Generalized Loss-Sensitive Adversarial Learning with Manifold Margins

Marzieh Edraki and Guo-Jun Qi[⋆]

Laboratory for MAchine Perception and LEarning (MAPLE)
http://maple.cs.ucf.edu/
University of Central Florida, Orlando FL 32816, USA
m.edraki@knights.ucf.edu, guojun.qi@ucf.edu

Abstract. The classic Generative Adversarial Net and its variants can be roughly categorized into two large families: the unregularized versus regularized GANs. By relaxing the non-parametric assumption on the discriminator in the classic GAN, the regularized GANs have better generalization ability to produce new samples drawn from the real distribution. It is well known that the real data like natural images are not uniformly distributed over the whole data space. Instead, they are often restricted to a low-dimensional manifold of the ambient space. Such a manifold assumption suggests the distance over the manifold should be a better measure to characterize the distinct between real and fake samples. Thus, we define a pullback operator to map samples back to their data manifold, and a manifold margin is defined as the distance between the pullback representations to distinguish between real and fake samples and learn the optimal generators. We justify the effectiveness of the proposed model both theoretically and empirically.

Keywords: Regularized GAN, Image generation, Semi-supervised classification, Lipschitz regularization.

1 Introduction

Since the Generative Adversarial Nets (GAN) was proposed by Goodfellow et al. [4], it has attracted much attention in literature with a number of variants have been proposed to improve its data generation quality and training stability. In brief, the GANs attempt to train a generator and a discriminator that play an adversarial game to mutually improve one another [4]. A discriminator is trained to distinguish between real and generated samples as much as possible, while a generator attempts to generate good samples that can fool the discriminator. Eventually, an equilibrium is reached where the generator can produce high quality samples that cannot be distinguished by a well trained discriminator.

The classic GAN and its variants can be roughly categorized into two large families: the unregularized versus regularized GANs. The former contains the original GAN and many variants [11,24], where the consistency between the

[⋆] Corresponding author: G.-J. Qi, email: guojunq@gmail.com distribution of their generated samples and real data is established based on the non-parametric assumption that their discriminators have infinite modeling ability. In other words, the unregularized GANs assume the discriminator can take an arbitrary form so that the generator can produce samples following any given distribution of real samples.

On the contrary, the regularized GANs focus on some regularity conditions on the underlying distribution of real data, and it has some constraints on the discriminators to control their modeling abilities. The two most representative models in this category are Loss-Sensitive GAN (LS-GAN) [11] and Wasserstein GAN (WGAN) [1]. Both are enforcing the Lipschitz constraint on training their discriminators. Moreover, it has been shown that the Lipschitz regularization on the loss function of the LS-GAN yields a generator that can produce samples distributed according to any Lipschitz density, which is a regularized form of distribution on the supporting manifold of real data.

Compared with the family of unregularized GANs, the regularized GANs sacrifice their ability to generate an unconstrained distribution of samples for better training stability and generalization performances. For examples, both LS-GAN and WGAN can produce uncollapsed natural images without involving batch-normalization layers, and both address vanishing gradient problem in training their generators. Moreover, the generalizability of the LS-GAN has also been proved with the Lipschitz regularity condition, showing the model can generalize to produce data following the real density with only a reasonable number of training examples that are polynomial in model complexity. In other words, the generalizability asserts the model will not be overfitted to merely memorize training examples; instead it will be able to extrapolate to produce unseen examples beyond provided real examples.

Although the regularized GANs, in particular LS-GAN [11] considered in this paper, have shown compelling performances, there are still some unaddressed problems. The loss function of LS-GAN is designed based on a margin function defined over ambient space to separate the loss of real and fake samples. While the margin-based constraint on training the loss function is intuitive, directly using the ambient distance as the loss margin may not accurately reflect the dissimilarity between data points.

It is well known that the real data like natural images do not uniformly distribute over the whole data space. Instead, they are often restricted to a low-dimensional manifold of the ambient space. Such manifold assumption suggests the "geodesic" distance over the manifold should be a better measure of the margin to separate the loss functions between real and fake examples. For this purpose, we will define a pullback mapping that can invert the generator function by mapping a sample back to the data manifold. Then a manifold margin is defined as the distance between the representation of data points on the manifold to approximate their distance. The loss function, the generator and the pullback mapping are jointly learned by a threefold adversarial game. We will prove that the fixed point characterized by this game will be able to yield a generator that can produce samples following the real distribution of samples.

2 Related work

The original GAN [4,14,17] can be viewed as the most classic unregularized model with its discriminator based on a non-parametric assumption of infinite modeling ability. Since then, great research efforts have been made to efficiently train the GAN by different criteria and architectures [15,22,19].

In contrast to unregularized GANs, Loss-Sensitive GAN (LS-GAN) [11] was recently presented to regularize the learning of a loss function in Lipschitz space, and proved the generalizability of the resultant model. [1] also proposed to minimize the Earth-Mover distance between the density of generated samples and the true data density, and they show the resultant Wasserstein GAN (WGAN) can address the vanishing gradient problem that the classic GAN suffers from. Coincidentally, the learning of WGAN is also constrained in a Lipschitz space.

Recent efforts [2,3] have also been made to learn a generator along with a corresponding encoder to obtain the representation of input data. The generator and encoder are simultaneously learned by jointly distinguishing between not only real and generated samples but also their latent variables in an adversarial process. Both methods still focus on learning unregularized GAN models without regularization constraints.

