Global versus Localized Generative Adversarial Nets

Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki
guojun.qi@ucf.edu, {lihengzhang1993,hao_hu,m.edraki}@knights.ucf.edu
Jingdong Wang, and Xian-Sheng Hua
welleast@outlook.com, xiansheng.hxs@alibaba-inc.com
Laboratory for MAchine Perception and LEarning (MAPLE)
maple.cs.ucf.edu
University of Central Florida and Alibaba Group

Abstract

In this paper, we present to learn a novel localized Generative Adversarial Net (GAN) on the manifold of real data. Compared with the classic GAN that globally *parameterizes a manifold, the Localized GAN (LGAN) uses local coordinate charts to parameterize local geometry of data transformations across different locations on the manifold. Specificall , around each point there exists a* local *generator to produce diverse data following various patterns of transformations along the manifold. The locality nature of LGAN enables it to directly access the local geometry with no need to invert the generator in the classic GAN to access its global coordinates. Furthermore, it can prevent the manifold from being locally collapsed to be dimensionally deficien by imposing an orthonormality prior between tangents. This provides a geometric approach to alleviating mode collapse on the manifold at least locally by preventing vanishing or dependent data variations along different coordinates. We will also demonstrate the LGAN can be applied to train a locally consistent classifie that is robust against perturbations along the manifold, and the resultant regularizer is closely related to the Laplace-Beltrami operator without relying on an approximate graph-based manifold representation. Our experiments show that the proposed LGANs can not only produce diverse image transformations, but also deliver superior classificatio performances.*

1. Introduction

The classic Generative Adversarial Net (GAN) [**?**] seeks to generate samples with indistinguishable distributions from real data. For this purpose, it learns a generator $G(\mathbf{z})$ as a function that maps from input random noises $\mathbf{z}$ drawn from a distribution $P_{\mathcal{Z}}$ to output data $G(\mathbf{z})$. A discriminator is learned to distinguish between real and generated samples. The generator and discriminator are jointly trained in an adversarial fashion so that the generator fools the discriminator by improving the quality of generated data.

All the samples produced by the learned generator form a manifold $\mathcal{M} = \{G(\mathbf{z})|\mathbf{z} \sim P_{\mathcal{Z}}\}$, with the input variables $\mathbf{z}$ as its global coordinates. However, a global coordinate system could be too restrictive to capture various forms of local transformations on the manifold. For example, a non-rigid object like human body and a rigid object like a car admit different forms of variations on their shapes and appearances, resulting in distinct geometric structures unfi into a single coordinate chart of image transformations.

Indeed, existence of a global coordinate system is a too strong assumption for many manifolds. For example, there does not exist a global coordinate chart covering an entire hyper-sphere embedded in a high dimensional space as it is even not topologically similar (i.e., homeomorphic) to an Euclidean space. This prohibits the existence of a global isomorphism between a single coordinate space and the hyper-sphere, making it impossible to study the underlying geometry in a global coordinate system. For this reason, mathematicians instead use an atlas of *local* coordinate charts located at different points on a manifold to study the underlying geometry [**?**].

Even when a global coordinate chart exists, a global GAN could still suffer two serious challenges. First, a point $\mathbf{x}$ on manifold cannot be directly mapped back to its global coordinates $\mathbf{z}$, i.e., findin $\mathbf{z}$ such as $G(\mathbf{z}) = \mathbf{x}$ for a given $\mathbf{x}$. But many applications need the coordinates of a given point $\mathbf{x}$ to access its local geometry such as tangents and curvatures. Thus, for a global GAN, one has to solve the inverse G^{-1} of a generator network (e.g., via an autoencoder such as VAE [**?**], ALI [**?**] and BiGAN [**?**]) to access the coordinates of a point $\mathbf{x}$ and then its local geometry of data transformations along the manifold.

Figure 1. Illustration of a curved manifold $\mathcal{M}$ embedded in 3-dimensional ambient space. At each location $\mathbf{x}$, its tangent space $\mathcal{T}_{\mathbf{x}}$ consists of all tangent vectors to the manifold. These tangent vectors characterize the geometry of local transformations allowed to move a point $\mathbf{x}$ on $\mathcal{M}$.

The other problem is the manifold generated by a global GAN could locally collapse. Geometrically, on a N-dimensional manifold, this occurs if the tangent space $\mathcal{T}_{\mathbf{x}}$ of a point $\mathbf{x}$ is dimensionally deficient i.e., $\dim \mathcal{T}_{\mathbf{x}} < N$ when tangents become linearly dependent along some coordinates [1]. In this case, data variations become redundant or even vanish along some directions on the manifold. Moreover, a locally collapsed tangent space at a point $\mathbf{x}$ could be related with a *collapsed mode* [**?**, **?**], around which a generator $G(\mathbf{z})$ would no longer produce diverse data as $\mathbf{z}$ changes in different directions. This provides us with an alternative geometric insight into mode collapse phenomena observed in literature [**?**].

