QUALITY EVALUATION OF GANS USING CROSS LOCAL INTRINSIC DIMENSIONALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative Adversarial Networks (GANs) are an elegant mechanism for data generation. However, a key challenge when using GANs is how to best measure their ability to generate realistic data. In this paper, we demonstrate that an intrinsic dimensional characterization of the data space learned by a GAN model leads to an effective evaluation metric for GAN quality. In particular, we propose a new evaluation measure, CrossLID, that assesses the local intrinsic dimensionality (LID) of input data with respect to neighborhoods within GAN-generated samples. In experiments on 3 benchmark image datasets, we compare our proposed measure to several state-of-the-art evaluation metrics. Our experiments show that CrossLID is strongly correlated with sample quality, is sensitive to mode collapse, is robust to small-scale noise and image transformations, and can be applied in a model-free manner. Furthermore, we show how CrossLID can be used within the GAN training process to improve generation quality.

1 INTRODUCTION

Generative Adversarial Networks (GANs) are powerful models for data generation, composed of two neural networks, known as the 'generator' and the 'discriminator'. The generator maps random noise vectors to locations in the data domain in an attempt to approximate the distribution of the input data. The discriminator accepts a data sample and returns a decision as to whether or not the sample is from the input or was artificially generated. While the discriminator is trained to distinguish input samples from generated ones, the generator's objective is to deceive the discriminator by producing data that cannot be distinguished from input data. The two networks are jointly trained to optimize an objective function resembling a two-player minimax game.

GANs were first formulated by Goodfellow et al. (2014), and have been applied to tasks such as image generation (Denton et al., 2015; Radford et al., 2016) and image inpainting (Pathak et al., 2016). Despite their elegant theoretical formulation (Goodfellow et al., 2014), training of GANs can be difficult in practice due to instability issues such as vanishing gradients and mode collapse. The vanishing gradient problem occurs whenever gradients become too small to allow sufficient progress towards an optimization goal within the allotted number of training iterations. The latter occurs when the generator produces samples for only a limited number of data modes, without covering the full distribution of the input data.

Deployment of GANs is further complicated by the difficulty of evaluating the quality of their output. Researchers often rely on visual inspection of the quality of the generated samples, which is both time-consuming and subjective. Use of a quantitative quality metric is clearly desirable, and several such methods do exist (Goodfellow et al., 2014; Salimans et al., 2016; Odena et al., 2017; Che et al., 2017; Heusel et al., 2017; Karras et al., 2018; Lucic et al., 2017; Lopez-Paz & Oquab, 2017; Shmelkov et al., 2018). However, past research has identified various limitations of some existing metrics (Theis et al., 2016; Barratt & Sharma, 2018), and effective evaluation of GAN models is still an open issue.

In this paper, we show how the data space learned by a GAN model can be understood in terms of the *Local Intrinsic Dimensionality* (LID) model of distance distributions (Houle, 2013). LID assesses the number of latent variables (the intrinsic dimensionality) needed to characterize the distribution of distances to a reference point x — or equivalently, the discriminability of a distance measure in the vicinity of x. For the GAN discriminator, input samples are more likely to be discriminable

from generated samples if the learning process maps them into a space where the distance measure becomes more discriminable — or equivalently, one where the local intrinsic dimensionality is relatively low. For the GAN generator, a generated sample is more likely to be accepted as realistic if the learning process maps it into a local submanifold whose dimensionality matches that of its neighbors among the input data samples (after the mapping to the learned space). Our objective here is to develop this intuition into a technique for assessing the quality of the GAN learning process.

The main contributions of the paper are as follows:

- We propose CrossLID, a cross estimation technique based on LID that is capable of assessing the alignment of the data embedding learned by the GAN generator with that learned by the GAN discriminator.
- We show how CrossLID can be employed to avoid mode collapse during GAN training, to identify classes for which some or all modes are not well-covered by the learning process. We also show how this knowledge can then be used to bias the GAN discriminator via an oversampling strategy so as to improve its performance on such classes.
- We provide experimentation showing that our proposed CrossLID measure is well correlated with GAN sample quality, and performs very strongly when compared to other state-of-the-art evaluation measures.

2 Evaluation Metrics for GAN Models

GAN-based learning is an extensively researched area. Here, we briefly review the topic most relevant to our work, evaluation metrics for GAN models. Past research has employed several different metrics, including log-likelihood measures (Goodfellow et al., 2014), the Inception score (Salimans et al., 2016), the MODE score (Che et al., 2017), Kernel MMD (Gretton et al., 2006), the MS-SSIM index (Odena et al., 2017), the Fréchet Inception Distance (Heusel et al., 2017), the sliced Wasserstein distance (Karras et al., 2018), and Classifier Two-Sample Tests (Lopez-Paz & Oquab, 2017). In our study, we focus on the two most widely used metrics for image data, the Inception score (IS) and the Fréchet Inception Distance (FID), as well as a recently proposed measure, the Geometry Score (GS) (Khrulkov & Oseledets, 2018).

The Inception score uses an associated Inception classifier (Szegedy et al., 2016) to extract output class probabilities for each image, and then computes the Kullback-Leibler (KL) divergence of these probabilities with respect to the marginal probabilities of all classes:

$$\text{IS} = \exp(\mathbb{E}_{x \sim p_G} D_{\text{KL}}(p(y|x)||p(y)), \tag{1}$$

where $x \sim p_G$ implies a sample x drawn from the generator outputs, $p(y|x)$ is the probability of class y being assigned to x by the Inception classifier, $p(y) = \int_x (p(y|x)dx$ is the marginal class distribution, and D_{KL} is the KL divergence. IS measures two aspects of a generative model: 1) the images generated should be both clear and highly distinguishable by the classifier, as indicated by low entropy of $p(y|x)$ when marginalized over y, and 2) all classes should have good representation over the set of generated images, which can be indicated by high entropy of $p(y|x)$ when marginalized over x. However, a recent study has shown that IS is susceptible to variations in the Inception network weights when trained on different platforms (Barratt & Sharma, 2018).

The Fréchet Inception Distance (FID) passes both input and generated images to an Inception classifier, and extracts activations from an intermediate pooling layer. The activations are assumed to follow a multidimensional Gaussian parameterized by their means and covariances. FID is defined as

$$\text{FID} = (||\mu_I - \mu_G||)_2^2 + \text{Tr}(\Sigma_I + \Sigma_G - 2((\Sigma_I \Sigma_G)^{\frac{1}{2}})), \tag{2}$$

where (μ_I, Σ_I) and (μ_G, Σ_G) represent the mean and covariance of activations for input and generated data samples, respectively. Compared to IS, FID has been shown to more consistent with human judgment and more robust to noise; however, it also requires an external Inception classifier for its calculation (Heusel et al., 2017).

A recently proposed metric, the Geometry Score (GS) (Khrulkov & Oseledets, 2018) assesses the conformity between manifolds of input and generated data, in terms of the persistence of certain

topological properties in a manifold approximation process. The topological relationships are extracted in terms of the counts of 1-dimensional loops in a graph structure built up from proximity relationships as a distance threshold is increased. Although it may be indirectly sensitive to variations in the dimensionality of the manifolds, the Geometry Score (GS) explicitly rewards only matches in terms of the specific topology of these loop structures in approximations of the manifolds. However, due to its strictly topological nature, GS is insensitive to differences in relative embedding distances or orientations within the manifold — this issue is acknowledged by the authors, who advise that GS would be best suited for use in conjunction with other metrics (Khrulkov & Oseledets, 2018).

3 Local Intrinsic Dimensionality

In its most general sense, the Local Intrinsic Dimensionality (LID) is a characteristic of smooth functions that vanish at zero. The LID value can be regarded as the degree of the polynomial with the best fit to the function, taken over an infinitesimally small domain that includes the origin.

Definition of LID: To formalize LID, let F be a function that is positive and continuously differentiable over some open interval containing $r > 0$. The LID of F at r is defined as:

$$\mathrm{LID}_F(r) := r\frac{F'(r)}{F(r)} = \lim_{\epsilon \to 0^+} \frac{\ln\left(F((1+\epsilon)r)/F(r)\right)}{\ln\left(1+\epsilon\right)} = \lim_{\epsilon \to 0^+} \frac{F((1+\epsilon)r) - F(r)}{\epsilon\, F(r)}, \tag{3}$$

wherever the limits exists. The local intrinsic dimensionality of F is then:

$$\mathrm{LID}^*_F = \lim_{r \to 0^+} \mathrm{LID}_F(r). \tag{4}$$

In our context, and as originally proposed in (Houle, 2013), we are interested in functions that are the distributions of distances induced by some global distribution of data points: for each data sample generated with respect to the global distribution, its distance to a predetermined reference point determines a sample from the local distance distribution.