Researchers also leverage the learned representations by deep generative networks to improve the classification accuracy when it is too difficult or expensive to label sufficient training examples. For example, Qi et al. [13] propose a localized GAN to explore data variations in proximity of datapoints for semi-supervised learning. It can directly calculate Laplace-Beltra operator, which makes it amenable to handle large-scale data without resorting to a graph Laplacian approximation. [6] presents variational auto-encoders [7] by combining deep generative models and approximate variational inference to explore both labeled and unlabeled data. [17] treats the samples from the GAN generator as a new class, and explore unlabeled examples by assigning them to a class different from the new one. [15] proposes to train a ladder network [22] by minimizing the sum of supervised and unsupervised cost functions through back-propagation, which avoids the conventional layer-wise pre-training approach. [19] presents an approach to learn a discriminative classifier by trading-off mutual information between observed examples and their predicted classes against an adversarial generative model. [3] seeks to jointly distinguish between not only real and generated samples but also their latent variables in an adversarial process. These methods have shown promising results for classification tasks by leveraging deep generative models.

3 The Formulation

3.1 Loss Functions and Margins

The Loss-Sensitive Adversarial Learning (LSAL) aims to generate data by learning a generator G that transforms a latent vector $\mathbf{z} \in \mathcal{Z}$ of random variables drawn from a distribution $P_Z(\mathbf{z})$ to a real sample $\mathbf{x} \triangleq G(\mathbf{z}) \in \mathcal{X}$, where $\mathcal{Z}$ and

$\mathcal{X}$ are the noise and data spaces respectively. Usually, the space $\mathcal{Z}$ is of a lower dimensionality than $\mathcal{X}$, and the generator mapping G can be considered as an embedding of $\mathcal{Z}$ into a low-dimensional manifold $G(\mathcal{Z}) \subset \mathcal{X}$. In this sense, each $\mathbf{z}$ can be considered as a compact representation of $G(\mathbf{z}) \in \mathcal{X}$ on the manifold $G(\mathcal{Z})$.

Then, we can define a loss function L over the data domain $\mathcal{X}$ to characterize if a sample $\mathbf{x}$ is real or not. The smaller the loss L, the more likely $\mathbf{x}$ is a real sample. To learn L, a margin $\Delta_x(\mathbf{x}, \mathbf{x}')$ that measures the dissimilarity between samples will be defined to separate the loss functions between a pair of samples $\mathbf{x}$ and $\mathbf{x}'$, so that the loss of a real sample should be smaller than that of a fake sample $\mathbf{x}'$ by at least $\Delta_x(\mathbf{x}, \mathbf{x}')$. Since the margin $\Delta_x(\mathbf{x}, \mathbf{x}')$ is defined over the samples in their original ambient space $\mathcal{X}$ directly, we called it *ambient margin.*

In the meantime, we can also define a manifold margin $\Delta_z(\mathbf{z}, \mathbf{z}')$ over the manifold representations to separate the losses between real and generated samples. This is because the ambient margin alone may not well reflect the difference between samples, in particular considering real data like natural images often only occupy a small low-dimensional manifold embedded in the ambient space. Alternatively, the manifold will better capture the difference between data points to separate their losses on the manifold of real data.

To this end, we propose to learn another pullback mapping Q that projects a sample $\mathbf{x}$ back to the latent vector $\mathbf{z} \triangleq Q(\mathbf{x})$ that can be viewed as the low-dimensional representation of $\mathbf{x}$ over the underlying data manifold. Then, we can use the distance $\Delta_z(\mathbf{z}, \mathbf{z}')$ between latent vectors to approximate the geodesic distance between the projected points on the data manifold, and use it to define the *manifold margin* to separate the loss functions of different data points.

3.2 Learning Objectives

Formally, let us consider a loss function $L(\mathbf{x}, \mathbf{z})$ defined over a joint space $\mathcal{X} \times \mathcal{Z}$ of data and latent vectors. For a real sample $\mathbf{x}$ and its corresponding latent vector $Q(\mathbf{x})$, its loss function $L(\mathbf{x}, Q(\mathbf{x}))$ should be smaller than $L(G(\mathbf{z}), \mathbf{z})$ of a fake sample $G(\mathbf{z})$ and its latent vector $\mathbf{z}$. The required margin between them is defined as a combination of margins over data samples and latent vectors

$$\Delta_{\mu,\nu}(\mathbf{x}, \mathbf{z}) \triangleq \mu \Delta_x(\mathbf{x}, G(\mathbf{z})) + \nu \Delta_z(Q(\mathbf{x}), \mathbf{z}) \tag{1}$$

where the first term is the ambient margin separating loss functions between data points in the ambient space $\mathcal{X}$, while the second term is the manifold margin that separates loss functions based on the distance between latent vectors.

When a fake sample is far away from a real one, a larger margin will be imposed between them to separate their losses; otherwise, a smaller margin will be used to separate the losses. This allows the model to focus on improving the poor samples that are still far away from real samples, instead of wasting efforts on improving those well-generated data that are already close to real examples.

Then we will use the following objective functions to learn the fixed points of loss function L^*, the generator G^* and the pullback mapping Q^* by solving the following optimization problems.

