Leverage Score Sampling for Complete Mode Coverage in Generative Adversarial Networks

Joachim Schreurs[1][0000−0001−8670−2553], Hannes De Meulemeester[1][0000−0002−5938−2387], Michaël Fanuel[2⋆][0000−0002−7438−0005], Bart De Moor[1][0000−0002−1154−5028], and Johan A.K. Suykens[1][0000−0002−8846−6352]

[1] ESAT-STADIUS, KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{joachim.schreurs,hannes.demeulemeester, bart.demoor, johan.suykens}@kuleuven.be
[2] Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
michael.fanuel@univ-lille.fr

Abstract. Commonly, machine learning models minimize an empirical expectation. As a result, the trained models typically perform well for the majority of the data but the performance may deteriorate in less dense regions of the dataset. This issue also arises in generative modeling. A generative model may overlook underrepresented modes that are less frequent in the empirical data distribution. This problem is known as complete mode coverage. We propose a sampling procedure based on ridge leverage scores which significantly improves mode coverage when compared to standard methods and can easily be combined with any GAN. Ridge leverage scores are computed by using an explicit feature map, associated with the next-to-last layer of a GAN discriminator or of a pre-trained network, or by using an implicit feature map corresponding to a Gaussian kernel. Multiple evaluations against recent approaches of complete mode coverage show a clear improvement when using the proposed sampling strategy.

Keywords: GANs · Leverage score sampling · Complete mode coverage.

1 Introduction

Complete mode coverage is a problem of generative models which has been clearly defined and studied in [26]. In layman's terms, a mode is defined as a local maximum of the data probability density. A closely related problem is mode collapse in GANs [7], which happens when a generative model is only capable of generating samples from a subset of all the modes. Multiple GAN variants have been proposed as a solution to this problem, however proposed solutions often assume that every mode has an (almost) equal probability of being sampled, which is often not the case in realistic datasets. Regularly, in critical applications, datasets contain a mixture of different subpopulations where the frequency of each subpopulation can be vastly different. The role of less abundant

⋆ Most of this work was done when MF was at KU Leuven.

subpopulations in machine learning data has been discussed recently in [6]. Also, it is often common to presume that an algorithm does not know the abundance of subpopulations. It is however important that a machine learning model performs well on all subpopulations. A standard example is medical data where some rare diseases are less abundant than common diseases. To illustrate the approach presented in this paper, a motivating example containing one majority mode and two minority modes is given in Figure 1. When sampling a mini-batch from the Probability Density Function (PDF) p, the side modes can be missed. We observe empirically that this is resolved by sampling from the ridge leverage score (RLS) distribution (see Section 2), which has been extensively used in randomized linear algebra and kernel methods. Figure 1 shows that the samples from the minority modes have larger RLSs. Thus, when sampling from the RLS distribution, there is a higher probability of including the minority modes.

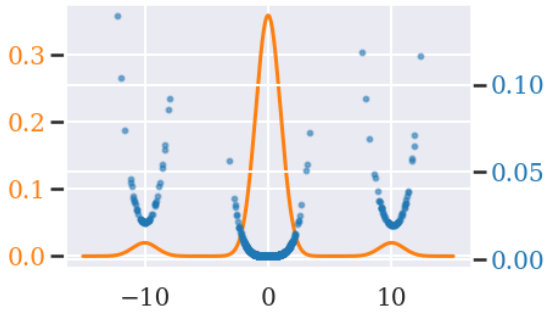

Fig. 1. Probability Density Function (orange) and RLS of a sample of this PDF (blue). We take the motivating example from [26], which consists of a 1D target PDF p with 1 majority mode and 2 minority modes: $p = 0.9 \cdot \mathcal{N}(0,1) + 0.05 \cdot \mathcal{N}(10,1) + 0.05 \cdot \mathcal{N}(-10,1)$ given in orange. The RLS distribution is calculated using a Gaussian kernel with $\sigma = 3$ and $\gamma = 10^{-3}$. When sampling a mini-batch from the PDF p, the side modes can be missed. This is resolved by sampling from the RLS distribution.

This paper is motivated by two situations where minority modes can occur: 1) the observed empirical distribution is different from the true distribution (biased data), and the data needs to be rebalanced. 2) The observed empirical distribution approximates the true distribution sufficiently well, but minority modes consist out of infrequent but very important points, e.g. rare diseases in a medical dataset.

Contribution. When training classical GANs, an empirical expectation of a loss $\mathbb{E}_{x \sim p_d}[\mathcal{L}(x)]$ is optimized in the context of a min-max problem. In this work, we propose a sampling procedure that promotes sampling out of minority modes by using ridge leverage scores. The common algorithmic procedure simulates the empirical distribution over the dataset $\mathcal{D} = \{x_1, \ldots, x_n\}$ by uniformly sampling over this set. We intentionally *bias* or *distort* this process by sampling x_i

with probability $p(x_i) \propto \ell_i$, where ℓ_i is the i-th ridge leverage score, defined in Section 2. Empirical evidence shows that our procedure rebalances the training distribution, as a result, the GAN model generates samples more uniformly over all modes. RLS sampling can easily be applied to any GAN. In particular, using our procedure in combination with a state-of-the-art method for complete mode coverage [26] shows a clear improvement. Finally, RLS sampling is combined with BuresGAN [4] and a state-of-the-art StyleGAN2 with differentiable data augmentations [25], which in both cases improves mode coverage[1].

