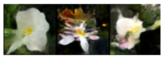
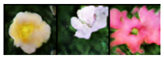
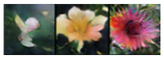
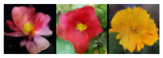
Vanilla GANs with $d = 5$
WGANs with $d = 5$
Progressive GANs with $d = 5$
LCC-GANs with $d = 5$
Vanilla GANs with $d = 100$

(b) Comparisons of different GANs with $d = 5$, where GANs with $d = 100$ are considered as the baseline.

Figure 4. Performance comparisons of various GANs on MNIST.

[1] PyTorch is from http://pytorch.org/.

Table 1. Inception scores of various generative models on Oxford-102. For each method, we produce $50,000$ samples for testing.

Samples					
Methods	Vanilla GANs (d=10)	WGANs (d=10)	Progressive GANs (d=10)	Vanilla GANs (d=100)	LCC-GANs (d=10)
Scores	2.21 ± 0.03	2.14 ± 0.02	2.43 ± 0.05	2.66 ± 0.03	2.71 ± 0.03

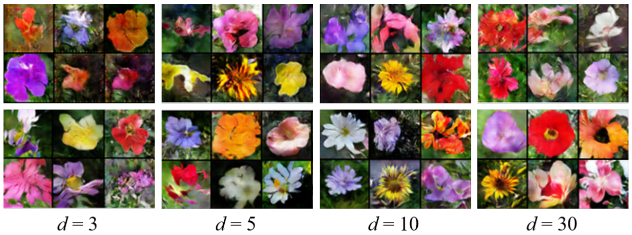


Figure 5. Results of LCC-GANs and Vanilla GANs on Oxford-102. Top: Vanilla GANs. Bottom: LCC-GANs.

6.1. Results on MNIST

In this experiment, we evaluate the performance of the proposed method on MNIST (LeCun et al., 1998), which contains handwritten digit images ranging from 0 to 9. In this small dataset, we adopt Vanilla GANs as the baseline to implement the proposed LCC-GANs. The visual comparisons are shown in Figure 4.

From Figure 4(a), given a very low dimensional input with $d = 3$, Vanilla GANs produce only few kinds of digits with almost the same shapes (see the yellow and red boxes in Figure 4(a)). In other words, Vanilla GANs produce images with very low diversity. In contrast, LCC-GANs with a small dimensional input $d = 3$ can produce digits with different styles and different orientations. Equipped with LCC, the proposed LCC-GANs effectively preserve the local information of data on the latent manifold and thus help the training of GANs.

In Figure 4(b), we increase the dimension of input to $d = 5$ and compare the proposed LCC-GANs with other state-of-the-art GAN methods. In this experiment, the baseline GAN methods often produce digits with obscure structure. Nevertheless, the proposed LCC-GANs significantly outperform the considered baseline methods and produce sharp images with high diversity. More critically, LCC-GANs with $d = 5$ are able to achieve comparable or even better performance than their GAN counterparts with $d = 100$ (see the red box in Figure 4(b)). These results show the effectiveness of the proposed LCC-GANs when training a generative model with the local information of the latent manifold. Compared to the baseline methods, LCC-GANs only need a relatively low dimensional input to produce visually promising images.

Table 2. Inception-Score (IS) and MS-SSIM on Oxford-102.

Methods	$d = 5$		$d = 10$		$d = 30$		$d = 100$	
	IS	SSIM	IS	SSIM	IS	SSIM	IS	SSIM
Vanilla GANs	2.03	0.205	2.37	0.180	2.57	0.166	2.66	0.160
VAE	2.14	0.203	2.38	0.185	2.54	0.163	2.68	0.162
Sparse Coding	2.44	0.197	2.63	0.179	2.68	0.157	2.72	0.153
LCC Coding	2.57	0.188	2.71	0.163	2.83	0.153	2.75	0.147