(I) Learning L with fixed G^* and Q^*:

$$L^* = \arg\min_L \mathcal{S}(L, G^*, Q^*) \triangleq \mathbb{E}_{\substack{\mathbf{x}\sim P_x(\mathbf{X}) \\ \mathbf{z}\sim P_Z(\mathbf{z})}} C\big[\Delta_{\mu,\nu}(\mathbf{x},\mathbf{z}) + L(\mathbf{x}, Q^*(\mathbf{x})) - L(G^*(\mathbf{z}),\mathbf{z})\big]$$

(II) Learning G with fixed L^* and Q^*:

$$G^* = \arg\min_G \mathcal{T}(L^*, G, Q^*) \triangleq \mathbb{E}_{\mathbf{z}\sim P_Z(\mathbf{z})} L^*(G(\mathbf{z}),\mathbf{z}))$$

(III) Learning Q with fixed L^* and G^*:

$$Q^* = \arg\max_Q \mathcal{R}(L^*, G^*, Q) \triangleq \mathbb{E}_{\mathbf{x}\sim P_X(\mathbf{x})} L^*(\mathbf{x}, Q(\mathbf{x}))$$

where 1) the expectations in the above three objective functions are taken with respect to the probability measure P_X of real samples $\mathbf{x}$ and/or the probability measure P_Z of latent vectors $\mathbf{z}$. 2) the function $C[\cdot]$ is the cost function measuring the degree of the loss function L violating the required margin $\Delta_{\mu,\nu}(\mathbf{x},\mathbf{z})$, and it should satisfy the following two conditions:

$$C[a] = a \quad \text{for} \quad a \geq 0, \quad \text{and} \quad C[a] \geq a \quad \text{for} \quad \text{any} \quad a \in \mathbb{R}.$$

For example, the hinge loss $[a]_+ = \max(0, a)$ satisfies these two conditions, and it results in a LSAL model by penalizing the violation of margin requirement. Any rectifier linear function $\mathrm{ReLU}(a) = \max(a, \eta a)$ with a slope $\eta \leq 1$ also satisfies these two conditions.

Later on, we will prove the LSAL model satisfying these two conditions can produce samples following the true distribution of real data, i.e., the distributional consistency between real and generated samples.

3) Problem (II) and (III) learn the generator G and pullback mapping Q in an adversarial fashion: G is learned by minimizing the loss function L^* since real samples and their latent vectors should have a smaller loss. In contrast, Q is learned by maximizing the loss function L^* – the reason will become clear in the theoretical justification of the following section when proving the distributional consistency between real and generated samples.

4 Theoretical Justification

In this section, we will justify the learning objectives of the proposed LSAL model by proving the distributional consistency between real and generated samples.

Formally, we will show that the joint distribution $P_{GZ}(\mathbf{x},\mathbf{z}) = P_Z(\mathbf{z})P_{X|Z}(\mathbf{x}|\mathbf{z})$ of generated sample $\mathbf{x} = G(\mathbf{z})$ and the latent vector $\mathbf{z}$ matches the joint distribution $P_{QX}(\mathbf{x},\mathbf{z}) = P_X(\mathbf{x})P_{Z|X}(\mathbf{z}|\mathbf{x})$ of the real sample $\mathbf{x}$ and its latent vector $\mathbf{z} = Q(\mathbf{x})$, i.e., $P_{GZ} = P_{QX}$. Then, by marginalizing out $\mathbf{z}$, we will be able to show the marginal distribution $P_{GZ}(\mathbf{x}) = \int_{\mathbf{z}} P_{GZ}(\mathbf{x},\mathbf{z})d\mathbf{z}$ of generated samples is consistent with $P_X(\mathbf{x})$ of the real samples. Hence, the main result justifying the distributional consistency for the LSAL model is Theorem 1 below.

4.1 Auxiliary Functions and their Property

First, let us define two auxiliary functions that will be used in the proof:

$$f_{QX}(\mathbf{x},\mathbf{z}) = \frac{dP_{QX}}{dP_{GQ}}, \quad f_{GZ}(\mathbf{x},\mathbf{z}) = \frac{dP_{GZ}}{dP_{GQ}} \tag{2}$$

where $P_{GQ} = P_{GZ} + P_{QX}$, and the above two derivatives defining the auxiliary functions are the Radon-Nikodym derivative that exists since P_{QX} and P_{GZ} are absolutely continuous with respect to P_{GQ}. Here, we will need the following property regarding these two functions in our theoretical justification.

Lemma 1. *If $f_{QX}(\mathbf{x},\mathbf{z}) \geq f_{GZ}(\mathbf{x},\mathbf{z})$ for P_{GQ}-almost everywhere, we must have $P_{GZ} = P_{QX}$.*

Proof. To show $P_{GZ} = P_{QX}$, consider an arbitrary subset $R \subseteq \mathcal{X} \times \mathcal{Z}$. We have

$$\begin{aligned} P_{QX}(R) &= \int_R dP_{QX} = \int_R \frac{dP_{QX}}{dP_{GQ}} dP_{GQ} = \int_R f_{QX} dP_{GQ} \\ &\geq \int_R f_{GZ} dP_{GQ} = \int_R \frac{dP_{GZ}}{dP_{GQ}} dP_{GQ} = \int_R dP_{GZ} = P_{GZ}(R). \end{aligned} \tag{3}$$

Similarly, we can show the following inequality on $\Omega \setminus R$ with $\Omega = \mathcal{X} \times \mathcal{Z}$

$$P_{QX}(\Omega \setminus R) \geq P_{GZ}(\Omega \setminus R).$$

Since $P_{QX}(R) = 1 - P_{QX}(\Omega \setminus R)$ and $P_{GZ}(R) = 1 - P_{GZ}(\Omega \setminus R)$, we have

$$P_{QX}(R) = 1 - P_{QX}(\Omega \setminus R) \leq 1 - P_{GZ}(\Omega \setminus R) = P_{GZ}(R). \tag{4}$$

Putting together Eq. (3) and (4), we have $P_{QX}(R) = P_{GZ}(R)$ for an arbitrary R, and thus $P_{QX} = P_{GZ}$, which completes the proof.