Related work. Several works discuss alternative sampling strategies in machine learning. In the context of risk-averse learning, the authors of [2] discuss an adaptive sampling algorithm that performs a stochastic optimization of the Conditional Value-at-Risk (CVaR) of a loss distribution. This strategy promotes models which do not only perform well on average but also on rare data points. In the context of generative models, AdaGAN [20] is a boosting approach to solve the missing mode problem, where at every step a new component is added into a mixture model by running the GAN training algorithm on a re-weighted sample. A supervised weighting strategy for GANs is proposed in [5]. In this paper, we compare against two state-of-the-art GANs that combat mode collapse, PacGAN [10] and BuresGAN [4]. PacGAN uses a procedure called packing. This modifies the discriminator to make decisions based on multiple samples from the same class, either real or artificially generated. In BuresGAN, an additional diversity metric in the form of the Bures distance between real and fake covariance matrices is added to the generator loss. Note that these methods tackle the traditional mode collapse problem, i.e., the data does not include minority modes. In [23, 24], it was shown that the convergence speed of stochastic gradient descent can be improved by actively selecting mini-batches using DPPs. In [19], coreset-selection is used to create mini-batches with a 'coverage' similar to that of the large batch – in particular, the small batch tries to 'cover' all the same modes as are covered in the large batch.

Before proceeding further, we discuss two main competitors more in-depth. The authors of [5] propose a solution to reduce selection bias in training data named Importance Weighted Generative Networks. A rescaling of the empirical data distribution is performed during training by employing a weighted Maximum Mean Discrepancy (MMD) loss such that the regions where the observed and the target distributions differ are penalized more. Each sample $i \in \{1, \ldots, n\}$ is scaled by $1/M(x_i)$, where M is the known or estimated Radon-Nykodym derivative between the target and observed distribution. A version of the vanilla GAN with importance weighting is introduced (IwGAN), as well as the weighting combined with MMDGAN (IwMmdGAN). Another approach to complete mode coverage by [26] and dubbed MwuGAN in this paper, iteratively trains a mixture of generators. At each iteration, the sampling probability is pointwise normalized so that the probability to sample a missing mode is increased. Hence, this generates a sequence of generative models which constitutes the mixture. More

[1] Code and supplementary at https://github.com/joachimschreurs/RLS_GAN

precisely, a weight $w_i > 0$ is given for each $i \in \{1, \dots, n\}$ and initialized such that $w_i = p(x_i)$ for some distribution[2] p. Next, a generative model is trained and the probability density $p_g(x_i)$ of each $i \in \{1, \dots, n\}$ is computed. If $p_g(x_i) < \delta p(x_i)$ for some threshold value $\delta \in (0, 1)$, the weight is updated as follows: $w_i \leftarrow 2w_i$, otherwise the weight is not updated. The probability is then recalculated as follows: $p(x_i) = w_i / \sum_j w_j$ for each $i \in \{1, \dots, n\}$. Another generative model is then trained by using $p(x_i)$ and the procedure is repeated.

Classical approach. A GAN consists of a discriminator $D : \mathbb{R}^d \to \mathbb{R}$ and a generator $G : \mathbb{R}^\ell \to \mathbb{R}^d$ which are typically defined by neural networks, and parametrized by real vectors. The value $D(x)$ gives the probability that x comes from the empirical distribution, while the generator G maps a point z in the latent space $\mathbb{R}^\ell$ to a point in input space $\mathbb{R}^d$. A typical training scheme for a GAN consists in solving, in an alternating way, the following problems:

$$\begin{aligned} &\max_D \mathbb{E}_{x \sim p_d}[\log(D(x))] + \mathbb{E}_{\tilde{x} \sim p_g}[\log(1 - D(\tilde{x}))], \\ &\min_G -\mathbb{E}_{\tilde{x} \sim p_g}[\log(D(\tilde{x}))], \end{aligned} \tag{1}$$

which include the vanilla GAN objective associated with the cross-entropy loss. In (1), the first expectation is over the empirical data distribution p_d and the second is over the generated data distribution p_g, implicitly given by the mapping by G of the latent prior distribution $\mathcal{N}(0, \mathbb{I}_\ell)$. The data distribution p_d is estimated using the empirical distribution over the training data $\hat{p}_d(x) = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \delta(x - x_i)$ as follows: $\mathbb{E}_{p_d(x)}[\mathcal{L}(x)] \approx \mathbb{E}_{\hat{p}_d(x)}[\mathcal{L}(x)] = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \mathcal{L}(x_i)$, where $\mathcal{L}$ is a general loss function. As noted by [21], positive weights w_i for $1 \leq i \leq n$ can be used to construct a weighted empirical distribution $\hat{p}_d^w(x) = \sum_{x_i \in \mathcal{D}} w_i \delta(x - x_i)$, then one can apply a weighting strategy to use samples distributed according to $\hat{p}(x)$ to estimate quantities with respect to $\hat{p}_d^w(x)$ as follows:

$$\mathbb{E}_{\hat{p}_d^w(x)}[\mathcal{L}(x)] = \mathbb{E}_{\hat{p}_d(x)}\left[\frac{\hat{p}_d^w(x)}{\hat{p}_d(x)} \mathcal{L}(x)\right] = \sum_{x_i \in \mathcal{D}} w_i \mathcal{L}(x_i). \tag{2}$$

A stochastic procedure is applied for minimizing the above expectation over $\hat{p}_d^w$. In this paper, mini-batches are sampled according to the distribution $\hat{p}_d^w$ with w_i given by the normalized RLSs (3) for $1 \leq i \leq n$.