The above challenges inspire us to develop a Localized GAN (LGAN) by learning local generators $G(\mathbf{x}, \mathbf{z})$ associated with individual points $\mathbf{x}$ [2] on a manifold. As illustrated in Figure **??**, local generators are located around different data points so that the pieces of data generated by different local generators can be sewed together to cover an entire manifold seamlessly. Different pieces of generated data are not isolated but could have some overlaps between each other to form a connected manifold [**?**].

The advantage of the LGAN is at least twofold. First, one can directly access the local geometry of transformations near a point without having to evaluate its global coordinates, as each point is directly localized by a local generator in the corresponding local coordinate chart. This *locality* nature of LGAN makes it straightforward to explore pointwise geometric properties across a manifold. Moreover, we will impose an orthonormality prior on the local tangents, and the resultant orthonormal basis spans a full dimensional tangent space, preventing a manifold from being locally collapsed. It allows the model to explore diverse patterns of data transformations disentangled in different directions, leading to a geometric approach at least *locally* alleviating the mode collapse problem on a manifold.

We will also demonstrate an application of the LGAN to train a robust classifie by encouraging a smooth change of the classificatio decision on the manifold formed by the LGAN. The classifie is trained with a regularizer that minimizes the square norm of the classifier s gradient on the manifold, which is closely related with Laplace-Beltrami operator. The local coordinate representation in LGAN makes it straightforward to train such a classifie with no need of computing global coordinates of training examples to access their local geometry of transformations. Moreover, the learned orthonormal tangent basis also allows the model to effectively explore various forms of independent transformations allowed on the underlying manifold.

The remainder of this paper is organized as follows. In the next section, we will review the related works, followed by Section **??** in which we present the proposed Localized GANs. In Section **??**, a semi-supervised learning algorithm is presented to use LGANs to train a robust classifie that is locally consistent over the manifold formed by a LGAN model.

2. Related Works

Global vs. Localized GANs. By different types of coordinate systems used to parameterize their data manifolds, we can categorize the GANs into *global* and *local* models. Existing models, including the seminal GAN model proposed by Goodfellow et al. [**?**, **?**] and many variants [**?**], are *global* GANs that use a global coordinate chart to parameterize the generated data. In contrast, the localized GAN presented in this paper is a local paradigm, which uses local coordinate charts centered at different data points to form a manifold by a collection of local generators.

The distinction between global and local coordinate systems results in conceptual and algorithmic differences between global and local GANs. Conceptually, the global GANs assume that the manifolds formed by their generators could be globally parameterized in a way that the manifolds are topologically similar to an Euclidean coordinate space. In contrast, the localized paradigm abandons the global parameterizability assumption, allowing us to use multiple local coordinate charts to cover an entire manifold. Algorithmically, if a global GAN needs to access local geometric information underlying its generated manifold, it has to invert the generator function in order to fin the global coordinates corresponding to a given data point. This is usually performed by learning an auto-encoder network along with the GAN models, e.g., BiGAN [**?**] and ALI [**?**]. On the contrary, the localized GAN enables direct access to local geometry without having to invert a global generator, since

[1] For example, on a 2-D surface, the manifold reduces to an 1-D curve or a 0-D singularity at some points.

[2] At firs glance, the form of a local generator $G(\mathbf{x}, \mathbf{z})$ looks like a conditional GAN (cGAN) with $\mathbf{x}$ as its condition. However, a local generator in LGAN intrinsically differs from cGAN in its geometric representation of a local coordinate chart. Refer to Section **??** for details.

the local geometry around a point is directly available from the corresponding local generator. Moreover, the orthormornality between local tangents could also maximize the capability of a local GAN in exploring independent local transformations along different coordinate directions, thereby preventing the manifold of generated data from being locally collapsed with deficien dimensions.

Semi-Supervised Learning. One of the most important applications of GANs lies in the classificatio problem, especially considering their ability of modeling the manifold structures for both labeled and unlabeled examples [**?**, **?**, **?**]. For example, [**?**] presented variational auto-encoders [**?**] by combining deep generative models and approximate variational inference to explore both labeled and unlabeled data. [**?**] treated the samples from the GAN generator as a fake class, and explore unlabeled examples by assigning them to a real class different from the fake one. [**?**] proposed to train a ladder network [**?**] by minimizing the sum of supervised and unsupervised cost functions through back-propagation, which avoids the conventional layer-wise pre-training approach. [**?**] presented an approach to learning a discriminative classifie by trading-off mutual information between observed examples and their predicted classes against an adversarial generative model. [**?**] sought to jointly distinguish between not only real and generated samples but also their latent variables in an adversarial fashion. [**?**] presented to train a semi-supervised classifie by exploring the areas where real samples are unlikely to appear.

In this paper, we will explore the LGAN's ability of modeling the data distribution and its manifold geometry to train a robust classifie , which can make locally consistent classificatio decisions in presence of small perturbations on data. The idea of training a locally consistent classifie could trace back almost two decades ago to TangentProp [**?**] that pursued classificatio invariance against image rotation and translation manually performed in an ad-hoc fashion. Kumar et al. [**?**] extended the TangentProp by training an augmented form of BiGAN to explore the underlying data distributions, but it still relied on a global GAN to indirectly access the local tangents by learning a separate encoder network. On the contrary, the local coordinates will enable the LGAN to directly access the geometry of image transformations to train a *locally consistent* classifie , along with the orthonormality between local tangents allowing the learned classifie to explore its local consistency against independent image transformations along different local coordinates on the manifold.