4.2 Main Result on Consistency

Now we can prove the consistency between generated and real samples with the following Lipschitz regularity condition on f_{QX} and f_{GZ}.

Assumption 1 *Both $f_{QX}(\mathbf{x},\mathbf{z})$ and $f_{GZ}(\mathbf{x},\mathbf{z})$ have bounded Lipschitz constants in $(\mathbf{x},\mathbf{z})$.*

It is noted that the bounded Lipschitz condition for both functions is only applied to the support of $(\mathbf{x},\mathbf{z})$. In other words, we only require the Lipschitz condition hold on the joint space $\mathcal{X} \times Z$ of data and latent vectors.

Then, we can prove the following main theorem.

Theorem 1. *Under Assumption 1, $P_{QX} = P_{GZ}$ for P_{GQ}-almost everywhere with the optimal generator G^* and the pullback mapping Q^*. Moreover, $f_{QX} = f_{GZ} = \dfrac{1}{2}$ at the optimum.*

The second part of the theorem follows from the first part. since $P_{QX} = P_{GZ}$ for the optimum G^* and Q^*, $f_{QX} = \dfrac{dP_{QX}}{dP_{GQ}} = \dfrac{dP_{QX}}{d(P_{QX} + P_{GZ})} = \dfrac{1}{2}$. Similarly, $f_{GZ} = \dfrac{1}{2}$. This shows f_{QX} and f_{GZ} are both Lipschitz at the fixed point depicted by Problem (I)-(III).

Here we give the proof of this theorem step-by-step in detail. The proof will shed us some light on the roles of the ambient and manifold margins as well as the Lipschitz regularity in guaranteeing the distributional consistency between generated and real samples.

Proof. **Step 1:** First, we will show that

$$\mathcal{S}(L^*, G^*, Q^*) \geq \mathbb{E}_{\mathbf{x},\mathbf{z}}[\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})], \tag{5}$$

where $\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})$ is defined in Eq. (1) with G and Q being replaced with their optimum G^* and Q^*.

This can be proved following the deduction below

$$\mathcal{S}(L^*, G^*, Q^*) \geq \mathbb{E}_{\mathbf{x},\mathbf{z}}[\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})] + \mathbb{E}_{\mathbf{x}} L^*(\mathbf{x}, Q^*(\mathbf{x})) - \mathbb{E}_{\mathbf{z}} L^*(G^*(\mathbf{z}), \mathbf{z})$$

This follows from $C[a] \geq a$. Continuing the deduction, we have the RHS of the last inequality equals

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},\mathbf{z}}[\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})] + \int_{\mathbf{x},\mathbf{z}} L^*(\mathbf{x}, \mathbf{z}) dP_{Z|X}(\mathbf{z} = Q^*(\mathbf{x})|\mathbf{x}) dP_X(\mathbf{x}) \\
&- \int_{\mathbf{x},\mathbf{z}} L^*(\mathbf{x}, \mathbf{z}) dP_{X|Z}(\mathbf{x} = G^*(\mathbf{z})|\mathbf{z}) dP_Z(\mathbf{z}) \\
&\geq \mathbb{E}_{\mathbf{x},\mathbf{z}}[\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})] + \int_{\mathbf{x},\mathbf{z}} L^*(\mathbf{x}, \mathbf{z}) dP_Z(\mathbf{z}) dP_X(\mathbf{x}) - \int_{\mathbf{x},\mathbf{z}} L^*(\mathbf{x}, \mathbf{z}) dP_X(\mathbf{x}) dP_Z(\mathbf{z}) \\
&= \mathbb{E}_{\mathbf{x},\mathbf{z}}[\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z})],
\end{aligned}$$

which follows from the Problem (II) and (III) where G^* and Q^* minimizes and maximizes L^* respectively. Hence, the second and third terms in the LHS are lower bounded when $P_{Z|X}(\mathbf{z} = Q^*(\mathbf{x})|\mathbf{x})$ and $P_{X|Z}(\mathbf{x} = G^*(\mathbf{z})|\mathbf{z})$ are replaced with $P_Z(\mathbf{z})$ and $P_X(\mathbf{x})$ respectively.

Step 2: we will show that $f_{QX} \geq f_{GZ}$ for P_{GQ}-almost everywhere so that we can apply Lemma 1 to prove the consistency.

With Assumption 1, we can define the following Lipschitz continuous loss function

$$L(\mathbf{x}, \mathbf{z}) = \alpha[-f_{QX}(\mathbf{x}, \mathbf{z}) + f_{GZ}(\mathbf{x}, \mathbf{z})]_+ \tag{6}$$

with a sufficiently small $\alpha > 0$. Thus, $L(\mathbf{x}, \mathbf{z})$ will also be Lipschitz continuous whose Lipschitz constants are smaller than μ and ν in $\mathbf{x}$ and $\mathbf{z}$ respectively. This will result in the following inequality

$$\Delta^*_{\mu,\nu}(\mathbf{x}, \mathbf{z}) + L(\mathbf{x}, Q^*(\mathbf{x})) - L(G^*(\mathbf{z}), \mathbf{z}) \geq 0.$$