2 Sampling with Ridge Leverage Scores

We propose to use a sampling procedure based on ridge leverage scores (RLSs) [1, 11]. RLSs correspond to the correlation between the singular vectors of a matrix and the canonical basis elements. The higher the score, the more *unique* the point. A sample from a minority mode would thus get a higher RLS. These

[2] In [26], this initial distribution is uniform. We discuss in Section 3.1 a choice of weights based on RLSs and initialize MwuGAN with the normalized RLSs in (3).

RLSs are used to bias the sampling, which in turn results in a more uniform sampling over all the modes, regardless of the original weight of the mode in the data distribution. Given a feature map $\varphi(\cdot)$, the corresponding kernel function is $K(x, y) = \varphi(x)^\top \varphi(y)$. Let the regularization parameter be $\gamma > 0$. Then, the γ-RLSs are defined for all $1 \leq i \leq n$ as:

$$\ell_i(\gamma) = \left(K(K + n\gamma\mathbb{I})^{-1}\right)_{ii} = \varphi(x_i)^\top (C + n\gamma\mathbb{I})^{-1}\varphi(x_i), \tag{3}$$

where $C = \sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top$ and $K_{ij} = \varphi(x_i)^\top \varphi(x_j)$ for $1 \leq i, j \leq n$. They have both a primal and a dual expression that can be leveraged when the size of the feature map or batch-size respectively are too large. When both the batch-size and feature map dimensions are large, one can rely on fast and reliable approximation algorithms with guarantees such as RRLS [15] and BLESS [17]. The role of $\gamma > 0$ is to filter the small eigenvalues of K in the spirit of Tikhonov regularization. RLSs induce the probability distribution: $p_i = \ell_i / \sum_{j=1}^n \ell_j$, for $1 \leq i \leq n$, which is classically used in randomized methods [1]. Figure 2 illustrates the interpretation of RLSs on two artificial datasets used in this paper. The datasets consist of a mixture of Gaussians. In the Ring example, the first 4 modes are minority modes (starting on top and going further clockwise). In the Grid example, the first 10 modes are minority modes (starting left). Similar to the first illustration (see Figure 1), large RLSs are associated with minority modes. More information on the artificial datasets is given in Section 3.1.

Fig. 2. RLS distribution using a Gaussian feature map with $\sigma = 0.15$ and regularization $\gamma = 10^{-3}$ on the unbalanced Ring (left) and Grid (right) data. The darker the shade, the higher the RLS. Dark modes correspond to minority modes.

RLS sampling has a rich history in kernel methods and randomized linear algebra but has not been used in the context of GANs. One of the key contributions of this paper is to illustrate the use of RLSs in this setting. To do so, we propose the use of different feature maps so that RLS sampling can be used both for low dimensional and high dimensional data such as images. In what follows, the feature map construction is first discussed. Next, two approximation schemes are introduced.

Choice of the feature map. Three choices of feature maps are considered in this paper to compute leverage scores:

- Fixed implicit feature map. In low dimensional examples, the feature map can be chosen implicitly such that it corresponds to the Gaussian kernel: $\varphi(x)^\top \varphi(y) = \exp\left(-\|x-y\|^2/\sigma^2\right)$, the bandwidth σ is a hyperparameter.
- Fixed explicit feature map. For image-based data, more advanced similarity metrics are necessary. Therefore, the next-to-last layer of a pre-trained classifier, e.g. the Inception network, is used to extract meaningful features. Note that the classifier does not need to be trained on the exact training dataset, but simply needs to extract useful features.
- Discriminator-based explicit feature map. The feature map can be obtained from the next-to-last layer of the discriminator. Let $D(x) = \sigma(w^\top \varphi_D(x))$, where $w \in \mathbb{R}^m$ contains the last dense layer's weights and σ is the sigmoid function. This feature map is useful in situations where no prior knowledge is available about the dataset.

For a fixed feature map, the RLSs only need to be calculated once before training. The discriminator-based explicit feature map changes throughout the training. Therefore the RLSs are recalculated at every step. Nonetheless, due to approximation schemes which are discussed hereafter, the computational cost stays low. The full algorithm is given in Supplementary Material.

2.1 Approximation schemes

Current day models are high dimensional, e.g., a DCGAN yields a feature space $\mathbb{R}^m$ of high dimension $m = 10^3$. Moreover, the size of datasets is commonly thousands up to millions of images. Therefore, two approximation schemes are proposed to speed up the computation of RLSs when using explicit feature maps:

- For the discriminator-based explicit feature map, we propose a two-stage sampling procedure in combination with a Gaussian sketch to reduce the dimension of high dimensional feature maps.
- For the fixed explicit feature map, the well-known UMAP is used to reduce the dimensionality [13].

Two-stage sampling procedure. For the explicit discriminator-based feature map, φ_D has to be re-calculated at each training step. To speed up the sampling procedure, we propose a sampling procedure in two stages. First, a subset of the data is uniformly sampled, e.g. equal to 20 times the desired batch size. Afterward, the RLSs are calculated only for the uniformly sampled subset, which are then used to sample the final batch used for training. This two-stage sampling procedure is similar to the core-set selection used in smallGAN [19]. A first difference is that core-sets are selected by combining a Gaussian sketch and a greedy selection in [19], while we use a randomized approach. Second, in this reference, cores-sets are used to reduce the batch size to improve scalability. In contrast, RLS sampling is used here to bias the empirical distribution.