3. Localized GANs

We present the proposed Localized GANs (LGANs). Before that, we firs briefl review the classic GANs in the context of differentiable manifolds.

3.1. Classic GAN and Global Coordinates

A Generative Adversarial Net (GAN) seeks to train a generator $G(\mathbf{z})$ by transforming a random noise $\mathbf{z} \in \mathbb{R}^N$ drawn from $P_{\mathcal{Z}}$ to a data sample $G(\mathbf{z}) \in \mathbb{R}^D$. Such a classic GAN uses a global N-dimensional coordinate system $\mathbf{z}$ to represent its generated samples $G(\mathbf{z})$ residing in an ambient space $\mathbb{R}^D$. Then all the generated samples form a N-dimensional manifold $\mathcal{M} = \{G(\mathbf{z})|\mathbf{z} \in \mathbb{R}^N\}$ that is embedded in $\mathbb{R}^D$.

In a global coordinate system, the local structure (e.g., tangent vectors and space) of a given data point $\mathbf{x}$ is not directly accessible, since one has to compute its corresponding coordinates $\mathbf{z}$ to localize the point on the manifold. One often has to resort to an inverse of the generator (e.g., via ALI and BiGAN) to fin the mapping from $\mathbf{x}$ back to $\mathbf{z}$.

Even worse, the tangent space $\mathcal{T}_{\mathbf{x}}$ could locally collapse at a point $\mathbf{x}$ if it is dimensionally deficien (i.e., $\dim \mathcal{T}_{\mathbf{x}} < N$). Actually, if $\dim \mathcal{T}_{\mathbf{x}}$ is extremely low (i.e., $<< N$), a locally collapsed point $\mathbf{x}$ could become a collapsed mode on the manifold, around which $G(\mathbf{z})$ would no longer produce significan data variations even though $\mathbf{z}$ changes in different directions. For example, if $\dim \mathcal{T}_{\mathbf{x}} = 1$, there is only a curve of data variations passing through $\mathbf{x}$. In an extreme case $\dim \mathcal{T}_{\mathbf{x}} = 0$, the data variations would completely vanish as $\mathbf{x}$ becomes a singular point on the manifold.

3.2. Local Generators and Tangent Spaces

Unlike the classic GAN, we propose a Localized GAN (LGAN) model equipped with a local generator $G(\mathbf{x}, \mathbf{z})$ that can produce various examples in the neighborhood of a point $\mathbf{x} \in \mathbb{R}^D$ on the manifold.

This forms a local coordinate chart $\{G(\mathbf{x}, \mathbf{z})|\mathbf{z} \subset \mathbb{R}^N \sim P_{\mathcal{Z}}\}$ around $\mathbf{x}$, with its local *coordinates* $\mathbf{z}$ drawn from a random distribution $P_{\mathcal{Z}}$ over an Euclidean space $\mathbb{R}^N$. In this manner, an atlas of local coordinate charts can cover an entire manifold $\mathcal{M}$ by a collection of local generators $G(\mathbf{x}, \mathbf{z})$ located at different points on $\mathcal{M}$.

In particular, for $G(\mathbf{x}, \mathbf{z})$, we assume that the origin of the local coordinates $\mathbf{z}$ should be located at the given point $\mathbf{x}$, i.e., $G(\mathbf{x}, \mathbf{0}) = \mathbf{x}$, where $\mathbf{0} \in \mathbb{R}^N$ is an all-zero vector.

To study the local geometry near a point $\mathbf{x}$, we need tangent vectors located at $\mathbf{x}$ on the manifold. By changing the value of a coordinate $\mathbf{z}^j$ while fixin the others, the points generated by $G(\mathbf{x}, \mathbf{z})$ form a coordinate curve passing through $\mathbf{x}$ on the manifold. Then, the vector tangent to this coordinate curve at $\mathbf{x}$ is

$$\boldsymbol{\tau}_{\mathbf{x}}^j \triangleq \frac{\partial G(\mathbf{x}, \mathbf{z})}{\partial \mathbf{z}^j}|_{\mathbf{z}=\mathbf{0}} \in \mathbb{R}^D. \tag{1}$$

All such N tangent vectors $\boldsymbol{\tau}_{\mathbf{x}}^j, j = 1, \cdots, N$ form a basis spanning a linear tangent space $\mathcal{T}_{\mathbf{x}} = \mathrm{Span}(\boldsymbol{\tau}_{\mathbf{x}}^1, \cdots, \boldsymbol{\tau}_{\mathbf{x}}^N)$ at $\mathbf{x}$. This tangent space consists of all vectors tangent to some curves passing through $\mathbf{x}$ on the

manifold. Each tangent $\boldsymbol{\tau} \in \mathcal{T}_{\mathbf{x}}$ characterizes some local transformation in the direction of this tangent vector.