Then, by $C[a] = a$ for $a \geq 0$, we have

$$\begin{aligned}\mathcal{S}(L, Q^*, G^*) &= \mathbb{E}_{\mathbf{x},\mathbf{z}}[\varDelta^*_{\mu,\nu}(\mathbf{x},\mathbf{z})] + \int_{\mathbf{x},\mathbf{z}} L(\mathbf{x},\mathbf{z})dP_{QX} - \int_{\mathbf{x},\mathbf{z}} L(\mathbf{x},\mathbf{z})dP_{GZ} \\ &= \mathbb{E}_{\mathbf{x},\mathbf{z}}[\varDelta^*_{\mu,\nu}(\mathbf{x},\mathbf{z})] + \int_{\mathbf{x},\mathbf{z}} L(\mathbf{x},\mathbf{z})f_{QX}(\mathbf{x},\mathbf{z})dP_{GQ} - \int_{\mathbf{x},\mathbf{z}} L(\mathbf{x},\mathbf{z})f_{GZ}(\mathbf{x},\mathbf{z})dP_{GQ}\end{aligned}$$

where the last equality follows from Eq. (2). By substituting (6) into the RHS of the above equality, we have

$$\mathcal{S}(L, Q^*, G^*) = \mathbb{E}_{\mathbf{x},\mathbf{z}}[\varDelta^*_{\mu,\nu}(\mathbf{x},\mathbf{z})] - \alpha \int_{\mathbf{x},\mathbf{z}} [-f_{QX}(\mathbf{x},\mathbf{z}) + f_{GZ}(\mathbf{x},\mathbf{z})]_+^2 dP_{GQ}$$

Let us assume that $f_{QX}(\mathbf{x},\mathbf{z}) < f_{GZ}(\mathbf{x},\mathbf{z})$ holds on a subset $(\mathbf{x},\mathbf{z})$ of nonzero measure with respect to P_{GQ}. Then since $\alpha > 0$, we have

$$\mathcal{S}(L^*, Q^*, G^*) \leq \mathcal{S}(L, Q^*, G^*) < \mathbb{E}_{\mathbf{x},\mathbf{z}}[\varDelta^*_{\mu,\nu}(\mathbf{x},\mathbf{z})]$$

The first inequality arises from Problem (I) where L^* minimizes $\mathcal{S}(L, Q^*, G^*)$. Obviously, this contradicts with (5), thus we must have $f_{QX}(\mathbf{x},\mathbf{z}) \geq f_{GZ}(\mathbf{x},\mathbf{z})$ for P_{GQ}-almost everywhere. This completes the proof of Step 2.
Step 3: Now the theorem can be proved by combining Lemma 1 and the result from Step 2.

As a corollary, we can show that the optimal Q^* and G^* are mutually inverse.

Corollary 1. *With optimal Q^* and G^*, ${Q^*}^{-1} = G^*$ almost everywhere. In other words, $Q^*(G^*(\mathbf{z})) = z$ for P_Z-almost every $\mathbf{z} \in \mathcal{Z}$ and $G^*(Q^*(\mathbf{x})) = x$ for P_X-almost every $\mathbf{x} \in \mathcal{X}$.*

The corollary is a consequence of the proved distributional consistency $P_{QX} = P_{GZ}$ for optimal Q^* and G^* as shown in [2]. This implies that the optimal pullback mapping $Q^*(\mathbf{x})$ forms a latent representation of $\mathbf{x}$ as the inverse of the optimal generator function G^*.

5 Semi-Supervised Learning

LSAL can also be used to train a semi-supervised classifier [25,21,12] by exploring a large amount of unlabeled examples when the labeled samples are scarce. To serve as a classifier, the loss function $L(\mathbf{x},\mathbf{z},\mathbf{y})$ can be redefined over a joint space of $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$ where $\mathcal{Y}$ is the label space. Now the loss function measures the cost of assigning jointly a sample $\mathbf{x}$ and its manifold representation $Q(\mathbf{x})$ to a label y^* by minimizing $L(\mathbf{x},\mathbf{z},\mathbf{y})$ over $\mathcal{Y}$ below

$$y^* = \arg\min_{y \in \mathcal{Y}} L(\mathbf{x},\mathbf{z},\mathbf{y}) \tag{7}$$

To train the Loss function of LSAL in a semi-supervised fashion, We define the following objective function

$$\mathcal{S}(L, G, Q) = \mathcal{S}_l(L, G, Q) + \lambda\, \mathcal{S}_u(L, G, Q) \tag{8}$$

where $\mathcal{S}_l$ is the objective function for labeled examples while $\mathcal{S}_u$ is for unlabeled samples, and λ is a positive coefficient balancing between the contributions of labeled and unlabeled data.

Since our goal is to classify a pair of $(\mathbf{x}, Q(\mathbf{x}))$ to one class in the label space $\mathcal{Y}$, we can define the loss function L by the negative log-softmax output from a network. So we have

$$L(\mathbf{x}, \mathbf{z}, \mathbf{y}) = -\log \frac{\exp(a_y(\mathbf{x}, \mathbf{z}))}{\sum_{y'} \exp(a_{y'}(\mathbf{x}, \mathbf{z}))}$$

which $a_y(\mathbf{x}, \mathbf{z})$ is the activation output of class $\mathbf{y}$. By the LSAL formulation, given a label example $(\mathbf{x}, \mathbf{y})$, the $L(\mathbf{x}, Q(\mathbf{x}), \mathbf{y})$ should be smaller than $L(G(\mathbf{z}), \mathbf{z}, \mathbf{y})$ by at least a margin of $\Delta_{\mu,\nu}(\mathbf{x}, \mathbf{z})$. So the objective S_l is defined as

$$\begin{aligned} &\mathcal{S}_l(L, G^*, Q^*) \triangleq \\ &\mathbb{E}_{\substack{\mathbf{x},\mathbf{y} \sim \mathbf{P}_{data}(\mathbf{x},\mathbf{y}) \\ \mathbf{z} \sim \mathbf{P}_Z(\mathbf{z})}} C\big[\Delta_{\mu,\nu}(\mathbf{x}, \mathbf{z}) + L(\mathbf{x}, Q^*(\mathbf{x}), \mathbf{y}) - L(G^*(\mathbf{z}), \mathbf{z}, \mathbf{y})\big] \end{aligned} \tag{9}$$