Sketching the discriminator feature map. Gaussian sketching is a commonly used method to reduce data dimension and was also used in [19] and [22] to reduce

the dimension of large neural nets. Let S be a sketching matrix of size $m \times k$ such that $S = A/\sqrt{k}$ with A a matrix with i.i.d. zero-mean standard normal entries. Consider the following random projection: let a batch be $\{i_1, \ldots, i_b\} \subset \{1, \ldots, n\}$. A random projection of this batch in feature space is then defined as follows:

$$\varphi(x_{i_\ell}) = S^\top \varphi_D(x_{i_\ell}) \in \mathbb{R}^k, \tag{4}$$

for all $\ell \in \{1, \ldots, b\}$. This random projection preserves approximately (squared) pairwise distances in the dataset and is motivated by an isometric embedding result in the spirit of Johnson-Lindenstrauss lemma. Let $0 < \epsilon < 1$ and any integer $b > 0$. Let k be an integer such that $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log b$. Then, for any set $\{\mathsf{x}_1, \ldots, \mathsf{x}_b\}$ in $\mathbb{R}^m$ there is a map $f : \mathbb{R}^m \to \mathbb{R}^k$ such that for any $\ell, \ell' \in \{1, \ldots, b\}$ we have

$$(1 - \epsilon)\|\mathsf{x}_\ell - \mathsf{x}_{\ell'}\|_2^2 \leq \|f(\mathsf{x}_\ell) - f(\mathsf{x}_{\ell'})\|_2^2 \leq (1 + \epsilon)\|\mathsf{x}_\ell - \mathsf{x}_{\ell'}\|_2^2.$$

The idea of this work is to consider the set of points given by the batch in the discriminator feature space $\mathsf{x}_\ell = \varphi_D(x_{i_\ell})$ for $1 \leq \ell \leq b$. It is proved in [3] that f exists and can be obtained with high probability with a random projection of the form (4). For a more detailed discussion, we refer to [16].

Dimensionality reduction of the fixed explicit feature map by UMAP. The Gaussian sketch is a simple and fast method to reduce the dimension of the feature map. This makes it a perfect candidate to reduce the dimension of the proposed discriminator feature map, which has to be recalculated at every iteration. Unfortunately, this speed comes at a price, namely, the gaussian sketch is deemed too simple to reduce the dimension of highly complex models like the Inception network. Therefore, UMAP is proposed [13]. This non-linear dimensionality reduction technique can extract more meaningful features. UMAP is considerably slower than the Gaussian sketch, therefore the use of UMAP is only advised for a fixed feature map like a pre-trained classifier or the Inception network, where the RLSs are only calculated once before training.

3 Numerical experiments

The training procedure is evaluated on several synthetic and real datasets, where we artificially introduce minority modes. The GANs are evaluated by analyzing the distribution of the generated samples. Ideally, the models should generate samples from every mode as uniformly as possible. A re-balancing effect should be visible. The proposed methods are compared with vanilla GAN, PacGAN, MwuGAN, BuresGAN, IwGAN, and IwMmdGAN. In particular, MwuGAN outperforms AdAGAN [20] on complete mode coverage problems [26]. BuresGAN [4] promotes a matching between the covariance matrices of real and generated data in a feature space defined thanks to the discriminator. Recall that the discriminator is $D(x) = \sigma(w^\top \varphi_D(x))$, where w is a weight vector and the sigmoid function is denoted by σ. The normalization $\bar{\varphi}_D(x) = \varphi_D(x)/\|\varphi_D(x)\|_2$ is used, after the centering of $\varphi_D(x)$. Then,

the covariance matrix is defined as follows: $C(p) = \mathbb{E}_{x\sim p}[\bar{\varphi}_D(x)\bar{\varphi}_D(x)^\top]$. The real data and generated data covariance matrices are denoted by $C_d = C(p_d)$ and $C_g = C(p_g)$, respectively. In BuresGAN, the Bures distance is added to the generator loss: $\min_G -\mathbb{E}_{\tilde{x}\sim p_g}[\log(D(\tilde{x}))] + \lambda\mathcal{B}(C_d, C_g)^2$, with the Bures distance $\mathcal{B}\left(C_d, C_g\right)^2 = \mathrm{Tr}(C_d + C_g - 2\left(C_d C_g\right)^{\frac{1}{2}})$, depends implicitly on $\varphi_D(x)$ (see e.g., [12]). The loss of the discriminator remains the same.

Overview of proposed methods. The RLS sampling procedure can easily be integrated into any GAN architecture or model. In this spirit, we used RLS sampling with a classical vanilla GAN (**RLS GAN**) and BuresGAN (**RLS BuresGAN**). The classical BuresGAN has been shown to outperform competitors in mode collapse problems [4]. We noticed empirically that RLS BuresGAN outperformed RLS GAN on the synthetic data (the comparison is shown in Supplementary Material). Therefore we only continue with RLS BuresGAN in the rest of the experiments. Likewise, RLS sampling is combined with MwuGAN, which is considered state-of-the-art in complete mode coverage. The method, called **RLS MwuGAN**, uses RLSs as initial starting weights to sample as opposed to uniform weights. Besides the initialization, the method remains unchanged. The number of generators in the mixture is always displayed in brackets. Unless specified otherwise, the models are trained for 30k iterations with a batch size of 64, by using the Adam [9] optimizer. Unless specified otherwise, we report the means and standard deviations for 10 runs. The largest mean is depicted in black, a * represents significance using a one-tailed Welch's t-test between the best performing proposed model and best performing competitor at a 0.05 confidence level. Further information about the used architectures, hyperparameters, and timings are given in Supplementary Material.