The LID model has the interesting property that the definition can be motivated in two different ways. The first limit stated in the definition follows from a modeling of the growth of probability measure in a small expanding neighborhood of the origin: as the radius r increases, the volume of data encountered can be expected to grow proportionally to the r to the power of the intrinsic dimension. In the setting of a uniform distribution with a manifold of dimension m, if F is the distribution of distances to a reference point in the relative interior of the manifold, then $\mathrm{LID}^*_F = m$.

The second limit expresses the (in)discriminability of F when interpreted as a distance measure evaluated at distance r (with low values of $\mathrm{LID}_F(r)$ indicating higher discriminability). As implied by Eq. 3, the LID framework is extremely convenient in that the local intrinsic dimensionality and the discriminability of distance measures are shown to be equivalent and interchangeable concepts. For more information on the formal definition of LID and its properties, see (Houle, 2017a;b).

Estimating LID: LID is a generalization of pre-existing expansion-based measures which implicitly use neighborhood set sizes as a proxy for probability measure. These earlier models include the expansion dimension (Karger & Ruhl, 2002) and its variants (Houle et al., 2012a), and the minimum neighbor distance (MinD) (Rozza et al., 2012), all of which have been shown to be crude estimators of LID (Amsaleg et al., 2018). Although the popular estimator due to Levina & Bickel (2005) can be regarded as a smoothed version of LID, its derivation depends on the assumption that the observed data can be treated as a homogeneous Poisson process. However, the only assumptions made by the LID model is that the underlying (distribution) function be continuously differentiable.

For this work, we estimate LID using the Maximum Likelihood Estimator (MLE) as proposed in (Amsaleg et al., 2018), due to its ease of implementation and its convergence properties. Given a set of data points X, and a distinguished data sample x, the MLE estimator of LID is:

$$\mathrm{LID}\left(x; X\right) = -\Big(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(x;X)}{r_{max}(x;X)}\Big)^{-1}. \tag{5}$$

where k is the neighborhood size, $r_i(x;X)$ is the distance from x to its i-th nearest neighbor in $X \setminus \{x\}$, and $r_{\max}(x;X)$ denotes the maximum distance within the neighborhood (which by convention can be $r_k(x;X)$). Due to the deep equivalence between the LID model and the statistical theory

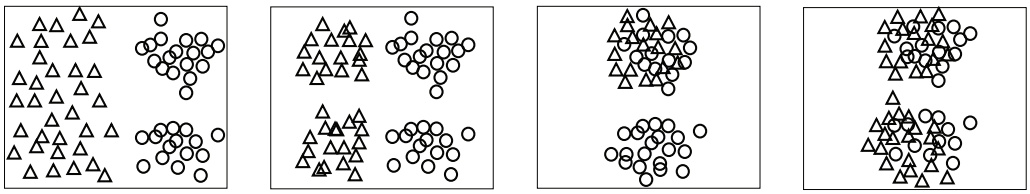
of extreme values (EVT) shown in (Houle, 2017a; Amsaleg et al., 2018), the MLE estimator of LID coincides with the well-known Hill estimator derived from EVT (Hill, 1975). Note that in these estimators, no explicit knowledge of the underlying function F is needed - this information is implicit in the distribution of neighbor distances themselves.

Estimation of LID characteristics has seen practical applications for assessing the complexity of search queries in approximate similarity search (Houle et al., 2012b; Houle & Nett, 2015; Casanova et al., 2017), as a measure of the outlierness of data (Houle et al., 2018) and in the detection of adversarial samples for deep neural networks (Ma et al., 2018a). LID has also been used to detect and prevent overfitting by DNN classifiers on datasets with noisy labels (Ma et al., 2018b).

LID can characterize the intrinsic dimensionality of the data submanifold in the vicinity of a distinguished point x. The $\text{LID}(x; X)$ values of all data samples x from a dataset X can thus be averaged to characterize the overall intrinsic dimensionality of the manifold within which X resides. In Romano et al. (2016); Houle (2017b), it was shown that this type of average is in fact an estimator of the correlation dimension over the sample domain (or manifold). Henceforth, whenever the context set X is understood, we will use the simplified notation $\text{LID}(x)$ to refer to $\text{LID}(x; X)$, and to denote the average of these estimates over all $x \in X$ by $\text{LID}(X)$.

4 Evaluating GANs via Cross Local Intrinsic Dimensionality

We propose a new measure, CrossLID, that evaluates the closeness of the underlying submanifolds of an input distribution p_I and a GAN-generated distribution p_G, as derived from the profiles of distances from samples of one distribution to samples of the other distribution. Our intuition is that if two distributions are similar, then the distance profiles of a generated sample with respect to a neighborhood of input samples should conform with the profiles observed within the input sample, and vice versa. As an illustration of the possible relationships between a generated distribution and the original input distribution, Fig. 1 shows four examples of how a GAN model could learn a bimodal Gaussian distribution. Decreasing CrossLID scores indicate an increasing conformity between the two-mode input data distribution and the generated data distribution.



(a) CrossLID = 15.16 (b) CrossLID = 7.33 (c) CrossLID = 4.78 (d) CrossLID = 2.10

Figure 1: Four examples showing how GAN-generated data samples (triangles) could relate to a bimodal Gaussian-distributed data set (circles), together with CrossLID scores: (a) generated data distributed uniformly, spatially far from the input data; (b) generated data with two modes, spatially far from the input data; (c) generated data associated with only one mode of the input data; and (d) generated data associated with both modes of the input data (the desired situation).

4.1 CrossLID for GAN Model Evaluation

We generalize the single data distribution based LID metric defined in Eq. 3 to a new metric that measures the cross LID characteristics between two distributions. Given two sets of samples A and B, the CrossLID of samples in A with respect to B is defined as:

$$\text{CrossLID}(A; B) = \mathbb{E}_{x \in A} \text{LID}(x; B). \tag{6}$$

Note that CrossLID is not symmetric, in that $\text{CrossLID}(A; B)$ does not necessarily equal $\text{CrossLID}(B; A)$.

Low $\text{CrossLID}(A; B)$ scores indicate low average spatial distance between the elements of A and their neighbor sets in B. To see this, consider the situation in which a positive correction d is

added to each of the distances from some reference sample $x \in A$ with respect to its neighbors in B. This distance correction would cause the LID estimate defined in Eq. 5 to become $-1/(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i+d}{r_{\max}+d})$, which is an increase over the original estimate LID — this follows from the observation that $1 \geq (r_i+d)/(r_{\max}+d) > r_i/r_{\max} > 0$. In contrast, if the correction d were allowed to go negative, LID(x) would decrease. Thus, a good alignment between A and B is revealed by good discriminablity (low LID) of the distance distributions induced by one set (B) relative to the members of the other (A). In general, CrossLID differs from LID in its sensitivity to differences in spatial position and orientation of the respective manifolds within which A and B reside (see Appendix A for more details).

Low values of CrossLID$(A; B)$ also indicate good coverage of the domain of A by elements of B. To see why, consider what would happen if this were not the case: if the samples in B did not provide good coverage of all modes of the underlying distribution of A, there would be a significant number of samples in A whose distances to its nearest neighbors in B would be excessively large in comparison to an alternative set B' providing better coverage of A (see Figs. 1c and 1d for an example). As discussed above, this increase in the distance profile would likely lead to an increase in many of the individual LID estimates that contribute to the CrossLID score.

Given a set of samples X_I from a GAN input distribution, and a set of samples X_G from the GAN generated distribution, a low value of CrossLID$(X_I; X_G)$ indicates a good alignment between the manifold associated with X_G and the manifold associated with X_I, as well as an avoidance of mode collapse in the generation of X_G. It should be noted, however, that low values of CrossLID$(X_G; X_I)$ do not discourage mode collapse. Since low values of CrossLID$(X_I; X_G)$ encourage a good integration of generated data into the submanifolds with respect to these learned representations, and an avoidance of mode collapse in sample generation, CrossLID$(X_I; X_G)$ is a good candidate measure for evaluating GAN learning processes.