For the unlabeled samples, we rely on the fact that the best guess of the label for a sample $\mathbf{x}$ is the one that minimizes $L(\mathbf{x}, \mathbf{z}, \mathbf{y})$ over the label space $\mathbf{y} \in \mathcal{Y}$. So the loss function for an unlabeled sample can be defined as

$$L_u(\mathbf{x}, \mathbf{z}) \triangleq \min_y L(\mathbf{x}, \mathbf{z}, \mathbf{y}) \tag{10}$$

We also update the $L(\mathbf{x}, \mathbf{z}, \mathbf{y})$ to $-\log \frac{\exp(a_y(\mathbf{x},\mathbf{z}))}{1+\sum_{y'} \exp(a_{y'}(\mathbf{x},\mathbf{z}))}$. Equipped with the new L_u, we can define the loss-sensitive objective for unlabeled samples as

$$\begin{aligned} &\mathcal{S}_u(L, G^*, Q^*) \triangleq \\ &\mathbb{E}_{\substack{\mathbf{x},\mathbf{y} \sim \mathbf{P}_{data}(\mathbf{x},\mathbf{y}) \\ \mathbf{z} \sim \mathbf{P}_Z(\mathbf{z})}} C\big[\Delta_{\mu,\nu}(\mathbf{x}, \mathbf{z}) + L_u(\mathbf{x}, Q^*(\mathbf{x}), \mathbf{y}) - L_u(G^*(\mathbf{z}), \mathbf{z}, \mathbf{y})\big] \end{aligned}$$

Like in the LSAL, G^* and Q^* can be found by solving the following optimization problems.

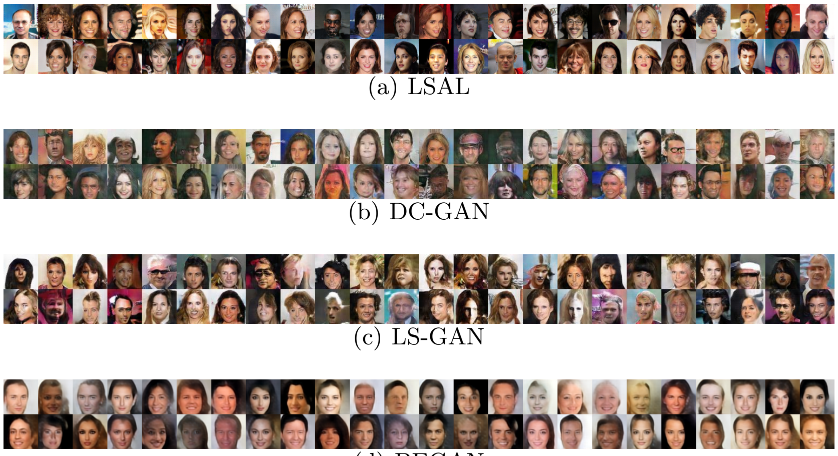
- Learning G with fixed L^* and Q^*:

$$G^* = \arg\min_G \mathcal{T}(L^*, G, Q^*) \triangleq \mathbb{E}_{\substack{\mathbf{y} \sim P_Y(\mathbf{y}) \\ \mathbf{z} \sim P_Z(\mathbf{z})}} L_u^*(G(\mathbf{z}), \mathbf{z}) + L^*(G(\mathbf{z}), \mathbf{z}, \mathbf{y})$$

- Learning Q with fixed L^* and G^*:

$$Q^* = \arg\max_Q \mathcal{R}(L^*, G^*, Q) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y} \sim P_{data}(\mathbf{x},\mathbf{y})} L_u^*(\mathbf{x}, Q(\mathbf{x})) + L^*(\mathbf{x}, Q(\mathbf{x}), \mathbf{y})$$

In experiments, we will evaluate the semi-supervised LSAL model in image classification task.



(a) LSAL

(b) DC-GAN

(c) LS-GAN

(d) BEGAN

Fig. 1. Generated samples by various methods. Size 64 × 64 on CelebA data-set. Best seen on screen.

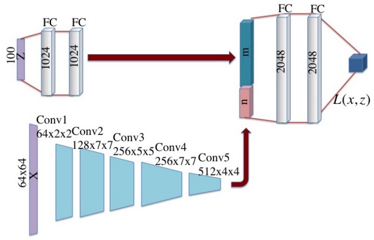


Fig. 2. Network architecture for the loss function $L(\mathbf{x}, \mathbf{z})$. All convolution layers have a stride of two to halve the size of their input feature maps.

6 Experiments

We evaluated the performance of the LSAL model on four datasets, namely Cifar10 [8], SVHN [10], CelebA [23] and 64 × 64 cropped center ImageNet [16]. We compared the image generation ability of the LSAL, both qualitatively and quantitatively, with other state-of-the-art GAN models. We also trained LSAL model in semi-supervised fashion for image classification task.