A Jacobian matrix $\mathbf{J}_{\mathbf{x}} \in \mathbb{R}^{D \times N}$ can also be define by stacking all N tangent vectors $\boldsymbol{\tau}_{\mathbf{x}}^j$ in its columns.

3.3. Regularity: Locality and Orthonormality

However, there exists a challenge that the tangent space $\mathcal{T}_{\mathbf{x}}$ would collapse if it is dimensionally deficient i.e, its dimension $\dim \mathcal{T}_{\mathbf{x}}$ is smaller than the manifold dimension N. If this occurs, the N tangents in (**??**) could reduce to dependent transformations that would even vanish along some coordinates $\mathbf{z}$.

To prevent the collapse of the tangent space, we need to impose a regularity condition that the N basis $\{\boldsymbol{\tau}_{\mathbf{x}}^j, j = 1, \cdots, N\}$ of $\mathcal{T}_{\mathbf{x}}$ should be linearly independent of each other. This guarantees the manifold be locally "similar" (diffeomorphic mathematically) to a N-dimensional Euclidean space, rather than being collapsed to a lower-dimensional subspace having dependent local coordinates.

As a linearly independent basis can always be transformed to an orthonormal counterpart by a proper transformation, one can set the orthonormal condition on the tangent vectors $\boldsymbol{\tau}_{\mathbf{x}}^j$, i.e.,

$$\langle \boldsymbol{\tau}_{\mathbf{x}}^i, \boldsymbol{\tau}_{\mathbf{x}}^j \rangle = \delta_{ij} \tag{2}$$

where $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ii} = 1$ otherwise. The resultant orthonormal basis of tangent vectors capture the independent components of local transformations near individual data points on the manifold.

In summary, the local generator $G(\mathbf{x}, \mathbf{z})$ should satisfy the following two conditions:
(i) locality: $G(\mathbf{x}, \mathbf{0}) = \mathbf{x}$, i.e., the origin of the local coordinates $\mathbf{z}$ should be located at $\mathbf{x}$;
(ii) orthonormality: $\mathbf{J}_{\mathbf{x}}^T \mathbf{J}_{\mathbf{x}} = \mathbf{I}_N$, which is a matrix form of (**??**) with I_N being the identity matrix of size N.

One can minimize the following regularizer on $G(\mathbf{x}, \mathbf{z})$ to penalize the violation of these two conditions [3],

$$\Omega_G(\mathbf{x}) = \mu \|G(\mathbf{x}, \mathbf{0}) - \mathbf{x}\|^2 + \eta \|\mathbf{J}_{\mathbf{x}}^T \mathbf{J}_{\mathbf{x}} - \mathbf{I}_N\|^2 \tag{3}$$

where μ and η are nonnegative weighting coefficient for the two terms. By using a deep network for computing $G(\mathbf{x}, \mathbf{z})$, this regularizer can be minimized by backpropagation algorithm.

3.4. Training $G(\mathbf{x}, \mathbf{z})$

Now the learning problem for the localized GANs boils down to train a $G(\mathbf{x}, \mathbf{z})$. Like the GANs, we will train a discriminator $D(\mathbf{x})$ to distinguish between real samples drawn from a data distribution $P_{\mathcal{X}}$ and generated samples by $G(\mathbf{x}, \mathbf{z})$ with $\mathbf{x} \sim P_{\mathcal{X}}$ and $\mathbf{z} \sim P_{\mathcal{Z}}$ as follows.

$$\max_D \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}} \log D(\mathbf{x}) + \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}, \mathbf{z} \sim P_{\mathcal{Z}}} \log(1 - D(G(\mathbf{x}, \mathbf{z}))$$

where $D(\mathbf{x})$ is the probability of $\mathbf{x}$ being real, and the maximization is performed wrt the model parameters of discriminator D.

On the other hand, the generator can be trained by maximizing the likelihood that the generated samples by $G(\mathbf{x}, \mathbf{z})$ are real as well as minimizing the regularization term (**??**).

$$\min_G -\mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}, \mathbf{z} \sim P_{\mathcal{Z}}} \log D(G(\mathbf{x}, \mathbf{z})) + \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}} \Omega_G(\mathbf{x})$$

where the minimization is performed wrt the model parameters of local generator G, and the regularization enforces the locality and orthonormality conditions on G.

Then D and G can be alternately optimized by stochastic gradient descent via a backpropagation algorithm.

4. Semi-Supervised LGANs

In this section, we will show that the LGAN can help us train a locally consistent classifie by exploring the manifold geometry. First we will discuss the functional gradient on a manifold in Section **??**, and show its connection with Laplace-Beltrami operator that generalizes the graph Laplacian in Section **??**. Finally, we will present the proposed LGAN-based classifie in detail in Section **??**.

4.1. Functional Gradient along Manifold

First let us discuss how to calculate the derive of a function on the manifold.