CrossLID also allows targeted quality assessment of GANs for refined sample groups of interest. For example, for a specific mode (X_I^m) from the input samples based on either cluster information or class information, CrossLID$(X_I^m; X_G)$ can be used to assess how well the GAN model learns the submanifold of this particular mode. CrossLID can therefore be exploited to detect and mitigate underlearned modes in GAN training. We will explore this further in Section 5.

4.2 Effective Estimation of CrossLID

We next discuss two important aspects in CrossLID estimation: 1) the choice of feature space where CrossLID is computed and 2) the choice of appropriate sample and neighborhood sizes for accurate and efficient CrossLID estimation.

Deep Feature Space for CrossLID Estimation: The representations that define the underlying manifold of a data distribution are well learned in the deep representation space. Recent work in representation learning (Goodfellow et al., 2016), adversarial detection (Ma et al., 2018a) and noisy label learning (Ma et al., 2018b) has shown that DNNs can effectively map high-dimensional inputs to low-dimensional submanifolds at different intermediate layers of the network. We denote the output of such a layer as a function $f(x)$, and estimate CrossLID in the deep feature space as follows:

$$\text{CrossLID}(f(X_I); f(X_G)) = \frac{1}{|X_I|}\sum_{x\in X_I} -\Big(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i(f(x), f(X_G))}{r_{\max}(f(x), f(X_G))}\Big)^{-1}. \tag{7}$$

It should be noted that successful learning by the GAN discriminator would entail the learning of a mapping f for which the intrinsic dimensionality of $f(X_I)$ is relatively low. This encourages the GAN generator to produce samples for which CrossLID$(f(X_I); f(X_G))$ is also low, which further enhances the value of CrossLID in GAN evaluation and training.

The transformation $f(x)$ can be computed using an external network trained separately on the real data distribution, such as the Inception network used by Inception score and FID. It can also be the discriminator network of a GAN, as they are known to be capable of learning quality representations suitable for classification (Radford et al., 2016). In Section 6.1 we will show that both choices work well for the estimation of CrossLID.

Sample Size and Neighborhood for CrossLID Estimation: Searching for the k-nearest neighbors of all input samples of X_I within the entire GAN-generated dataset X_G can be prohibitively expensive. Although most input data sets require neighborhood sizes on the order of $k = 100$ for the convergence of the LID estimators (Amsaleg et al., 2018), previous work using the LID measure in adversarial detection (Ma et al., 2018a) and noisy label learning (Ma et al., 2018b) has demonstrated that LID estimation at the deep feature level can be effectively performed within small batches of training samples — with neighborhood sizes as small as $k = 20$ drawn from batches of 100 samples. For the estimation of CrossLID$(f(X_I); f(X_G))$, we use $|X_I| = 20000$ input samples and $|X_G| = 20000$ GAN-generated samples. For each input sample $f(x)$ where $x \in X_I$, we search $k = 100$ nearest neighbours within 1000 samples selected randomly from $f(X_G)$, and use the distances from $f(x)$ to these $k = 100$ nearest neighbors to estimate the CrossLID$(f(x); f(X_G))$. The mean of the CrossLID estimates over all 20000 input samples determines the final estimate of the overall CrossLID score.

5 Oversampling in GAN training with Mode-wise CrossLID

So far we have introduced a new metric, CrossLID, for the evaluation of GAN performance. Here, we further show that a mode-wise computation of CrossLID scores can be exploited during GAN training to monitor and improve GAN models.

A GAN distribution may not equally capture the distributions of all modes presenting in a real data distribution. Due to the inherent randomness in stochastic learning, the decision boundary of the discriminator may be closer to regions of some modes than others at different stages of the training process. The closer modes may get stronger gradients, and the generator will learn these modes better than the others. If imbalances in learning can be detected and addressed during training, we might expect to achieve a better convergence to good solutions. To achieve this, we propose a GAN training strategy with oversampling based on mode-wise CrossLID scores (as defined in section 4.1).

We describe our training strategy in the context of labeled data, where we simply take the classes as the modes. Note that for unlabeled data, the modes can be determined by computing clusters within the data. Here, we compute the average CrossLID score for samples from each class, and use it to measure how well a class has been learned — the lower CrossLID score, the more effective the learning. To help generate good gradients for all classes during the training, we dynamically modify the input of the discriminator by oversampling the poorly learned classes (those with high class-wise CrossLID scores). The objective is to bias the discriminator's decision boundary towards the regions of poorly learned classes in order to produce stronger gradients for the generator in favor of these underlearned classes.

The steps are described in Algorithm 1. For each class $c \in \{1, \cdots, C\}$, we randomly select a subset of input samples $X_c^{'}$ from that class, of size proportional to a deviation factor $\gamma_c = (\text{CrossLID}(X_I^c; X_G) - \text{CrossLID}(X_I^c; X_I^c))/\text{CrossLID}(X_I^c; X_I^c)$, and augment the original input dataset with the members of $X_c^{'}$ for subsequent training. γ_c measures the relative deviation of the CrossLID score CrossLID$(X_I^c; X_G)$ from the self-CrossLID score CrossLID$(X_I^c; X_I^c)$, which simply indicates the correlation dimensional of the class input data distribution. When the GAN model has already fully learned the distribution of a given class (that is, when CrossLID$(X_I^c; X_G) =$ CrossLID$(X_I^c; X_I^c)$), $\gamma_c = 0$, indicating that no oversampling will be applied to this class.

Our proposed strategy can effectively deal with the mode collapse issues encountered in GAN training. When the generator learns a class only partially, or not at all, it will receive a relatively high CrossLID scores for that class. In subsequent iterations, the imbalance in learning will be addressed by our oversampling step in favor of these classes (Step 8 in Algorithm 1).

6 Experimental Results

6.1 Evaluation of CrossLID as a GAN Quality Metric

We first demonstrate that CrossLID score is well correlated with sample quality of GANs. We then assess CrossLID with respect to the following five aspects: 1) sensitivity to mode collapse, 2) robustness to input noise, 3) robustness to image transformations, 4) robustness to sample size

Algorithm 1 Oversampling in GAN training with mode-wise CrossLID

for every T generator iterations **do**
  Generate N_1 GAN samples X_G.
  for c in $\{1, \cdots, C\}$ **do** ▷ C: number of classes.
    Sample N_2 input samples X_I^c from class c
    $\gamma_c = (\text{CrossLID}(X_I^c; X_G) - \text{CrossLID}(X_I^c; X_I^c))/\text{CrossLID}(X_I^c; X_I^c)$
  end for
  $\gamma_c = \gamma_c / \sum_{j=1}^{C} \gamma_j$, for $c \in \{1, \ldots, C\}$. ▷ Normalization for next step oversampling.
  $X_{\text{aug}} = \{X_1^{'}, \cdots, X_C^{'}\} \cup X_I$ where $X_c^{'}$ is a random sample from X_I^c of size $|X_c^{'}| = m \times \gamma_c$ and m is a size parameter
  Continue GAN training with X_{aug} for the next T generator iterations.
end for

used for estimation, and 5) dependency on external models. We also compare CrossLID score with Geometry score, Inception score and FID, on MNIST (LeCun et al., 1990), CIFAR-10 (Krizhevsky & Hinton, 2009) and SVHN (Netzer et al., 2011) datasets.

For CrossLID score, we used external CNNs trained on the original training set of real images for feature extraction (see Appendix H.2 for more details), except for evaluation scenario 5). To compute Inception score, we followed Salimans et al. (2016) using the pretrained Inception network except for MNIST, for which we pretrained a different CNN model as in Li et al. (2017). FID scores were computed as in Heusel et al. (2017). Our code is available at `https://www.dropbox.com/s/bqadqzr5plc6xud/CrossLIDTestCode.zip`.

Correlation of CrossLID and Sample Quality of GANs: We show CrossLID score is strongly correlated with sample quality of GAN models. In the left three subfigures of Fig. 2, as GAN training proceeds, the data manifold learned approaches that of the real distribution and CrossLID score decreases. CrossLID$(X_I; X_G)$ was estimated over 20000 generated samples using deep features extracted from the external CNN model. The rightmost subfigure in Fig. 2 illustrates the negative correlation between CrossLID score and Inception score over different training epochs. Supporting images for visual verification of correlation between CrossLID and sample quality are in Appendix D. We also found that Geometry score did not exhibit a clear correlation with sample quality, which is consistent with their claims in the paper (Khrulkov & Oseledets, 2018) (see Appendix B). Therefore, we did not include Geometry score in the rest of comparisons.