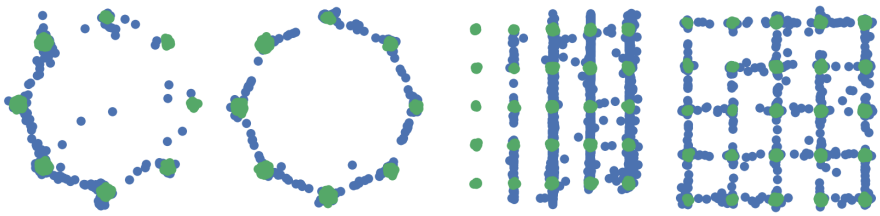

Fig. 3. Visualization of the generation quality on RING and GRID. Each column shows 2.5k samples from the trained generator in blue and 2.5k samples from the true distribution in green. The vanilla GAN (first and third) does not cover the minority modes. This is not the case for the RLS BuresGAN Discr. (second and fourth).

3.1 Synthetic data

Unbalanced versions of two classical synthetic datasets are generated: an unbalanced ring with 4 minority modes (RING) and an unbalanced grid (GRID) with

10 minority modes (see Figure 2). RING is a mixture of eight two-dimensional isotropic Gaussians in the plane with means $2.5 \times (\cos((2\pi/8)i), \sin((2\pi/8)i))$ and std 0.05 for $i \in \{1, \ldots, 8\}$. The probability of sampling from the first 4 consecutive Gaussians is only 0.05 times the probability of sampling from the last 4 modes. GRID is a mixture of 25 two-dimensional isotropic normals with standard deviation 0.05 and with means on a square grid with spacing 2. The first rectangular blocks of 2×5 adjacent modes are depleted with a factor 0.05.

Table 1. Experiments on the synthetic datasets RING and GRID. Two RLS BuresGAN are considered: RLSs calculated with the Gaussian kernel (Gauss.) and the next-to-last layer of the discriminator (Discr.). RLS MwuGAN is initialized with RLSs using an implicit feature map associated with the Gaussian kernel.

	Ring with 8 modes		Grid with 25 modes	
	Nb modes (↑)	% in 3σ (↑)	Nb modes (↑)	% in 3σ (↑)
GAN	5.0(1.1)	**0.92**(0.02)*	8.3(3.4)	0.29(0.3)
PacGAN2	5.4(1.4)	**0.92**(0.03)	10.3(2.6)	0.13(0.02)
BuresGAN	5.8(1.4)	0.76(0.27)	16.7(0.9)	**0.82**(0.01)*
RLS GAN Gauss.	7.4(0.9)	0.86(0.05)	13.8(7.4)	0.51(0.32)
RLS GAN Discr.	7.6(0.8)	0.90(0.02)	20.4(2.6)	0.81(0.03)
IwGAN	**8**(0)	0.85(0.08)	13.4(5.7)	0.29(0.25)
IwMmdGAN	**8**(0)	0.84(0.02)	1.7(5.1)	0.03(0.05)
MwuGAN (15)	7.9(0.3)	0.86(0.02)	15.2(1.7)	0.47(0.11)
RLS MwuGAN (15) (ours)	**8**(0)	0.84(0.06)	22.3(1.9)	0.60(0.1)
RLS BuresGAN Gauss. (ours)	**8**(0)	0.90(0.02)	24.0(1.5)	0.76(0.11)
RLS BuresGAN Discr. (ours)	**8**(0)	0.90(0.02)	**24.4**(0.92)*	0.78(0.06)

Evaluation. The evaluation is done by sampling 10k points from the generator network. High-quality samples are within 3 standard deviations of the nearest mode. A mode is covered if there are at least 50 generated samples within 3 standard deviations of the center of the mode. The knowledge of the full Radon-Nikodym derivative M is given to IwMmdGAN and IwGAN by dividing the true probability of each sample (a Gaussian mixture with equal weights) by the adapted Gaussian mixture containing several minority modes. The results of the experiments are given in Table 1. For RLS sampling, we use a Gaussian kernel with bandwidth $\sigma = 0.15$, and the discriminator network as feature extractor, both with regularization parameter $\gamma = 10^{-3}$. The models are trained using a fully connected architecture (see Supplementary Material). As the models are rather simple, no dimensionality reduction is needed.

Generated samples from models trained with and without RLS sampling are displayed in Figure 3. One can clearly see that training a GAN with uniform sampling results in missing the first 4 minority modes. This is solved by using RLS

sampling and can be interpreted by comparing the two sampling distributions on Figure 4 (uniform) and Figure 2 (RLS). The RLSs are larger for samples in minority modes, which results in a more uniform mini-batch over all modes. Note that the RLS sampling procedure, given the feature map, is completely unsupervised and has no knowledge of the desired unbiased distribution. The evaluation metrics in Table 1 confirm our suspicions. Only methods designed for complete mode coverage can recover (almost) all modes for the RING dataset. For the unbalanced GRID, only the proposed method has an acceptable performance. Our method even outperforms multiple generator architectures like MwuGAN and RLS MwuGAN, which are considerably more costly to train. Moreover, IwGAN, with full knowledge of M, is not capable of consistently capturing all modes. This was pointed out by the authors in [5]: the method may still experience high variance if it rarely sees data points from a class it wants to boost.