Consider a function $f(\mathbf{x})$ define on the manifold. At a given point $\mathbf{x}$, its neighborhood on the manifold is depicted by $G(\mathbf{x}, \mathbf{z})$ with the local coordinates $\mathbf{z}$. By viewing f as a function of $\mathbf{z}$, we can compute the derivative of f when it is restricted on the manifold.

It is not hard to obtain the derivative of $f(G(\mathbf{x}, \mathbf{z}))$ with respect to a coordinate $\mathbf{z}^j$ by the chain rule,

$$\frac{\partial f(G(\mathbf{x}, \mathbf{z}))}{\partial \mathbf{z}^j}|_{\mathbf{z}=\mathbf{0}} = \langle \boldsymbol{\tau}_{\mathbf{x}}^j, \nabla_{\mathbf{x}} f(\mathbf{x}) \rangle$$

where $\nabla_{\mathbf{x}} f(\mathbf{x})$ is the gradient of f at $\mathbf{x}$, and $\langle \cdot, \cdot \rangle$ is the inner product between two vectors. It depicts how fast f changes as a point moves away from $\mathbf{x}$ along the coordinate $\mathbf{z}^j$ on the manifold.

Then, the gradient of f at $\mathbf{x}$ when f is restricted on the manifold $G(\mathbf{x}, \mathbf{z})$ can be written as

$$\nabla_{\mathbf{x}}^G f \triangleq \nabla_{\mathbf{z}} f(G(\mathbf{x}, \mathbf{z}))|_{\mathbf{z}=\mathbf{0}} = \mathbf{J}_{\mathbf{x}}^T \nabla_{\mathbf{x}} f(\mathbf{x}) \tag{4}$$

Geometrically, it shows the gradient of f along the manifold can be obtained by projecting the regular gradient $\nabla_{\mathbf{x}} f$ onto

[3] Alternatively, we can parameterize $G(\mathbf{x}, \mathbf{z})$ as $\mathbf{x} + B(\mathbf{x}, \mathbf{z}) - B(\mathbf{x}, \mathbf{0})$ with a network B modeling a perturbation on $\mathbf{x}$. Such a parameterization of local generator directly satisfie the locality constraint $G(\mathbf{x}, \mathbf{0}) = \mathbf{x}$.

the tangent space $\mathcal{T}_{\mathbf{x}}$ with the Jacobian matrix $\mathbf{J}_{\mathbf{x}}$. Here we denote the resultant gradient *along manifold* by $\nabla_{\mathbf{x}}^{G} f$ to highlight its dependency on $G(\mathbf{x}, \mathbf{z})$

4.2. Connection with Laplace-Beltrami Operator

If f is a classifie , $\nabla_{\mathbf{z}} f(G(\mathbf{x}, \mathbf{z}))$ depicts the change of the classificatio decision on the manifold formed by $G(\mathbf{x}, \mathbf{z})$. At $\mathbf{x}$, the change of f restricted on $G(\mathbf{x}, \mathbf{z})$ can be written as

$$|f(G(\mathbf{x}, \mathbf{z}+\delta \mathbf{z}))-f(G(\mathbf{x}, \mathbf{z}))|^{2} \approx\|\nabla_{\mathbf{x}}^{G} f\|^{2} \delta \mathbf{z} \tag{5}$$

It shows that penalizing $\|\nabla_{\mathbf{x}}^{G} f\|^{2}$ can train a robust classifie that is resilient against a small perturbation $\delta \mathbf{z}$ on a manifold. It is supposed to deliver locally consistent classificatio results in presence of noises.

The functional gradient is closely related with the Laplace-Beltrami operator, the one that is widely used as a regularizer on the graph-based semi-supervised learning [**?**, **?**, **?**, **?**].

It is well known that the divergence operator $-\operatorname{div}$ and the gradient ∇ are formally adjoint, i.e., $\int_{\mathcal{M}}\langle\mathbf{V}, \nabla_{\mathbf{x}}^{G} f\rangle d P_{\mathcal{X}}=\int_{\mathcal{M}} \operatorname{div}(\mathbf{V}) f d P_{\mathcal{X}}$. Thus we have

$$\int_{\mathcal{M}}\|\nabla_{\mathbf{x}}^{G} f\|^{2} d P_{\mathcal{X}}=\int_{\mathcal{M}} f \operatorname{div}(\nabla_{\mathbf{x}}^{G} f) d P_{\mathcal{X}} \tag{6}$$

where $\Delta f \triangleq \operatorname{div}(\nabla_{\mathbf{x}}^{G} f)$ is the Laplace-Beltrami operator.

In graph-based semi-supervised learning, one constructs a graph representation of data points to approximate the underlying data manifold [**?**], and then use a Laplacian matrix to approximate the Laplace-Beltrami operator Δf.