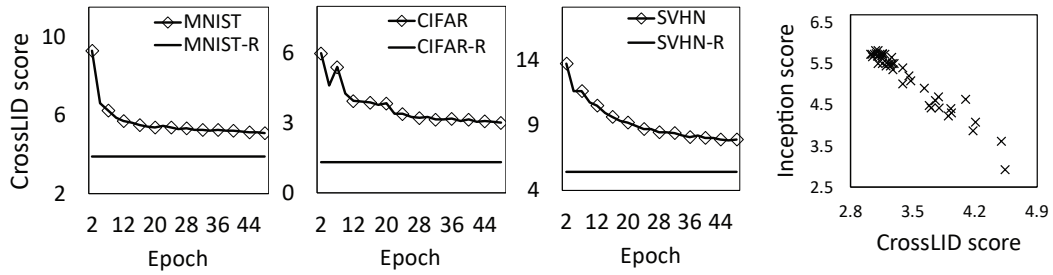


Figure 2: *Left three*: The CrossLID$(X_I; X_G)$ score for generated samples by a DCGAN after different training epochs. Results are shown for the first 50 epochs of training on MNIST, CIFAR-10 and SVHN, and MNIST-R/CIFAR-R/SVHN-R denote the CrossLID$(X_I; X_I)$ scores for real MNIST/CIFAR-10/SVHN samples. *Rightmost*: Correlation between CrossLID score and Inception score for CIFAR-10 dataset, each point is associated with a model at a certain epoch.

Sensitivity to mode collapse: A challenge of GAN training is to overcome mode collapse where the generated samples cover only a limited number of modes, not necessarily from the real distribution, instead of learning the entire real data distribution. An effective evaluation metric for GANs should be sensitive to such situations.

We simulate two types of mode collapse by downsampling the training data: 1) intra-class mode dropping and 2) inter-class mode dropping. 1) is when the GAN generates samples covering all classes, but the diversity of samples within each class is low, while 2) is when the GAN generates samples from a limited number of classes. For both 1) and 2), we randomly select a subset of samples ($n < N$ samples of c classes) from the original training set (N samples of C classes), then randomly sample with replacement from the subset to create a new dataset with the same number of N samples as in the original training set. Particularly, for 1), we let $c = C$ and vary $n \in [30, 40, 50, 70, 100]$, while for 2), we let $n = 50$ and vary $c \in [2, 4, 6, 8, 10]$. Overall, we created five new datasets for each type of mode collapse and each dataset, and computed the three scores on the new datasets.

In Fig. 3, CrossLID is sensitive to different degrees of intra-class mode dropping, but Inception score failed to identify intra-class mode dropping on MNIST and SVHN and responded inconsistently for different levels of intra-class mode dropping on CIFAR-10. FID is also sensitive to intra-class mode dropping. Fig. 4 shows that CrossLID is sensitive to increasing levels of inter-class mode-dropping, and is more sensitive than FID. However, Inception score in general fails to characterize this on CIFAR-10 and SVHN, but can identify inter-class mode-dropping on MNIST.

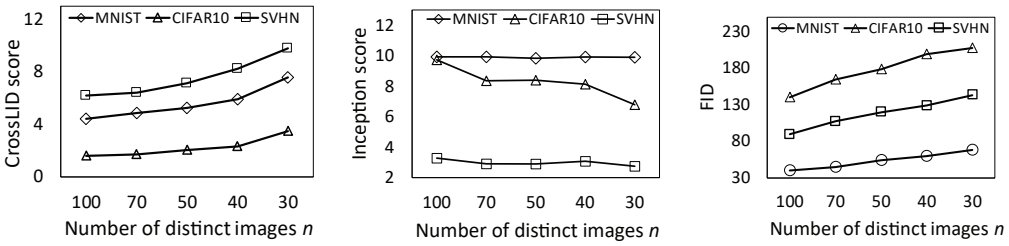


Figure 3: Test results for intra-class mode dropping: The CrossLID scores (left), Inception scores (middle) and FID scores (right) on varying numbers of unique samples in the datasets.

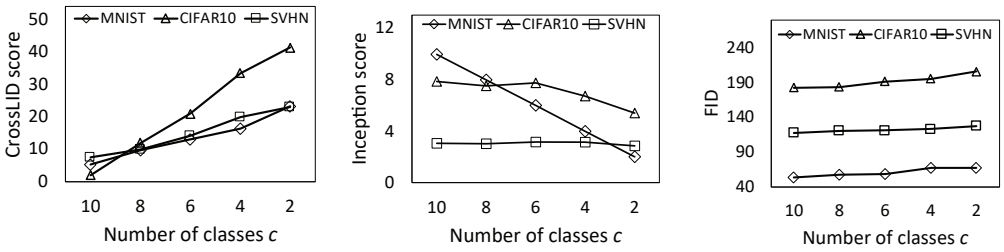


Figure 4: Test results for inter-class mode dropping: The CrossLID scores (left), Inception scores (middle) and FID scores (right) on varying numbers of unique classes in the datasets.

Robustness to small input noise: We further examine the robustness of the three scores to small levels of Gaussian input noise. We add input noise drawn from a Gaussian distribution with both mean and variance equal to 127.5 (255/2) to a certain proportion of pixels in the original images. In Fig. 5, CrossLID exhibited proportionate variation as the proportion of changed pixels increased from 0.2% to 2%. In contrast, Inception scores and FID both demonstrated large variations, particularly for CIFAR-10 and SVHN. We also investigated the robustness of the three metrics against salt-and-pepper noise which can be found in Appendix E. We observed stable and consistent performance of CrossLID, and degraded performance for Inception score and FID.

Robustness to input transformation: We test the robustness of the measures to small input transformations. As long as the transformations do not alter the visual appearance of GAN images, a robust metric should be able to give consistent evaluations. This is important as GAN generated images often exhibit small distortions compared to natural images, and such small imperfections should not significantly down-rate the quality of GANs. However, as demonstrated in the left and middle subfigures of Fig. 6, while CrossLID and Inception score are moderately robust to small translations and rotations on CIFAR-10 images, FID score increased significantly with increasing degrees of transformations.

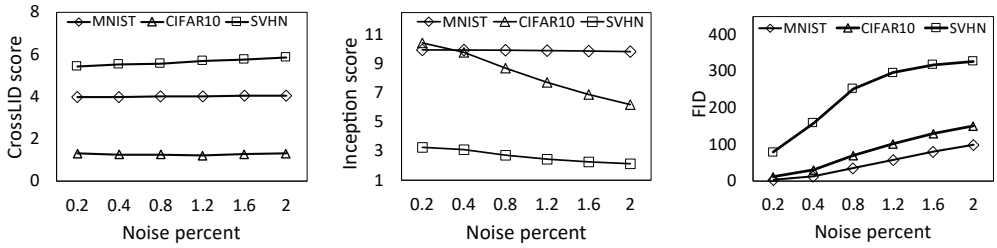


Figure 5: Robustness to Gaussian noise for CrossLID, Inception score and FID score (from left to right). Noise percent indicates proportion of pixels of GAN images that have been added noise.

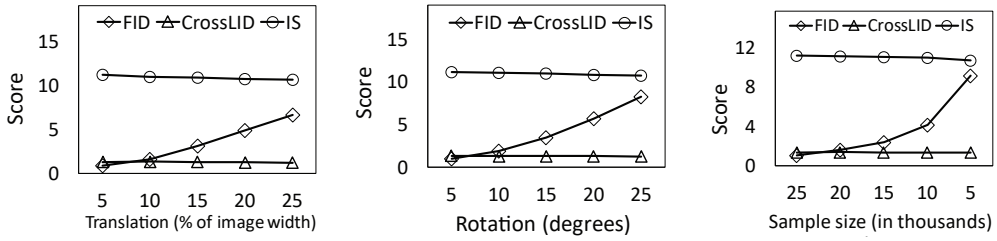


Figure 6: Robustness test of the three metrics on CIFAR-10 dataset to small image transformations including translation (left) and rotation (middle), and sample size used for calculation (right).

Robustness to sample size: We test the robustness of the three metrics to the sample size used for calculation. A good measure should accurately assess the quality of GANs when there are few GAN samples available. We select a subset of CIFAR-10 training images and compute the three scores for this subset. Results are in the right subfigure of Fig. 6. CrossLID and the Inception score are stable as subset size decreases from 25k to 5k. FID score is very sensitive to sample size. The lower variation of CrossLID against sample size enables us to reliably compute it with a smaller sample size (e.g., 20k) compared to what is typically used by the other two metrics (e.g., 50k).