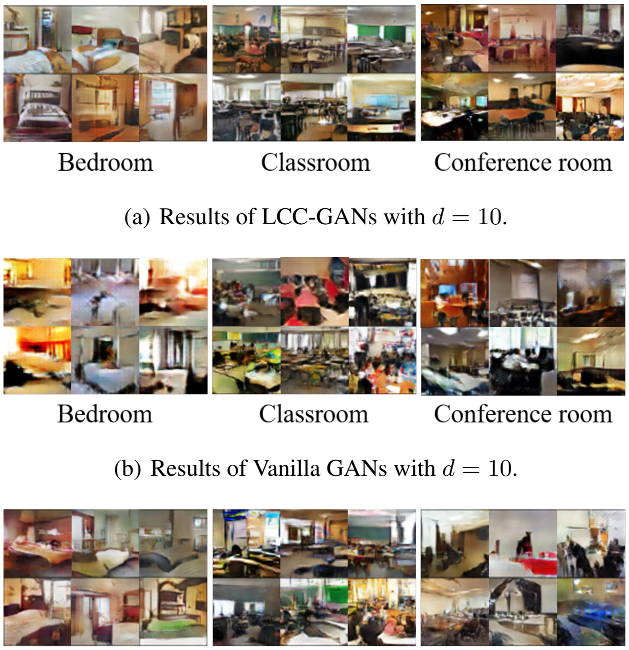
6.2. Results on Oxford-102 Flowers

We further evaluate the proposed LCC-GANs on a larger dataset Oxford-102 which contains flower images of 102 categories. In this experiment, we adjust the input of generative models with different dimensions, *i.e.*, $d = \{3, 5, 10, 30\}$, and adopt Vanilla GANs to implement the proposed LCC-GANs and investigate the effect of different input dimensions. The results are shown in Figure 5.

From Figure 5, we have the following observations. First, for Vanilla GANs, the performance highly depends on the input dimension. Given a small dimension, *i.e.*, $d = 3$ or $d = 5$, Vanilla GANs often fail to produce meaningful flowers and obtain images with a blurring structure and distorted regions. In contrast, the proposed LCC-GANs are able to produce promising images with clear structure given an input with $d = 5$. With such a low dimensional input, LCC-GANs effectively capture the local information of the latent manifold and produce perceptually convincing images. Second, we further investigate the effect of input dimension. From Figure 5, the proposed LCC-GANs consistently outperform Vanilla GANs given the inputs of different dimensions.

Moreover, we compare the proposed LCC-GANs with several state-of-the-art GAN methods and report the results in Table 1. From Table 1, the proposed LCC-GANs with $d = 10$ significantly outperform the other baseline methods and achieve the best performance with a score of 2.71. More critically, LCC-GANs with $d = 10$ achieve even better performance than Vanilla GANs with $d = 100$, which require the input with much higher dimension.

Comparisons of different representation methods. On Oxford-102, we compare different representation methods and adopt Inception Score and MS-SSIM to evaluate the quality and diversity of the generated images, respectively. We adjust the input with different dimensions, *i.e.*, $d = \{5, 10, 30, 100\}$, and adopt Vanilla GANs to implement LCC-GANs. The results are shown in Table 2.

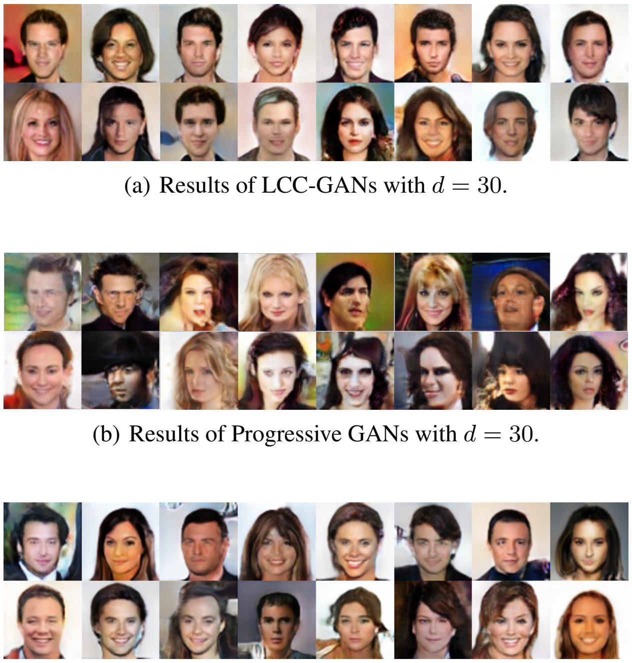


(a) Results of LCC-GANs with $d = 10$.