In contrast, with the help of LGAN, we can directly obtain Δf on $G(\mathbf{x}, \mathbf{z})$ without having to resort to a graph representation. Actually, as the tangent space at a point $\mathbf{x}$ has an orthonormal basis, we can write

$$\Delta f=\operatorname{div}(\nabla_{\mathbf{x}}^{G} f)=\sum_{j=1}^{N} \frac{\partial^{2} f(G(\mathbf{x}, \mathbf{z}))}{\partial(\mathbf{z}^{i})^{2}} \tag{7}$$

In the following, we will learn a locally consistent classifie on the manifold by penalizing a sudden change of its classificatio function f in the neighborhood of a point. We can implement it by minimizing either the square norm of the gradient or the related Laplace-Beltrami operator. For simplicity, we will choose to penalize the gradient of the classifie as it only involves computing the first order derivatives of a function compared with the Laplace-Beltrami operator having the higher-order derivatives.

4.3. Locally Consistent Semi-Supervised Classifie

We consider a semi-supervised learning problem with a set of training examples $(\mathbf{x}_{l}, y_{l})$ drawn from a distribution $P_{\mathcal{L}}$ of labeled data. We also have some unlabeled examples $\mathbf{x}_{u}$ drawn from the data distribution $P_{\mathcal{X}}$ of real samples. The amount of unlabeled examples is often much larger than their labeled counterparts, and thus can provide useful information for training G to capture the manifold structure of real data.

Suppose that there are K classes, and we attempt to train a classifie $P(y|\mathbf{x})$ for $y \in\{1,2, \cdots, K+1\}$ that outputs the probability of $\mathbf{x}$ being assigned to a class y [**?**]. The firs K are real classes and the last one is a fake class denoting $\mathbf{x}$ is a generated example.

This probabilistic classifie can be trained by the following objective function

$$\begin{aligned}\max_{P} & \mathbb{E}_{(\mathbf{x}_{l}, y_{l}) \sim P_{\mathcal{L}}} \log P(y_{l}|\mathbf{x}_{l})+\mathbb{E}_{\mathbf{x}_{u} \sim P_{\mathcal{X}}} \log P(y_{u} \leq K|\mathbf{x}_{u}) \\ & +\mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}, \mathbf{z} \sim P_{\mathcal{Z}}} \log P(y=K+1|G(\mathbf{x}, \mathbf{z})) \\ & -\sum_{k=1}^{K} \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}}\|\nabla_{\mathbf{x}}^{G} \log P(y=k|\mathbf{x})\|^{2}\end{aligned} \tag{8}$$

where $\nabla_{\mathbf{x}}^{G} \log P(y=k|\mathbf{x})$ of the last term is the gradient of the log-likelihood along the manifold $G(\mathbf{x}, \mathbf{z})$ at $\mathbf{x}$, that is $\nabla_{\mathbf{z}} \log P(y=k|G(\mathbf{x}, \mathbf{z}))|_{\mathbf{z}=\mathbf{0}}$. Let us explain the objective (**??**) in detail below.

- The firs term maximizes the log-likelihood that a labeled training example drawn from the distribution $P_{\mathcal{L}}$ of labeled examples is correctly classifie by $P(y|\mathbf{x})$.
- The second term maximizes the log-likelihood that an unlabeled example $\mathbf{x}_{u}$ drawn from the data distribution $P_{\mathcal{X}}$ is assigned to one of K real classes (i.e., $y_{u} \leq K$).
- The third term enforces $P(y|\mathbf{x})$ to classify a generated sample by $G(\mathbf{x}, \mathbf{z})$ as fake (i.e., $y=K+1$).
- The last term penalizes a sudden change of classifica tion function on the manifold, thus yielding a locally consistent classifie as expected. This can be seen by viewing $\log P(y|\mathbf{x})$ as f in (**??**).

On the other hand, with a fi ed classifie $P(y|\mathbf{x})$, the local generator G is trained by the following objective:

$$\min_{G} \mathcal{K}_{G}+\mathcal{L}_{G}+\mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}} \Omega_{G}(\mathbf{x}) \tag{9}$$

where

- The firs term is *label preservation* term

  $$\mathcal{K}_{G}=-\mathbb{E}_{(\mathbf{x}_{l}, y_{l}) \sim P_{\mathcal{L}}, \mathbf{z} \sim P_{\mathcal{Z}}} \log P(y_{l}|G(\mathbf{x}_{l}, \mathbf{z}))$$

  which enforces generated samples should not change the labels of their original examples. This label preservation term can help explore intra-class variance by generating new variants of training examples without changing their labels.

Figure 4. Handwritten digits generated by LGAN on the MNITS dataset. The middle column in a red bounding box represents the image at the origin $\mathbf{z} = 0$ of a local coordinate chart. In each row, the images are generated in one direction of a local coordinate.

prior when training a local generator in its local coordinate chart.

We used Adam solver to update the network parameters where the learning rate is set to 5×10^{-5} and 10^{-3} for training discriminator and generator networks respectively. The two hyperparameters μ and η imposing locality and orthonormality priors in the regularizer were chosen based on an independent validation set held out from the training set.