Dependency on external model: So far we have evaluated CrossLID score using a separately trained CNN classifier as the feature extractor for its estimation in deep feature space. Recent work has demonstrated that the discriminator network of a GAN model is capable of learning class-independent features that are suitable for classification (Radford et al., 2016). Next, we show that our proposed CrossLID score can be accurately estimated directly using the feature space of the discriminator network, and an external feature extractor is not necessary. We compute CrossLID scores of different DCGANs (Radford et al., 2016) trained on MNIST, CIFAR-10, and SVHN, based on the discriminator's outputs at the second-to-last layer. We train all GANs for 50 epochs and report the CrossLID score. The CrossLID scores computed using the discriminator features follow a correlation with training epochs similar to the CrossLID scores computed with an external feature extractor in Fig 2. The results are reported in Appendix F. This indicates that CrossLID may be computed without a pre-trained model like Inception score or FID and so potentially can be used to evaluate GANs trained on non-image data or unlabeled data.

Summary of comparison: Table 1 summarizes our comparison results.

Table 1: Evaluation results of CrossLID, Inception score and FID.

EVALUATION CRITERIA	CrossLID score	Inception score	FID
Sensitivity to mode collapse.	High	Low	High
Robustness to small input noise.	High	Low	Low
Robustness to small input transformations.	Moderate	Moderate	Low
Robustness to sample size for estimation.	High	Moderate	Low
Dependency on external model.	Optional	Dependent	Dependent

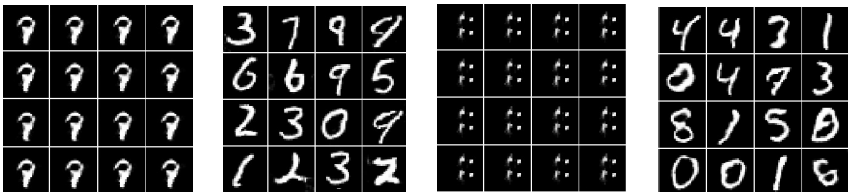
6.2 EVALUATION OF THE PROPOSED GAN TRAINING WITH OVERSAMPLING

We evaluate the effectiveness of CrossLID guided oversampling approach in GAN training. We choose two popular GAN models including DCGAN (Radford et al., 2016) and WGAN (Arjovsky et al., 2017), and assess the benefit of CrossLID guided oversampling on top of standard training. We compare, on MNIST, CIFAR-10 and SVHN datasets, DCGAN and WGAN models trained using standard training to the same models trained with CrossLID guided oversampling. Further details of model architecture, experimental settings, and output images can be found in Appendix H.1.

The performances in terms of CrossLID score and Inception score are reported in Table 2, where DCGAN+ and WGAN+ refer to training with our proposed oversampling. Our training approach achieved better CrossLID scores than standard training for both DCGANs and WGANs. Note that Inception scores correlate well with CrossLID scores except on SVHN.

Table 2: Performance of CrossLID guided oversampling on DCGAN and WGAN

	DCGAN				WGAN			
	CrossLID Score (Lower is better)		Inception score (Higher is better)		CrossLID Score (Lower is better)		Inception score (Higher is better)	
	DCGAN	DCGAN+	DCGAN	DCGAN+	WGAN	WGAN+	WGAN	WGAN+
MNIST	5.11	**4.97**	8.65 ± 0.03	$\mathbf{8.76 \pm 0.01}$	8.42	**8.24**	6.40 ± 0.06	$\mathbf{6.62 \pm 0.05}$
CIFAR-10	3.00	**2.75**	6.14 ± 0.09	$\mathbf{6.32 \pm 0.09}$	3.65	**3.52**	5.46 ± 0.03	$\mathbf{5.70 \pm 0.06}$
SVHN	7.36	**7.09**	$\mathbf{3.03 \pm 0.03}$	3.01 ± 0.02	10.11	**9.88**	$\mathbf{2.90 \pm 0.02}$	2.89 ± 0.01



(a) DCGAN (b) DCGAN+ (c) DCGAN (d) DCGAN+

Figure 7: Images generated at the end of the 30-th epoch by DCGAN and DCGAN with our proposed oversampling approach (DCGAN+) on MNIST dataset, with (a-b) no batch normalization in the generator, and (c-d) no batch normalization in neither the generator nor discriminator

Effectiveness in Preventing Mode Collapse: As explained in section 5, our approach can help avoid mode collapse. We next show on MNIST, when the batch-normalization (BN) layers were removed from the generator or from both the discriminator and the generator, standard DCGAN training suffered significant mode collapse and failed to learn the full real distribution, as shown in Fig. 7 (a) and (c). Our approach, however, was still able to produce high quality images without any sign of mode collapse during the training, as shown in Fig. 7 (b) and (d). More details of the training process and visual inspections can be found in Appendix G.

7 CONCLUSION

We have proposed a new metric for quality evaluation of GANs, based on cross local intrinsic dimensionality (CrossLID). Our measure can effectively assess sample quality and mode collapse in GAN outputs. It is reasonably robust to input noise, image transformations, and sample size. We also demonstrated a simple oversampling approach based on the mode-wise CrossLID that can improve GAN training and help avoid mode collapse. We believe CrossLID is not only a promising new tool for assessing the quality of GANs, but also a promising tool to help improve GAN training strategies.

REFERENCES

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken-ichi Kawarabayashi, and Michael Nett. Extreme-value-theoretic estimation of local intrinsic dimensionality. *Data Mining and Knowledge Discovery*, 32(6):1768–1805, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

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

Guillaume Casanova, Elias Englmeier, Michael E. Houle, Peer Kröger, Michael Nett, and Arthur Zimek. Dimensional testing for reverse k-nearest neighbor search. *PVLDB*, 10(7):769–780, 2017.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *ICLR*, 2017.

Emily L Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pp. 1486–1494, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Arthur Gretton, Karsten Borgwardt, Malte J Rasch, Bernhard Scholkopf, and Alexander J Smola. A kernel method for the two-sample problem. In *NIPS*, 2006.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*, pp. 6626–6637, 2017.

Bruce M. Hill. A simple general approach to inference about the tail of a distribution. *Annals of Statistics*, 3(5):1163–1174, 1975.

Michael E Houle. Dimensionality, discriminability, density and distance distributions. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, pp. 468–473. IEEE, 2013.

Michael E. Houle. Local intrinsic dimensionality I: an extreme-value-theoretic foundation for similarity applications. In *International Conference on Similarity Search and Applications*, pp. 64–79. Springer, 2017a.

Michael E. Houle. Local intrinsic dimensionality II: multivariate analysis and distributional support. In *International Conference on Similarity Search and Applications*, pp. 80–95. Springer, 2017b.

Michael E Houle and Michael Nett. Rank-based similarity search: Reducing the dimensional dependence. *IEEE trans. on pattern analysis and machine intelligence*, 37(1):136–150, 2015.

Michael E. Houle, Hisashi Kashima, and Michael Nett. Generalized expansion dimension. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pp. 587–594. IEEE, 2012a.

Michael E. Houle, Xiguo Ma, Michael Nett, and Vincent Oria. Dimensional testing for multi-step similarity search. In *ICDM*, pp. 299–308, 2012b.

Michael E. Houle, Erich Schubert, and Arthur Zimek. On the correlation between local intrinsic dimensionality and outlierness. In *11th International Conference on Similarity Search and Applications*. Springer, 2018.

David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proc. 34th Annual ACM Symposium on Theory of Computing*, pp. 741–750. ACM, 2002.

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

Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. In *ICML*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pp. 396–404, 1990.

Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *NIPS*, pp. 777–784, 2005.

Chunyuan Li, Hao Liu, Changyou Chen, Yunchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *NIPS*, 2017.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *ICLR*, 2017.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Michael E. Houle, Grant Schoenebeck, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018a.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. *ICML*, 2018b.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pp. 2536–2544, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

Simone Romano, Oussama Chelly, Vinh Nguyen, James Bailey, and Michael E Houle. Measuring dependency via intrinsic dimensionality. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 1207–1212. IEEE, 2016.

Alessandro Rozza, Gabriele Lombardi, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. Novel high intrinsic dimensionality estimators. *Machine learning*, 89(1-2):37–65, 2012.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pp. 2234–2242, 2016.