(b) Results of Vanilla GANs with $d = 10$.

(c) Results of Vanilla GANs with $d = 100$.

Figure 6. Results of LCC-GANs with Vanilla GANs for different dimensions of the latent distribution on LSUN.



(a) Results of LCC-GANs with $d = 30$.

(b) Results of Progressive GANs with $d = 30$.

(c) Results of Progressive GANs with $d = 100$.

Figure 7. Performance comparisons of LCC-GANs with Progressive GANs.

From Table 2, LCC-GANs consistently outperform other methods with various d in both measures. These results show the effectiveness of the proposed LCC-GANs in producing perceptually promising images with higher quality and larger diversity than the considered baselines.

6.3. Results on LSUN

In this experiment, we evaluate the proposed LCC-GANs on LSUN which is a collection of natural images of indoor scenes. We train the generative models to produce images of different categories, including bedroom, classroom, conference room, etc. In this experiment, we also adopt Vanilla GANs as the baseline models to implement LCC-GANs. We show the visual comparison results in Figure 6.

From Figure 6, when we train the models using an input with a small dimension $d = 10$, Vanilla GANs often fail to generate clear and meaningful images. In contrast, LCC-GANs significantly outperform their GAN counterparts and produce images with sharp structure and rich details. Moreover, when generating images of different scenes, LCC-GANs consistently outperform Vanilla GANs. Note that the scene images in LSUN are much more complex than the images of MNIST and Oxford-102. Therefore, training a generative model can be more difficult. However, with the help of LCC, the proposed LCC-GANs are able to effectively capture the local common features and produce visually convincing images.

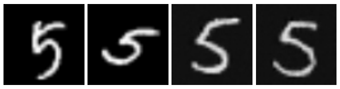
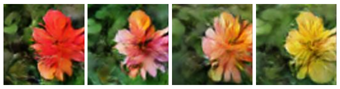
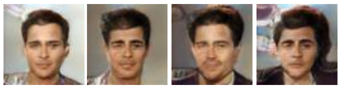
In this experiment, we also present the generated samples of Vanilla GANs with a high dimensional input $d = 100$. Compared to this method, LCC-GANs only require an input with $d = 10$ to produce even better images. In other words, this LCC sampled input effectively preserves the local information of real images on the latent manifold and thus helps the training of GANs. With the help of LCC sampling, most of the generated images show sharper structure and contain more meaningful details.

6.4. Results on CelebA

In this experiment, we evaluate the proposed method on the large-scale dataset CelebA, which is composed of a set of celebrity faces. Here, Progressive GANs (Karras et al., 2018) are adopted to implement LCC-GANs. We conduct comparisons and show the results in Figure 7.

Since face images often share a common face outline and only differ in detailed attributes, *e.g.*, hair, eyes, mouth, skin features, it requires an input with a larger dimension to capture the local information. In this way, we adopt the input with a larger dimension for both Progressive GANs and the proposed LCC-GANs in the training. From Figure 7, the performance of Progressive GANs degrades severely given an input with a small dimension $d = 30$, compared

Table 3. Generated images from LCC sampling on MNIST, Oxford-102 and CelebA. The last column shows the most similar images in training set to the generated samples on the left.

Datasets	Generated Samples	Nearest Real data
MNIST		
Oxford-102		
CelebA		

to $d = 100$. However, with the help of LCC, the proposed LCC-GANs with the input of $d = 30$ are able to produce images of better quality than Progressive GANs with high dimensional inputs of $d = 100$. According to these results, LCC-GANs greatly benefit from the LCC sampling and make the training much easier than directly matching the standard Gaussian distribution.

6.5. Demonstration of LCC Sampling

In this experiment, we investigate the effectiveness of the proposed LCC sampling method. To achieve this, we can simply fix a specific set of bases and only change the corresponding weights to produce images. Ideally, these images should be located in a local area of the latent manifold and share some common features.