6.1 Architecture and Training

While this work does not aim to test new idea of designing architectures for the GANs, we adopt the exisiting architectures to make the comparison with other models as fair as possible. Three convnet models have been used to represent the generator $G(\mathbf{z})$, the pullback mapping $Q(\mathbf{x})$ and the loss function $L(\mathbf{x}, \mathbf{z})$. We use hinge loss as our cost function $C[\cdot] = \max(0, \cdot)$. Similar to DCGAN [14], we use strided-convolutions instead of pooling layers to down-sample feature maps and fractional-convolutions for the up-sampling purpose. Batch-normalization [5] (BN) also has been used before Rectified Linear (ReLU) activation function in the generator and pullback mapping networks while weight-normalization [18] (WN) is applied to the convolutional layers of the loss function. We also apply the dropout [20] with a ratio of 0.2 over all fully connected layers. The loss function $L(\mathbf{x}, \mathbf{z})$ is computed over the joint space $\mathcal{X} \times \mathcal{Z}$, so its input consists of two parts: the first part is a convnet that maps an input image $\mathbf{x}$ to an n-dim vector representation; the second part is a sequence of fully connected layers that successively maps the latent vector $\mathbf{z}$ to an m-dim vector too. Then an $(n+m)$-dim vector is generated by concatenation of these two vectors and goes further through a sequence of fully connected layers to compute the final loss value. For the semi-supervised LSAL, the loss function $L(\mathbf{x}, \mathbf{z}, \mathbf{y})$ is also defined over the label space $\mathcal{Y}$. In this case, the loss function network defined above can have multiple outputs, each for one label in $\mathcal{Y}$. The main idea of loss function network is illustrated in Figure 2. LSAL code also is available here.

The Adam optimizer has been used to train all of the models on four datasets. For image generation task, we use the learning rate of 10^{-4} and the first and second moment decay rate of $\beta_1 = 0.5$ and $\beta_2 = 0.99$. In the semi-supervised classification task, the learning rate is set to 6×10^{-4} and decays by 5% every 50 epochs till it reaches 3×10^{-4}. For both Cifar10 and SVHN datasets, the coefficient λ of unlabeled samples, and the hyper parameters μ and ν for manifold and ambient margins are chosen based on the validation set of each dataset. The L1-norm has been used in all of the experiments for both margins.

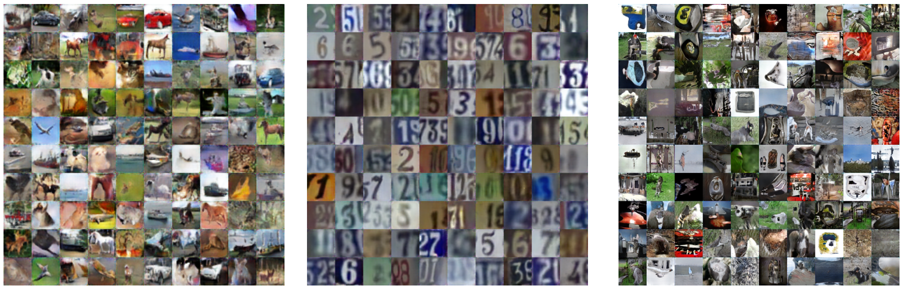
Model	Inception score
Real data	11.24 ± 0.12
ALI[3]	4.98 ± 0.48
LS-GAN[11]	5.83 ± 0.22
LSAL	6.43 ± 0.53

Table 1. Comparison of Inception score for various GAN models on Cifar10 data-set. Inception score of real data represents the upper bound of the score.

Finally, it is noted that, from theoretical perspective, we do not need to do any kind of pairing between generated and real samples in showing the distributional consistency in Section 4. Thus, we can randomly choose a real image rather than a "ground truth" counterpart (e.g., the most similar image) to pair with a generated sample. The experiments below also empirically demonstrate the random sampling strategy works well in generating high-quality images as well as training competitive semi-supervised classifiers.

6.2 Image Generation Results

Qualitative comparison: To show the performance of the LSAL model, we qualitatively compared the generated images by proposed model on CelebA dataset with other state of the art GANs models. As illustrated in Figure 1, the LSAL can produce details of face images as compared to other methods. Faces have well defined borders and nose and eyes have real shape while in LS-GAN 1(c) and DC-GAN 1(b) most of generated samples don't have clear face borders and samples of BEGAN model 1(d) lack stereoscopic features. Figure 3 shows the samples generated by LSAL for Cifar10, SVHN, and tiny ImageNet datasets. We also walk through the manifold space $\mathcal{Z}$, projected by the pullback mapping Q. To this end, pullback mapping network Q has been used to find the manifold representations $\mathbf{z}_1$ and $\mathbf{z}_2$ of two randomly selected samples from the validation set. Then G has been used to generate new samples for $\mathbf{z}$'s on the linear interpolate of $\mathbf{z}_1$ and $\mathbf{z}_2$. As illustrated in Figure 4, the transition between pairs of images are smooth and meaningful.



(a) Cifar10 (b) SVHN (c) Tiny ImageNet

Fig. 3. Generated samples by LSAL on different data-sets. Samples of (a) Cifar10 and (b) SVHN are of size 32 × 32. Samples of (c) Tiny ImageNet are of size 64 × 64.

Quantitative comparison: To quantitively assess the quality of LSAL generated samples, we used *Inception Score* proposed by [17]. We chose this metric as it had been widely used in literature so we can fairly compare with the other models directly. We applied the Inception model to images generated by various GAN models trained on Cifar10. The comparison of Inception scores on $50,000$ images generated by each model is reported in Table 1.