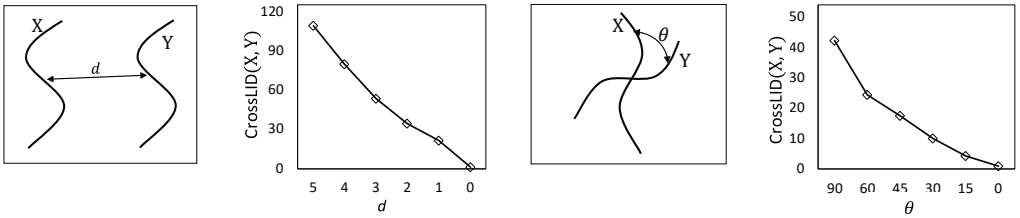
Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *ECCV*, 2018.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pp. 2818–2826, 2016.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. 2016.

A MORE UNDERSTANDING OF CROSSLID ON MANIFOLD POSITIONING AND ORIENTATION

In Fig. 1 section 4, we have shown the power of CrossLID in capturing submanifold closeness via a toy example where a GAN model attempts to learn a bimodal Gaussian distribution. Here, we provide more insights into the understanding of CrossLID. Fig. 8 illustrates how our proposed CrossLID can effectively characterize the closeness of two manifolds (X and Y) with identical geometric structures but different positioning or orientation in space. As the two manifolds move closer and closer to each other either in position (deceasing distance d) or orientation (decreasing angle θ), the CrossLID scores (CrossLID$(X;Y)$) tend to decrease and are close to one (CrossLID$(X;X)$) when the two manifolds are completely overlapping with each other. This substantiates how effectively CrossLID can measure any difference in spatial position and orientation of two manifolds even when they have same structures.



(a) (b) (c) (d)

Figure 8: (a) Two manifolds X and Y have identical geometric structures, but different positions in space; (b) CrossLID score decreases as X and Y moving closer to each other (decreasing d); (c) The same manifolds (as in (a)) but having different orientations (rotated with respect to each other); (d) CrossLID score decreases as the relative orientation angle θ decreases.

B SAMPLE QUALITY EVALUATION OF GEOMETRY SCORE

We train DCGANs on MNIST, CIFAR-10 and SVHN datasets, and compute the Geometry score using 2000 generated samples after each epoch of training for the first 50 epochs. The DCGAN architectures used here are the same as used in other experiments in section 6.1, and are described in Appendix H.1. As demonstrated in Fig. 9, Geometry score exhibits high variations and does not have a clear correlation with the sample quality, which is consistent with the claims in Khrulkov & Oseledets (2018). Visual verification of the improving sample quality over epochs can be found in Fig. 11 and 12. Although a larger sample size (e.g. 10000) was used in the paper, we found it is computationally expensive to compute Geometry scores with such a sample size over epochs.

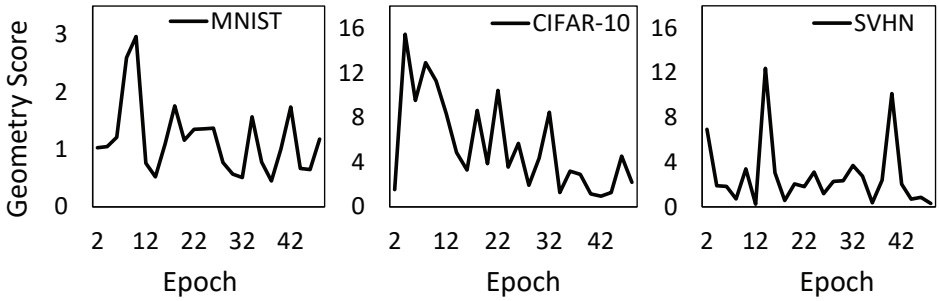


Figure 9: Evaluation of Geometry score on sample quality for DCGAN models trained on MNIST (left), CIFAR-10 (middle) and SVHN (right). After each epoch of training (for the first 50 peochs), we generate 2000 samples using the generator network and compute the Geometry score.

C BENEFITS OF CROSSLID ESTIMATION IN DEEP FEATURE SPACE

In section 4.2 we have proposed to use deep feature space for the estimation of CrossLID, here we further show that such a choice contributes to some advantages (as shown in section 6.1) of CrossLID over Inception score and FID. We test the estimation of CrossLID on purely real samples (eg. CrossLID(X_I, X_I)) on MNIST dataset under two settings: 1) directly at the pixel space, or 2) at the deep feature space defined by an external CNN classifier. Fig. 10 demonstrates the robustness of CrossLID under the two settings in three scenarios: 1) small scale input noise, 2) translations and 3) rotations. The CrossLID scores estimated in the feature space stays invariant consistently across all test scenarios, while scores estimated in the pixel space exhibit some variations (value increases). This confirms the denoising and representation learning capabilities of convolutional networks, and the advantage of CrossLID estimation in deep feature space.

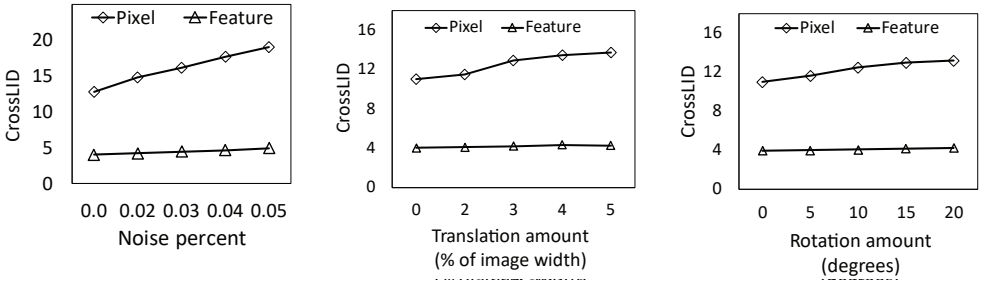


Figure 10: Comparison of CrossLID scores estimated in the pixel space versus that in the deep feature space on MNIST dataset. Robustness against small-scale salt-and-pepper noise (a form of noise sometimes seen on images, which also known as impulse noise) (left), translations (middle) and rotations (right).

D VISUAL INSPECTIONS ON CROSSLID SCORE AND SAMPLE QUALITY

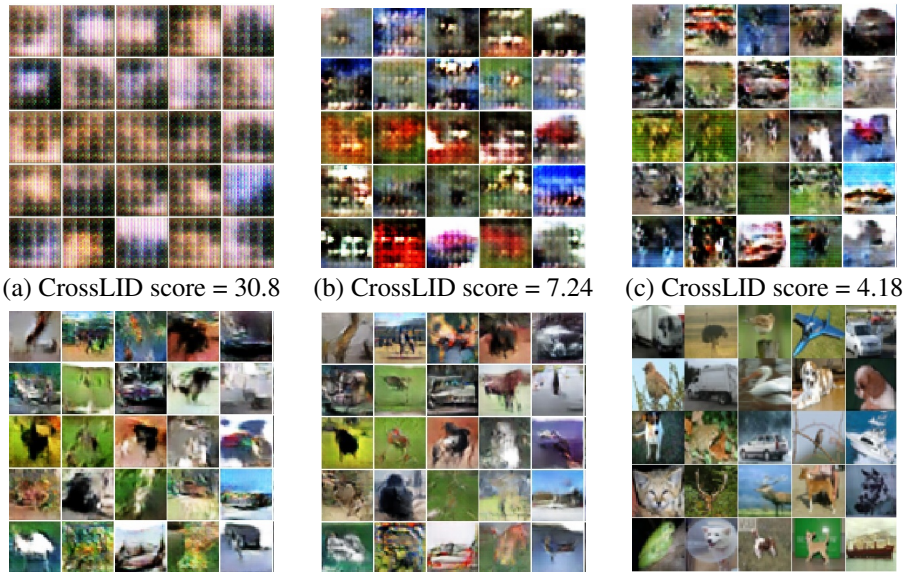
We have, in section 6.1, demonstrated that CrossLID score is closely correlated with the sample quality in a GAN training process, e.g. CrossLID score decreases consistently as the model progressively learns to generate samples with better quality. Here, we visually inspect those images generated at different training stages of a DCGAN model on MNIST and CIFAR-10 datasets. Fig. 11 and 12 show some generated CIFAR-10 and MNIST images respectively. As can be seen, the sample quality increases as training proceeds, and there is a strong correlation between decreasing CrossLID score and increasing sample quality. Note that the last subfigure in both Fig. 11 and 12 shows the real images and the CrossLID score of the real data distribution (CrossLID$(X_I; X_I)$), which is the lowest overall.