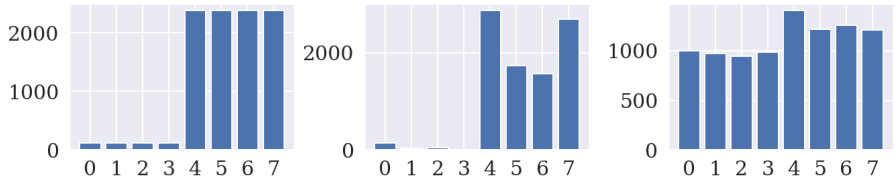

Fig. 4. RING. Number of training samples in each mode for the RING dataset (left) Generated samples in each mode by a vanilla GAN (middle). Generated samples by RLS BuresGAN Discr. (right). A rebalancing effect is visible.

3.2 Unbalanced MNIST

For this experiment, we create two unbalanced datasets out of MNIST. The first modified dataset, named UNBALANCED 012-MNIST, consists of only the digits $0, 1$ and 2. The class 2 is depleted so that the probability of sampling 2 is only 0.05 times the probability of sampling from the digit 0 or 1. The second dataset, named UNBALANCED MNIST, consists of all digits. The classes 0,1,2,3, and 4 are all depleted so that the probability of sampling out of the minority classes is only 0.05 times the probability of sampling from the majority digits. For these experiments, we use a DCGAN architecture. The following metrics are used for performance evaluation: the number of generated digits in each mode, which measures mode coverage, and the KL divergence [14] between the classified labels of the generated samples and a balanced label distribution, which measures sample quality. The mode of each generated image is identified by using a MNIST classifier which is trained up to 98.43% accuracy (see Supplementary Material). The metrics are calculated based on 10k generated images for all the models. For the RLS computation, we use both the discriminator with a Gaussian sketch and the next-to-last layer of the pre-trained classifier with a UMAP dimensionality reduction as a feature map. For the UNBALANCED 012-MNIST, both feature

maps are reduced to $k = 25$ and the regularization parameter is $\gamma = 10^{-4}$. In the UNBALANCED MNIST, we take $k = 10$ and $\gamma = 10^{-4}$. An ablation study over different k and γ is given in Supplementary Material. We also compare the performance of the classical MwuGAN, initialized with uniform weights, with RLS MwuGAN where the weights are initialized by the RLSs calculated using the fixed explicit feature map with the same parameters mentioned above. Both methods contain a mixture of 15 GANs, the experiments are repeated 3 times for MwuGAN variants. In our simulations, IwMmdGAN could not be trained

Fig. 5. Generated images from UNBALANCED 012-MNIST by a vanilla GAN (first), by RLS BuresGAN Discr. (second) and generated images from UNBALANCED MNIST by a vanilla GAN (third), by RLS BuresGAN Class. (fourth). The minority digits are generated more frequently in the proposed methods that include RLS sampling.

successfully with a DCGAN architecture. The Radon-Nikodym derivative M, which is used by IwMmdGAN and IwGAN, is defined as follows: $M_i = 1$ for digits 0 and 1 and $M_i = 0.05$ for digits 2, analogous for the UNBALANCEDMNIST dataset. Only the proposed models trained with RLS sampling are capable of covering all modes consistently. The diversity of images generated by RLS BuresGAN can be visualized in Figure 5 where digits from minority modes appear more frequently. A quantitative analysis of mode coverage and sample quality is reported in Tables 2 and 3. In the UNBALANCED 012-MNIST dataset, there is a clear advantage in using RLS sampling with BuresGAN since mode coverage and the KL divergence are improved compared to the other methods. The second best method is RLS MwuGAN which outperforms RLS with uniform starting weights in KL. For the more difficult UNBALANCED MNIST dataset, using the fixed explicit feature map to calculate the RLSs clearly outperforms other methods.

3.3 Unbalanced CIFAR10

We conclude this section with an experiment on colored images, namely the CIFAR10 dataset. This highly diverse dataset contains 32×32 color images from 10 different classes. We consider two unbalanced variations. The first modified dataset, named UNBALANCED 06-CIFAR10, consists of only the classes 0 and 6 or images of airplanes and frogs respectively. The class 0 is depleted with a factor 0.05. The second dataset, named UNBALANCED 016-CIFAR10, consists of the classes 0,1 and 6. Compared to the previous dataset, we add images from the class

Table 2. Experiments on the UNBALANCED 012-MNIST dataset. Two RLS BuresGAN are considered: RLSs calculated with an explicit feature map obtained from a pre-trained classifier (Class.) and the next-to-last layer of the discriminator (Discr.). RLS MwuGAN is initialized with RLSs using the explicit feature maps obtained from a pre-trained classifier. Minority modes are highlighted in black in the first row.

	Mode 1	Mode 2	**Mode 3**	KL
GAN	4381(172)	5412(179)	129(36)	0.31(0.01)
PacGAN2	4492(237)	5328(242)	123(29)	0.32(0.01)
BuresGAN	4586(287)	5190(292)	142(19)	0.30(0.01)
IwGAN	4368(295)	5414(287)	147(32)	0.32(0.01)
IwMmdGAN	34(12)	0(0)	69(12)	0.56(0.10)
MwuGAN (15)	4886(473)	4865(466)	176(14)	0.31(0.01)
RLS MwuGAN (15) (ours)	3982(218)	4666(164)	870(65)	0.14(0.01)
RLS BuresGAN Class. (ours)	3414(161)	4862(134)	1461(183)	**0.08**(0.01)*
RLS BuresGAN Discr. (ours)	5748(172)	2416(268)	**1566**(293)*	0.16(0.02)