5.2. Image Generation with Diversity

Figure **??** illustrates the generated images on the CelebA dataset. In this task, 32-D local coordinates were used in the LGAN, and each row was generated by varying one of 32 local coordinates while fixin the others. In other worlds, each row represents image transformations in one coordinate direction. The middle column in a red bounding box corresponds to the original image at the origin of local coordinates. The figur shows how a face transforms as it moves away from the origin along different coordinate directions on the manifold. The results demonstrate LGAN can generate sharp-looking faces with various patterns of transformations, including the variations in facial expressions, beards, skin colors, haircuts and poses. This also illustrates the L-GAN was able to disentangle different patterns of image transformations in its local coordinate charts on CelebA because of the orthonormality imposed on local tangent basis.

Moreover, we note that a face generated by LGAN could transform to the face of a different person in Figure **??**. For example, in the firs and the sixth row of the left figure we can see that a female face transforms to a male face. Similarly, in the forth and the fift row of the right figure the male face gradually becomes more female. This shows that local generators can not only manipulate attributes of input images, but are also able to extrapolate these inputs to generate very different output images.

We also illustrate the image generation results on the M-NIST dataset in Figure **??**. Again, we notice the factorized transformations in different tangent directions – across d-

Figure 5. Network architecture for local generators on SVHN and CIFAR-10.

coordinates were used on CIFAR-10 as natural scene images could contain more patterns of image transformations than street view house numbers. To reduce computational cost, in each minibatch, ten coordinates were randomly chosen when computing the back-propagated errors on the orthonormal prior between local tangents. We also tested by sampling more coordinates but did not observe any significan improvement on the accuracy. So we only sampled ten coordinates in a minibatch iteration to make a balance between cost and performance.

Table **??** reports the experiment results on both SVHN and CIFAR-10. On SVHN, we used 500 and $1,000$ labeled images to train the semi-supervised LGAN, which is 50 and 100 labeled examples per class, and the remaining training examples were left unlabeled when they were used to train the model. Similarly, on CIFAR-10, we used $1,000$ and $4,000$ labeled examples with the remaining training examples being left unlabeled. The results show that on both datasets, the proposed semi-supervised LGAN outperforms the other compared GAN-based semi-supervised methods.

(b) CIFAR-10

Figure 6. Tangent images generated by LGAN along ten randomly chosen coordinates on SVHN and CIFAR-10 datasets. The firs column in the red bounding box shows the original images, followed by their tangent images in each row.

Furthermore, we illustrate tangent images in Figure **??** on SVHN and CIFAR-10 datasets. The firs column in the red bounding box shows the original images, followed by their tangent images generated by the learned local generators along ten randomly chosen coordinates in each row. These tangent images visualize the local variations captured by LGAN along different coordinate directions. This shows how the model is able to learn a locally consistent classifie by exploring the geometry of image transformations along these tangent directions in a neighborhood of the underlying manifold.

6. Conclusion

This paper presents a novel paradigm of localized GAN (LGAN) model along with its application in semi-supervised learning tasks. The model uses an atlas of local coordinate charts and associated local generators to cover an entire manifold, allowing it to capture distinct geometry of local transformations across the manifold. It also enables a direct access to manifold structures from local coordinates, tangents to Jacobian matrices without having to invert the global generator in the classic GAN. Moreover, by enforcing orthonormality between tangents, it can prevent the manifold from being locally collapsed to a dimensionally deficien subspace, which provides a geometric insight into alleviating mode collapse problem encountered in literature. Its application to semi-supervised learning reveals the connection with Laplace-Beltrami operator on the manifold, yielding a locally consistent classifie resilient against perturbations in different tangent directions. Experiment results on both image generation and classificatio tasks demonstrate its superior performances to the other state-of-

the-art models.

A. More Results on Semi-supervised Learning

This section describes more experiment results on the semi-supervised learning using LGAN. Besides the small discriminator structure employed in Section 5.3, we further test LGAN with a larger CNN architecture, which is the same one as the "Conv-Large" used in [**?**]. For convenience, we will refer this larger discriminator as Conv-Large, while the one used in Section 5.3 as "Conv-Small" in the following. We compare the Conv-Large LGAN with the state-of-the-art semi-supervised learning methods (which are not necessarily the GAN-based) and report the results on both CIFAR-10 and CIFAR-100 datasets in this section. The architecture of the generator will keep the same as that used in Conv-Small experiments.

A.1. Discriminator Architectures

Table **??** and **??** summarize the architecture of Conv-Small and Conv-Large, respectively. We also apply the weight normalization [**?**] to all convolutional and dense layers in both architectures.

A.2. Training Details for Conv-Large

Like in training the Conv-Small, we adopt Adam optimizer to train both the discriminator and generator. The learning rate is set to 4×10^{-4}, and the maximal training epoch is $1,200$. We gradually anneal the learning rates to zero during the last 400 epochs. The other settings are kept as same as those for training Conv-Small (Section 5.3). For CIFAR-100, which consists of $50,000$ 32×32 training images and $10,000$ test images in a hundred classes, we change the dropout rate of drop1 layer from 0.2 to 0.1 and the output dimension of the last layer to 100. We also adopt early stopping – the training is terminated if the validation error stops decreasing over 100 consecutive epochs after the 600th epoch. The two hyper-parameters μ and η are chosen based on a separate validation set.