E MORE ROBUSTNESS EVALUATION ON INPUT NOISE

In addition to the robustness evaluation against Gaussian input noise in Fig. 6 section 6.1, we provide some analysis on one form of real-world image noise called the salt-and-pepper noise (also known as the impulse noise). As shown in the left subfigure in Fig. 13, although the maximum amount of noise tested is only 1%, FID degrades very sharply with increasing values and Inception score also changes considerably. CrossLID score, on the contrary, remains stable (proportionate variation) under different noise rates.

F CROSSLID ESTIMATION WITH GAN DISCRIMINATOR

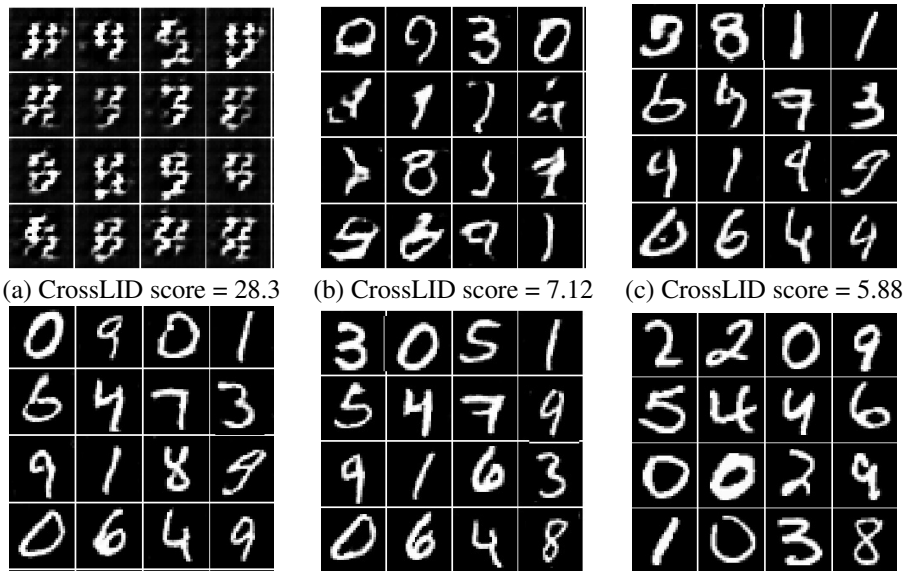
This set of experiments is to confirm that CrossLID score can be reliably estimated using only features from a GAN discriminator without any external CNN classifiers. One can expect that the CrossLID scores estimated by discriminator features are of different scales from those estimated with an external CNN classifier, however, they shall still be able to reflect GAN quality properly



(a) CrossLID score = 30.8 (b) CrossLID score = 7.24 (c) CrossLID score = 4.18

(d) CrossLID score = 3.76 (e) CrossLID score = 3.04 (f) CrossLID score = 1.34

Figure 11: (a-e) 25 randomly selected DCGAN generated CIFAR-10 images and the CrossLID score of the DCGAN model after epoch 1, 5, 10, 20, and 49; (f) Real CIFAR-10 images and the CrossLID$(X_I; X_I)$ score.



(a) CrossLID score = 28.3 (b) CrossLID score = 7.12 (c) CrossLID score = 5.88

(d) CrossLID score = 5.36 (e) CrossLID score = 5.06 (f) CrossLID score = 3.90

Figure 12: (a-e) 16 randomly selected DCGAN generated MNIST images and the CrossLID score of the DCGAN model after epoch 1, 4, 10, 20, and 50; (f) Real MNIST images and the CrossLID$(X_I; X_I)$ score.

as long as they respond similarly (with the same increasing/decreasing trends) to different levels of sample qualities, like the scores estimated using external CNNs. As shown in the right subfigure in Fig. 13, discriminator estimated CrossLID scores indeed follow the same trends as those scores estimated by external classifiers (see left subfigure in Fig. 2, section 6.1). This partially confirms that CrossLID can be reliably estimated directly with the discriminator, more detailed analysis can be an interesting future work.

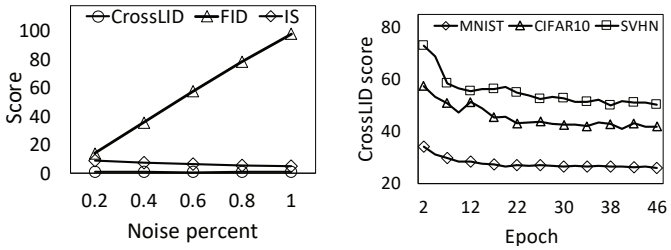
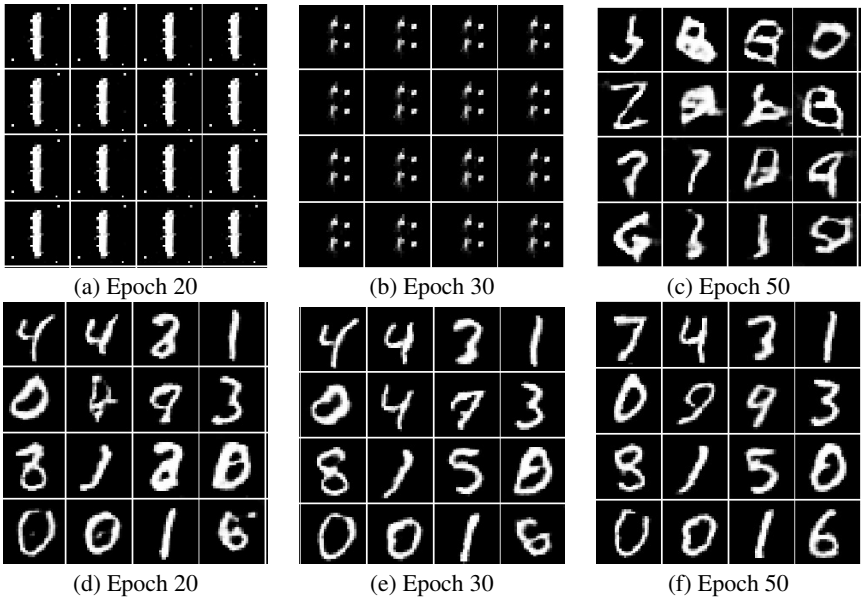


Figure 13: *Left*: Salt-and-pepper noise test of, CrossLID, Inception score (IS) and FID on CIFAR-10 dataset. *Right*: CrossLID scores estimated using discriminator features, over different epochs of GAN training on MNIST, SVHN and CIFAR-10 datasets. The GAN model adopted is the DCGAN. Features of the third convolution layer of the discriminator network were used to compute the average CrossLID score of 20000 generated images.

G STABILITY OF THE PROPOSED GAN TRAINING WITH OVERSAMPLING

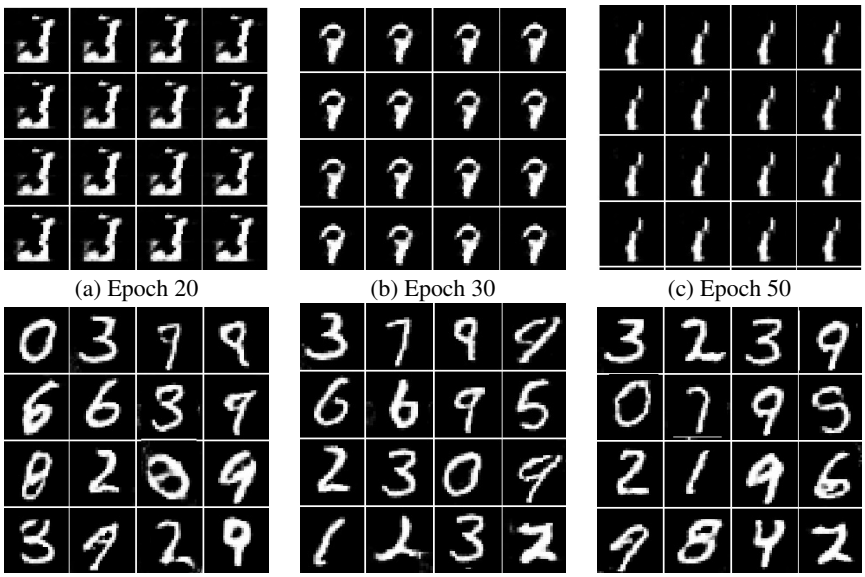
Here, we provide more details about training a DCGAN without batch normalization layers 1) in the generator or 2) in both the generator and the discriminator, following the similar methodology used in (Arjovsky et al., 2017) to verify model stability. We show on MNIST dataset, the generated images at different epochs, how our proposed oversampling strategy can avoid mode collapse and help learning. As visualized in Fig. 14, when batch normalization layers were removed from both the generator and the discriminator, standard training suffered from mode collapse at the beginning and the generated images at the end of training are of low quality. More severe mode collapse was observed with standard training when the batch normalization layers were removed from the discriminator only: the model failed to generate any realistic images (see Fig. 15). On the Contrary, when trained with our proposed oversampling strategy, mode collapse was not observed in any of the two scenarios, and the generated images are of higher quality consistently throughout training.