We conduct experiments on LCC sampling and show the results in Table 3. The second column of Table 3 shows the generated images sampled by LCC sampling method on different datasets. The last column is the real image with the largest similarity to the generated images. From Table 3, LCC-GANs produce digits with sharp shapes and different orientations or styles (see the top row in Table 3). Each generated image contains a digit “5” but with obvious individual differences. In other words, the proposed LCC sampling method is able to generate new data by effectively exploiting the local information on the latent manifold.

When synthesizing flowers and faces, we draw a similar conclusion that verifies the effectiveness of the proposed LCC sampling method. Specifically, LCC-GANs produce flowers with similar shapes but with different colors. Similarly, LCC-GANs also produce varying face images of promising quality which share some common features. These results demonstrate that the proposed LCC-GANs generalize well to unseen data rather than simply memorizing the training samples.

Table 4. MS-SSIM on different datasets. Here, $d = 100$ is for all baselines and $d = 30$ for LCC-GANs.

Methods	MNIST	Oxford-102	LSUN	CelebA
Vanilla GANs	0.242	0.160	0.224	0.337
WGANs	0.251	0.157	0.237	0.324
Progressive GANs	0.239	0.151	0.213	0.308
LCC-GANs	0.224	0.153	0.203	0.305

6.6. More Quantitative Results

In this experiment, we adopt MS-SSIM as the evaluation measure and compare the proposed LCC-GANs with several GAN methods on four benchmark datasets. We use Vanilla GANs to implement LCC-GANs. To show the superiority of the proposed method, we set $d = 30$ for LCC-GANs and $d = 100$ for the other baselines. Here, we can only report MS-SSIM because Inception Score is no longer a valid measure and may give misleading results on CelebA (Barratt & Sharma, 2018). The quantitative results are shown in Table 4.

From Table 4, with a low dimensional input, the proposed method is able to produce images with larger or comparable diversity (smaller MS-SSIM score) than the considered baselines with high dimensional inputs on most datasets. These results show the effectiveness of the proposed LCC-GANs in generating images with large diversity.

7. Conclusions

In this paper, we have proposed a novel generative model by exploiting the local information on the latent manifold of real data to improve GANs using Local Coordinate Coding (LCC). Unlike existing methods, based on a generator approximation, we have developed an LCC based sampling method to train GANs. In this way, we are able to conduct analysis on the generalization ability of GANs and theoretically prove that a small dimensional input will help to achieve good generalization. Extensive experiments on several benchmark datasets demonstrate the superiority of the proposed method over the state-of-the-art methods. Specifically, with the proposed LCC sampling, the proposed method outperforms the considered baselines by producing sharper images with higher diversity.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) 61502177 and 61602185, and Recruitment Program for Young Professionals, and Guangdong Provincial Scientific and Technological funds 2017B090901008, 2017A010101011, 2017B090910005, and Fundamental Research Funds for the Central Universities D2172500, D2172480, and Pearl River S&T Nova Program of Guangzhou 201806010081 and CCF-Tencent Open Research Fund RAGR20170105.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. Generalization and equilibrium in generative adversarial nets (GANs). In *International Conference on Machine Learning*, volume 70, pp. 224–232, 2017.

Barratt, S. and Sharma, R. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, pp. 1026–1034, 2015.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *International Conference on Computer Vision*, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Kim, T., Cha, M., Kim, H., Lee, J., and Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, pp. 3730–3738, 2015.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Mathieu, M., Couprie, C., and LeCun, Y. Deep Multi-scale Video Prediction beyond Mean Square Error. In *International Conference on Learning Representations*, 2016.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (language) Modeling: a Baseline for Generative Models of Natural Videos. *arXiv preprint arXiv:1412.6604*, 2014.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290 (5500):2323–2326, 2000.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Tolstikhin, I., Olivier, B., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Yu, K., Zhang, T., and Gong, Y. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems*, pp. 2223–2231, 2009.

Zhang, P., Liu, Q., Zhou, D., Xu, T., and He, X. On the discrimination-generalization tradeoff in GANs. In *International Conference on Learning Representations*, 2018.