Name	Description
Input	32×32 RGB image
drop1	Dropout $p = 0.2$
conv1a	96, 3×3, pad=1, stride=1, LReLU
conv1b	96, 3×3, pad=1, stride=1, LReLU
conv1c	96, 3×3, pad=1, stride=2, LReLU
drop2	Dropout $p = 0.5$
conv2a	192, 3×3, pad=1, stride=1, LReLU
conv2b	192, 3×3, pad=1, stride=1, LReLU
conv2c	192, 3×3, pad=1, stride=2, LReLU
drop3	Dropout $p = 0.5$
conv3a	192, 3×3, pad=0, stride=1, LReLU
conv3b	192, 1×1, LReLU
conv3c	192, 1×1, LReLU
pool1	Global mean pooling $6 \times 6 \rightarrow 1 \times 1$
dense	Fully connected $192 \rightarrow 10$
output	Softmax

Table 2. The network architectures of Conv-Small

A.3. Experimental Results for Conv-Large

We compare the LGAN using Conv-Large discriminator with state-of-the-art semi-supervised baselines. The results are reported in Table **??**. Note that we used $4,000$ and $10,000$ labeled training examples for CIFAR-10 (400 images per class) and CIFAR-100 (100 images per class) respectively and the rest of training data unlabeled. From the table, we can see that LGAN with Conv-Large outperforms the other compared methods on both datasets.

Name	Description
Input	32×32 RGB image
drop1	Dropout $p = 0.2$
conv1a	128, 3×3, pad=1, stride=1, LReLU
conv1b	128, 3×3, pad=1, stride=1, LReLU
conv1c	128, 3×3, pad=1, stride=1, LReLU
pool1	Maxpooling 2×2
drop2	Dropout $p = 0.5$
conv2a	256, 3×3, pad=1, stride=1, LReLU
conv2b	256, 3×3, pad=1, stride=1, LReLU
conv2c	256, 3×3, pad=1, stride=1, LReLU
pool2	Maxpooling 2×2
drop3	Dropout $p = 0.5$
conv3a	512, 3×3, pad=0, stride=1, LReLU
conv3b	256, 1×1, LReLU
conv3c	128, 1×1, LReLU
pool3	Global mean pooling $6 \times 6 \rightarrow 1 \times 1$
drop4	Dropout $p = 0.1$
dense	Fully connected $128 \rightarrow 10$
output	Softmax

Table 3. The network architectures of Conv-Large

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, January 2017.

[2] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[4] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*, 2017.

Method	CIFAR-10	CIFAR-100
Π model [?]	12.36 ± 0.31	39.19 ± 0.36
Temporal Ensembling [?]	12.16 ± 0.24	38.65 ± 0.51
Sajjadi et al. [?]	11.29 ± 0.24	-
VAT [?]	10.55	-
VadD [?]	11.32 ± 0.11	-
LGAN (Conv-Large)	$\mathbf{9.77 \pm 0.13}$	$\mathbf{35.52 \pm 0.33}$

Table 4. Classificatio errors on both CIFAR-10 and CIFAR-100 with $4,000$ and $10,000$ labeled training examples respectively. The best result is highlighted in bold.

[5] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[6] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

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

[8] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[9] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[11] A. Kumar, P. Sattigeri, and P. T. Fletcher. Improved semi-supervised learning with gans using manifold invariances. *arXiv preprint arXiv:1705.08850*, 2017.

[12] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[13] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. *arXiv preprint arXiv:1703.02291*, 2017.

[14] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017.

[15] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[16] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon. Adversarial dropout for supervised and semi-supervised learning. *arXiv preprint arXiv:1707.03631*, 2017.

[17] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[18] G.-J. Qi, W. Liu, C. Aggarwal, and T. Huang. Joint intermodal and intramodal label transfers for extremely rare or unseen classes. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1360–1373, 2017.

[19] G.-J. Qi, Q. Tian, and T. Huang. Locality-sensitive support vector machine by exploring local correlation and global regularization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 841–848. IEEE, 2011.

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

[21] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[22] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016.

[24] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

[25] P. Y. Simard, Y. A. Le Cun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition: Tangent distance and propagation.

[26] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[27] J. Tang, X.-S. Hua, G.-J. Qi, and X. Wu. Typicality ranking via semi-supervised multiple-instance learning. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 297–300. ACM, 2007.

[28] J. Tang, H. Li, G.-J. Qi, and T.-S. Chua. Integrated graph-based semi-supervised multiple/single instance learning framework for image annotation. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 631–634. ACM, 2008.

[29] H. Valpola. From neural pca to deep unsupervised learning. *Adv. in Independent Component Analysis and Learning Machines*, pages 143–171, 2015.

[30] T. J. Willmore. *An introduction to differential geometry*. Courier Corporation, 2013.

[31] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

[32] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian field and harmonic functions. In

Proceedings of the 20th International conference on Machine learning (ICML-03), pages 912–919, 2003.