automobile. Now, the class 6 consisting of frogs is depleted with a factor 0.05. We show the improvement of RLS sampling in a StyleGAN2 with differentiable data augmentation (StyleGAN2 + Aug.) [25] [3]. By clever use of various types of differentiable augmentations on both real and fake samples, the GAN can match the top performance on CIFAR10 with only 20% training data and is considered state-of-the-art. The StyleGAN2 models are trained for 156k iterations with a mini-batch size of 32 using 'color, translation, and cutout' augmentations, which is suggested by the authors when only part of the CIFAR10 dataset is used. All the other parameters remained the same, only the sampling strategy is changed to RLS sampling in RLS StyleGAN2 + Aug. For the RLS computation, we use the discriminator feature map with Gaussian sketching and a fixed explicit feature map given by the next-to-last layer of the Inception network where UMAP is used to reduce the dimension. For both the RLSs, the dimension is reduced to $k = 25$ and the regularization parameter is $\gamma = 10^{-4}$. The performance is assessed using 10k generated samples at the end of training by the Inception Score (IS) and the Fréchet inception distance (FID) between the generated fake dataset and the balanced dataset. Mode coverage is evaluated by the number of generated samples in each class. The class of a generated sample is evaluated by a trained CIFAR10 classifier using a resnet56 type architecture [8] which is trained up to 93.77% accuracy[4]. The results of the experiments are given in Table 4, examples of generated images are given in Figure 6. Including RLS sampling clearly improves the performance in unbalanced datasets, this is especially the case for the fixed feature map given by the Inception network. The minority mode is oversampled by approximately a factor 10 or even 100 in the case of

[3] Code taken from https://github.com/mit-han-lab/data-efficient-gans
[4] Classifier is available at https://github.com/gahaalt/ResNets-in-tensorflow2

Table 3. Experiments on the UNBALANCED MNIST dataset. Two RLS BuresGAN variants are considered: RLSs calculated with an explicit feature map obtained from a pre-trained classifier (Class.) and the next-to-last layer of the discriminator (Discr.). RLS MwuGAN is initialized with RLSs using the explicit feature maps obtained from a pre-trained classifier. Only the number of samples in the minority modes are visualized. The number of samples in the remaining modes are given in Supplementary Material.

	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**	KL
GAN	123(22)	137(103)	81(31)	161(97)	161(23)	0.48(0.02)
PacGAN2	109(29)	142(70)	89(23)	147(100)	152(40)	0.48(0.02)
BuresGAN	126(31)	157(97)	108(30)	153(62)	147(26)	0.46(0.02)
IwGAN	117(33)	139(33)	97(31)	212(69)	154(34)	0.46(0.02)
IwMmdGAN	1(1)	0(0)	23(16)	**1140**(367)*	3(3)	1.92(0.1)
MwuGAN (15)	144(29)	113(28)	146(13)	172(18)	167(28)	0.46(0.02)
RLS MwuGAN (15) (ours)	336(47)	283(32)	191(23)	381(38)	276(33)	0.30(0.02)
RLS BuresGAN Class. (ours)	**875**(112)*	**663**(122)*	**360**(198)*	831(59)	**615**(82)*	**0.09**(0.01)*
RLS BuresGAN Discr. (ours)	235(62)	183(141)	264(44)	255(109)	219(54)	0.37(0.02)

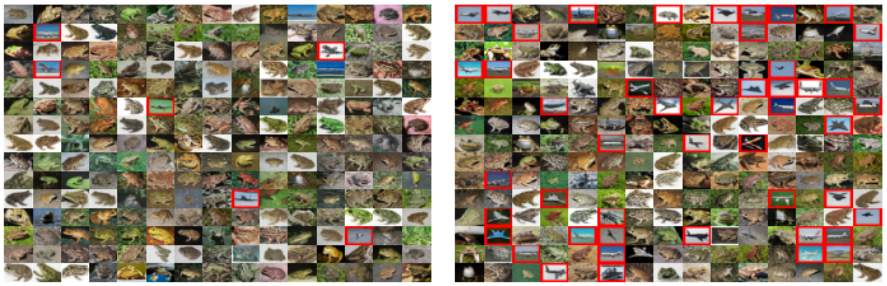

Fig. 6. Generated images from 06-CIFAR10 by a StyleGAN2 + Aug. (left) and by RLS StyleGAN2 + Aug. (right). Including RLS sampling promotes sampling from the minority class. Generated samples classified as planes are marked by a red border.

the unbalanced 06-CIFAR10 and 016-CIFAR10 datasets respectively. Both the IS and FID also improve significantly. Note that the maximum achievable performance for IS and FID is lower when only a subset of classes is included, as pointed out by [18].

4 Conclusion

We introduced the use of RLS sampling for training GANs. This 'diverse' sampling procedure was motivated by a notion of complete mode coverage in the presence of minority modes. RLS sampling is easy to integrate into any GAN model. Three feature maps have been discussed. An implicit feature map performs well for low-dimensional data. A fixed explicit feature map, such as a pre-trained classifier, achieves good results in high-dimensional cases. Lastly, the discriminator can

Table 4. Experiments on the UNBALANCED 06-CIFAR10 and UNBALANCED 016-CIFAR10 dataset. Including RLS sampling in the StyleGAN2 + Aug. clearly improves the performance. Minority modes are highlighted in black in the second row.

	06-CIFAR10				016-CIFAR10				
	Mode 1	Mode 2	IS (↑)	FID (↓)	Mode 1	Mode 2	**Mode 3**	IS (↑)	FID (↓)
StyleGAN2 + Aug.	261	9500	4.8	67.5	4526	5206	18	4.3	48.8
RLS StyleGAN2 + Aug. Disc. (ours)	994	8659	5.7	46.4	4449	5132	139	4.6	44.4
RLS StyleGAN2 + Aug. Class. (ours)	**2438**	7212	**6.2**	**31.3**	4156	4393	**1155**	**5.7**	**27.2**

be used as a feature map when no prior knowledge exists about the data. Two approximation methods for the explicit feature maps are also discussed: dimensionality reduction of explicit feature maps and a two-stage sampling procedure to efficiently speed up online RLS computation. We demonstrated empirically that the use of RLS sampling in GANs successfully combats the missing mode problem.