(a) Epoch 20 (b) Epoch 30 (c) Epoch 50

(d) Epoch 20 (e) Epoch 30 (f) Epoch 50

Figure 14: MNIST images generated at epoch 20, 30 and 50 (50 epochs in total) by DCGAN models without batch normalization layers in both the generator and discriminator. *Top row*: Images generated by a DCGAN model trained using the standard training. *Bottom row*: Images generated by a DCGAN model trained with our proposed oversampling strategy.



(a) Epoch 20 (b) Epoch 30 (c) Epoch 50

(d) Epoch 20 (e) Epoch 30 (f) Epoch 50

Figure 15: MNIST images generated at epoch 20, 30 and 50 (50 epochs in total) by DCGAN models without batch normalization layers in the discriminator (the generator network still has batch normalization). *Top row*: Images generated by a DCGAN model trained using the standard training. *Bottom row*: Images generated by a DCGAN model trained with our proposed oversampling strategy.

H NETWORK ARCHITECTURES AND EXPERIMENTAL SETTINGS

H.1 GAN ARCHITECTURE, TRAINING DETAILS AND EXEMPLARY OUTPUTS

On each dataset, we used the same architecture for both DCGAN and WGAN, following the architectural guidelines in Radford et al. (2016). The generator and discriminator networks used for MNIST and CIFAR-10/SVHN are described in Table 3 and Table 4 respectively. Note that the Tanh output activation was removed from WGAN generator so as to produce a linear output for the training of WGANs with the Wasserstein loss.

The DCGANs were trained for 60 and 100 epochs on MNIST and CIFAR-10/SVHN respectively, using Adam optimizer (Kingma & Ba, 2015) with learning rate 0.0001 and decay 0.00001. The WGANs models were trained for 200 epochs on all datasets using RMSProp optimizer with learning rate 0.00005. For our proposed training strategy, we used $N_1 = 20K, N_2 = 2K$ for class-wise CrossLID estimation, and sample size $m = 30K$ for DCGANs and $m = 20K$ for WGANs. For T, we simply used the number of generator iterations in one epoch, i.e., we applied oversampling after every epoch. Fig. 16 shows some randomly selected images generated by 1) DCGANs trained using standard training versus 2) DCGANs trained with our proposed oversampling strategy.

H.2 EXTERNAL CNN MODELS USED FOR FEATURE EXTRACTION

Table 5 describes the architectures of external CNN models used for feature extraction on MNIST, CIFAR-10 and SVHN datasets, and the selected feature layers are highlighted in **bold**. These CNN classifiers were trained separately on the original training sets of the three datasets. For MNIST and SVHN networks, the outputs of the first fully connected (FC) layer was used as features, while for CIFAR-10, the output of the last max pooling layer was used. For the estimation of our proposed CrossLID, the networks were applied to extract the features for both real and fake images, and the extracted features were then used to compute the CrossLID scores.

Table 3: The generator and discriminator network used for MNIST dataset. Conv(x, y, z) represents a convolution layer with x filters of kernel size $y \times y$ and stride z. ConvTr(x, y, z) represents a transposed convolution layer with x filters of kernel size $y \times y$ and stride z. FC(x) represents a fully connected layer with x output nodes. BN represents a batch normalization layer, R represents the reshape operation and LReLU is the LeakyRelu.

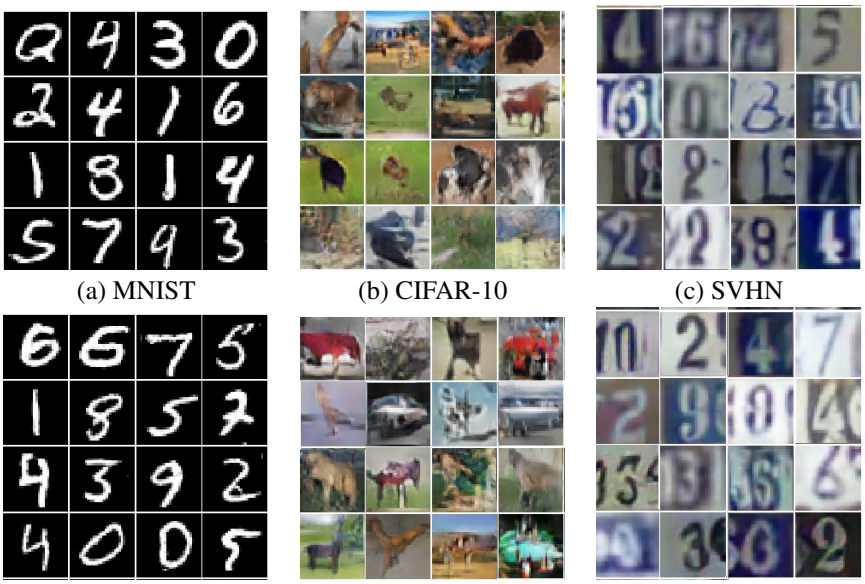
Generator	Discriminator
Input: Z(100)	Input: (28,28,1)
R(1,1,100)	Conv(32,3,2), BN, LReLU
ConvTr(128,3,1), BN, ReLU	Conv(64,3,2), BN, LReLU
ConvTr(64,3,2), BN, ReLU	Conv(128,3,2), BN, LReLU
ConvTr(32,3,2), BN, ReLU	Conv(1,3,1), Sigmoid
ConvTr(1,3,2), Tanh	Output: 1
Output: (28, 28, 1)	

Table 4: The generator and discriminator network used for CIFAR-10/SVHN dataset. Conv(x, y, z) represents a convolution layer with x filters of kernel size $y \times y$ and stride z. ConvTr(x, y, z) represents a transposed convolution layer with x filters of kernel size $y \times y$ and stride z. FC(x) represents a fully connected layer with x output nodes. BN represents a batch normalization layer, R represents the reshape operation and LReLU is the LeakyRelu

Generator	Discriminator
Input: Z(100)	Input: (32,32,3)
R(1,1,100)	Conv(64,4,2), BN, LReLU
ConvTr(256,4,1), BN, ReLU	Conv(128,4,2), BN, LReLU
ConvTr(128,4,2), BN, ReLU	Conv(256,4,2), BN, LReLU
ConvTr(64,4,2), BN, ReLU	Conv(1,4,1), Sigmoid
ConvTr(3,4,3), Tanh	Output: 1
Output: (32, 32, 3)	

Table 5: Network architecture of external CNN models used for feature extraction. Conv(x, y, z) represents a convolution layer with x filters of kernel size $y \times y$ and stride=z. MaxPool(x, y) represents a max-pooling layer with pool size $x \times y$. FC(x) represents a fully connected layer with x output nodes. The selected feature layers are highlighted in **bold**.

Dataset	Architecture of external CNN models
MNIST	Conv(32,3,1), Conv(64,3,1), MaxPool(2,2), **FC(128)**, FC(10)
SVHN	Conv(32,3,1), Conv(32,3,1), MaxPool(2,2), Conv(64,3,1), Conv(64,3,1), MaxPool(2,2), Conv(128,3,1), Conv(128,3,1), MaxPool(2,2), **FC(512)**, FC(10)
CIFAR-10	Conv(64,3,1), Conv(64,3,1), MaxPool(2,2), Conv(128,3,1), Conv(128,3,1), MaxPool(2,2) Conv(256,3,1), Conv(256,3,1), Conv(256,3,1), MaxPool(2,2), Conv(512,3,1), Conv(512,3,1), Conv(512,3,1), MaxPool(2,2), Conv(512,3,1), Conv(512,3,1), Conv(512,3,1), **MaxPool(2,2)**, FC(512), FC(10)



(a) MNIST (b) CIFAR-10 (c) SVHN

(d) MNIST (e) CIFAR-10 (f) SVHN

Figure 16: *Top row:* Images generated by standard DCGAN training (without our proposed oversampling) on the three datasets (a-c). *Bottom row:* Images generated by DCGANs trained with our proposed oversampling strategy on the three datasets (d-f).