6.3 Semi-supervised classification

Using semi-supervised LSAL to train an image classifier, we achieved competitive results in comparison to other GAN models. Table 3 shows the error rate of the semi-supervised LSAL along with other semi-supervised GAN models when only $1,000$ labeled examples were used in training on SVHN with the other

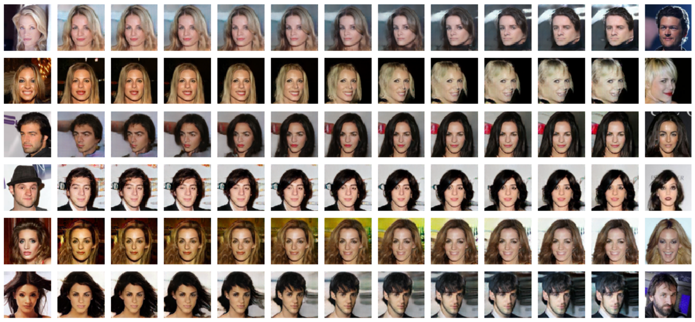


Fig. 4. Generated images for the interpolation of latent representations learned by pullback mapping Q for CelebA data-set. First and last column are real samples from validation set.

examples unlabeled. For Cifar10, LSAL was trained with various numbers of labeled examples. In Table 2, we show the error rates of the LSAL with $1,000$, $2,000$, $4,000$, and $8,000$ labeled images. The results show the proposed semi-supervised LSAL successfully outperforms the other methods.

# of labeled samples	1000	2000	4000	8000
Model		Classification error		
Ladder network[15]			20.40	
CatGAN [19]			19.58	
CLS-GAN[11]			17.3	
Improved GAN [17]	21.83 ± 2.01	19.61 ± 2.32	18.63 ± 2.32	17.72 ± 1.82
ALI[3]	19.98 ± 0.89	19.09 ± 0.44	17.99 ± 1.62	17.05 ± 1.49
LSAL	**18.83 ± 0.44**	**17.97 ± 0.74**	**16.22 ± 0.31**	**14.17 ± 0.62**

Table 2. Comparison of classification error on Cifar10

6.4 Trends of Ambient and Manifold Margins

We also illustrate the trends of ambient and manifold margins as the learning algorithm proceeds over epochs in Figure 5. The curves were obtained by training the LSAL model on Cifar10 with $4,000$ labeled examples, and both margins are averaged over mini-batches of real and fake pairs sampled in each epoch.

From the illustrated curves, we can see that the manifold margin continues to decrease and eventually stabilize after about 270 epochs. As manifold margin decreases, we find the quality of generated images continues to improve even though the ambient margin fluctuates over epochs. This shows the importance of

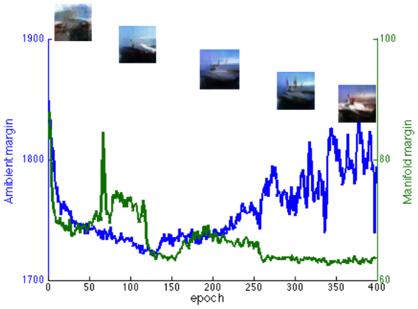


Fig. 5. Trends of manifold and ambient margins over epochs on the Cifar10 dataset. Example images are generated at epoch 10, 100, 200, 300, 400.

Model	Classification error
Skip deep generative model[9]	16.61 ± 0.24
Improved GAN [17]	8.11 ± 1.3
ALI[3]	7.42 ± 0.65
CLS-GAN[11]	5.98 ± 0.27
LSAL	**5.46 ± 0.24**

Table 3. Comparison of classification error on SVHN test set for semi-supervised learning using 1000 labeled examples.

manifold margin that motivates the proposed LSAL model. It also demonstrates the manifold margin between real and fake images should be a better indicator we can use for the quality of generated images.

7 Conclusion

In this paper, we present a novel regularized LSAL model, and justify it from both theoretical and empirical perspectives. Based on the assumption that the real data are distributed on a low-dimensional manifold, we define a pullback operator that maps a sample back to the manifold. A manifold margin is defined as the distance between the pullback representations to distinguish between real and fake samples and learn the optimal generators. The resultant model also demonstrates it can produce high quality images as compared with the other state-of-the-art GAN models.

Acknowledgement

The research was partly supported by NSF grant #1704309 and IARPA grant #D17PC00345. We also appreciate the generous donation of GPU cards by NVIDIA in support of our research.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (January 2017)
2. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016)
3. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., Courville, A.: Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
5. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456 (2015)
6. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems. pp. 3581–3589 (2014)
7. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
8. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
9. Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O.: Auxiliary deep generative models. arXiv preprint arXiv:1602.05473 (2016)
10. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
11. Qi, G.J.: Loss-sensitive generative adversarial networks on lipschitz densities. arXiv preprint arXiv:1701.06264 (January 2017)
12. Qi, G.J., Aggarwal, C.C., Huang, T.S.: On clustering heterogeneous social media objects with outlier links. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 553–562. ACM (2012)
13. Qi, G.J., Zhang, L., Hu, H., Edraki, M., Wang, J., Hua, X.S.: Global versus localized generative adversarial nets. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2018)
14. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
15. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. pp. 3546–3554 (2015)
16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
17. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2226–2234 (2016)
18. Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: Advances in Neural Information Processing Systems. pp. 901–909 (2016)

19. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv preprint arXiv:1511.06390 (2015)
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
21. Tang, J., Hua, X.S., Qi, G.J., Wu, X.: Typicality ranking via semi-supervised multiple-instance learning. In: Proceedings of the 15th ACM international conference on Multimedia. pp. 297–300. ACM (2007)
22. Valpola, H.: From neural pca to deep unsupervised learning. Adv. in Independent Component Analysis and Learning Machines pp. 143–171 (2015)
23. Yang, S., Luo, P., Loy, C.C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3676–3684 (2015)
24. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)
25. Zhu, X.: Semi-supervised learning. In: Encyclopedia of machine learning, pp. 892–897. Springer (2011)