Acknowledgments. EU: ERC Advanced Grants(787960, 885682). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: projects C14/18/068, C16/15/059, C3/19/053, C24/18/022, C3/20/117, Industrial Research Fund: 13-0260, IOF/16/004; Flemish Government: FWO: projects: GOA4917N, EOS Project no G0F6718N (SeLMA), SBO project S005319N, Infrastructure project I013218N, TBM Project T001919N; PhD Grants (SB/1SA1319N, SB/1S93918, SB/1S1319N), EWI: the Flanders AI Research Program. VLAIO: Baekeland PhD (HBC.20192204) and Innovation mandate (HBC.2019.2209), CoT project 2018.018. Foundation 'Kom op tegen Kanker', CM (Christelijke Mutualiteit). Ford KU Leuven Research Alliance Project KUL0076.

References

1. Alaoui, A., Mahoney, M.W.: Fast randomized kernel ridge regression with statistical guarantees. In: Neural Information Processing Systems. pp. 775–783 (2015)
2. Curi, S., Levy, K., Jegelka, S., Krause, A., et al.: Adaptive sampling for stochastic risk-averse learning. Neural Information Processing Systems (2020)
3. Dasgupta, S., Gupta, A.: An Elementary Proof of a Theorem of Johnson and Lindenstrauss. Random Struct. Algorithms **22**(1), 60–65 (Jan 2003)
4. De Meulemeester, H., Schreurs, J., Fanuel, M., De Moor, B., Suykens, J.A.: The bures metric for generative adversarial networks. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2021)
5. Diesendruck, M., Elenberg, E.R., Sen, R., Cole, G.W., Shakkottai, S., Williamson, S.A.: Importance weighted generative networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 249–265 (2019)
6. Feldman, V.: Does learning require memorization? a short tale about a long tail. In: 52nd Annual ACM SIGACT Symposium on Theory of Computing. p. 954–959. STOC 2020 (2020)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Neural information processing systems. pp. 2672–2680 (2014)

8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) 2015 (2014)
10. Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. In: Neural Information Processing Systems. vol. 31, pp. 1498–1507 (2018)
11. Ma, P., Mahoney, M., Yu, B.: A statistical perspective on algorithmic leveraging. In: 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 91–99 (2014)
12. Massart, E., Absil, P.A.: Quotient Geometry with Simple Geodesics for the Manifold of Fixed-Rank Positive-Semidefinite Matrices. SIAM Journal on Matrix Analysis and Applications **41**(1), 171–198 (2020)
13. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. preprint arXiv:1802.03426 (2018)
14. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled Generative Adversarial Networks. In: International Conference on Learning Representations (ICLR) (2017)
15. Musco, C., Musco, C.: Recursive sampling for the nystrom method. In: Neural Information Processing Systems. pp. 3833–3845 (2017)
16. Oymak, S., Recht, B., Soltanolkotabi, M.: Isometric sketching of any set via the Restricted Isometry Property. Information and Inference: A Journal of the IMA **7**(4), 707–726 (03 2018)
17. Rudi, A., Calandriello, D., Carratino, L., Rosasco, L.: On fast leverage score sampling and optimal learning. In: Neural Information Processing Systems. pp. 5672–5682 (2018)
18. Sajjadi, M.S.M., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. In: 32nd International Conference on Neural Information Processing Systems. p. 5234–5243 (2018)
19. Sinha, S., Zhang, H., Goyal, A., Bengio, Y., Larochelle, H., Odena, A.: Small-GAN: Speeding up GAN training using core-sets. In: 37th International Conference on Machine Learning. vol. 119, pp. 9005–9015 (2020)
20. Tolstikhin, I.O., Gelly, S., Bousquet, O., Simon-Gabriel, C.J., Schölkopf, B.: Adagan: Boosting generative models. In: Neural Information Processing Systems. pp. 5424–5433 (2017)
21. Tripp, A., Daxberger, E., Hernández-Lobato, J.M.: Sample-efficient optimization in the latent space of deep generative models via weighted retraining. Neural Information Processing Systems **33** (2020)
22. Yang, Z., Moczulski, M., Denil, M., De Freitas, N., Smola, A., Song, L., Wang, Z.: Deep fried convnets. In: IEEE International Conference on Computer Vision. pp. 1476–1483 (2015)
23. Zhang, C., Kjellstrom, H., Mandt, S.: Determinantal point processes for mini-batch diversification. Uncertainty in Artificial Intelligence (2017)
24. Zhang, C., Öztireli, C., Mandt, S., Salvi, G.: Active mini-batch sampling using repulsive point processes. In: AAAI Conference on Artificial Intelligence. vol. 33, pp. 5741–5748 (2019)
25. Zhao, S., Liu, Z., Lin, J., Zhu, J.Y., Han, S.: Differentiable augmentation for data-efficient gan training. Neural Information Processing Systems (2020)
26. Zhong, P., Mo, Y., Xiao, C., Chen, P., Zheng, C.: Rethinking generative mode coverage: A pointwise guaranteed approach. In: Neural Information Processing Systems. pp. 2088–2099 (2019)