APSIPA Transactions on Signal and Information Processing, 2023, 12, e33

Original Paper

Lightweight Quality Evaluation of Generated Samples and Generative Models

Ganning Zhao[1*], Vasileios Magoulianitis[1], Suya You[2] and C.-C. Jay Kuo[1]

[1] *University of Southern California, Los Angeles, California, USA*
[2] *DEVCOM Army Research Laboratory, Adelphi, Maryland, USA*

ABSTRACT

Although there are metrics to evaluate the performance of generative models, little research is conducted on the quality evaluation of individual generated samples. A lightweight generated sample quality evaluation (LGSQE) method is proposed in this work. LGSQE trains a binary classifier to differentiate real and synthetic images from a generative model and, then, uses it to assign a soft label between zero and one to a generated sample as its quality index. LGSQE can reject poor generations and serve as a post-processing module for quality control. Furthermore, by aggregating quality indices of a large number of generated samples, LGSQE offers four metrics (i.e., classification accuracy (Acc), the area under the curve (AUC), precision, and recall) to evaluate the performance of a generative model as a byproduct. LGSQE demands a significantly smaller memory size and faster evaluation time while preserving the same rank order predicted by the Fréchet Inception Distance (FID). Extensive experiments are conducted to demonstrate the effectiveness and efficiency of LGSQE.

Keywords: Generative model, quality evaluation, quality control, green learning, green AI.

*Corresponding author: Ganning Zhao, ganningz@usc.edu.

Received 18 November 2022; Revised 19 May 2023
ISSN 2048-7703; DOI 10.1561/116.00000076

1 Introduction

Image generative models have been widely used in various applications such as image generation, image inpainting, image-to-image translation, etc. With the advancement of generative models, measuring the quality of generated samples is needed. The evaluation of generative models has been an active research area. The methodology includes both subjective and objective ones. Subjective evaluation, which involves human visual comparison, is laborious. Furthermore, since an individual could be biased, getting a diversified group of human subjects in the evaluation is essential. Developing specific algorithms to compute objective measures is more economical and less subjective.

Quite a few quantitative metrics for generative models have been proposed [5, 6]. Examples include the Inception Score (IS) [50], the Fréchet Inception Distance (FID) [18], the Classifier two-sample test [40] and the Precision and Recall (P&R) [49], etc. Each metric has its strengths and weaknesses. IS can measure the quality and diversity of generated images but cannot detect mode dropping. FID performs better in detecting mode dropping. It is closer to human judgment and more robust to noise. Yet, its Gaussian assumption might not hold in practice. P&R uses two metrics to evaluate the quality and variety of generated data. It may fail to identify two identical distributions as shown in [43]. It is also not robust to outliers. The two-sample test [40] trains a binary classifier as a proxy to evaluate the quality of generated samples. Nevertheless, its performance is poor due to a lack of discriminant features.

All above-mentioned evaluation methods share two common problems. First, they apply to the whole generated dataset rather than an individual generated sample. If there is an effective and efficient mechanism to measure the quality of a sample, one can reject samples of poor quality on the fly. Then, this mechanism can serve as a post-processing step for quality control of generative samples. Second, most quality evaluation methods demand high computational complexity and a large model size. For example, many state-of-the-art methods use features from large networks (e.g., Inception-V3, VGG16, or ResNet) trained on the ImageNet dataset. Their evaluation is biased towards ImageNet and may not be generalized well to other datasets. Besides, large network models are challenging to deploy in mobile/edge computing. Although improvements have been made, e.g., [1, 19, 43], the two fundamental problems still exist.

We propose a lightweight generated sample quality evaluation (LGSQE) method to address them in this work. LGSQE consists of three main modules.

1. *Find simple yet effective representations for real/synthetic images from a source dataset.* LGSQE adopts the Successive Subspace Learning (SSL) [11, 12] under the green learning (GL) [30] framework to find a rich set of multi-scale spatial-spectral representations. SSL is an unsupervised represenation learning tool.

2. *Select discriminant features from the representation set obtained in the first module.* We choose a subset of discriminant representations as features based on a supervised feature learning method recently proposed in [62]. Then, we feed these selected features into a binary classifier in the third module.

3. *Conduct binary classification.* LGSQE trains a binary classifier to differentiate real samples and synthetic samples generated by a generative model. In the training stage, real and generated samples are labeled "zero" and "one", respectively. In the evaluation (or inference) stage, we get a soft label for each sample with a value between zero and one. The soft label of a generated sample serves as its quality index. Its quality is good (or bad) if its soft label is farther away from (or close to) one.

By aggregating quality indices of a large number of generated samples, LGSQE can offer quality metrics for their generative model as a byproduct. Intuitively, a poorly-performing (or high-performing) generative model tends to yield more (or less) bad samples. Since the distribution of generated data from a poorly-performing generative model is very different from that of real data, the test accuracy of the binary classifier should be higher. In contrast, the test accuracy for an ideal generative model should be close to the chance level (i.e. 0.5) for the whole dataset. To this end, we show that four metrics of a binary classifier can be used as evaluation metrics for generative models. They are: 1) accuracy, 2) the area under the curve (AUC), 3) precision, and 4) recall. Compared with the state-of-the-art FID metrics, the four LGSQE metrics preserve the same rank order of performance while their evaluation demands less time and lower memory requirement.

The integration of the above-mentioned three modules was applied to the object classification task in [60]. Yet, it has never been applied to quality evaluation of a generated sample, neither quality evaluation of a generative model. The novel application of existing tools to an important problem in the new context is the main contribution of our work. Besides, we need to show the effectiveness and efficiency of this new methodology in this application. To this end, we conduct extensive experiments to demonstrate the effectiveness of LGSQE. Furthermore, we report the model size and computational complexity to show the efficiency of LGSQE. Since no pretraining by another larger dataset (say, ImageNet) is needed, LGSQE has a small model size. The dataset chosen as the generation target (e.g., CIFAR-10, Celeb-A, LSUN etc.) is used to train LGSQE from scratch. Thus, LGSQE is dataset-specific.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The LGSQE method is detailed in Section 3. Experimental results are presented in Section 4. Concluding remarks and future research directions are given in Section 5.

2 Related Work

With the advancement of powerful deep-learning-based (DL-based) generative models, the performance evaluation of different generative models has received much attention. The focus has been on analyzing the gap between actual and synthetic data distributions. Besides assessing the synthesizing capability, it is desired that the evaluation tool can guide the design of more powerful models.

Many quantitative metrics for generative model evaluation have been proposed in the last 7-8 years. Yet, we see little work that leverages developed metrics to improve generative models. Furthermore, little research has been conducted on the quality evaluation of individually generated samples.

In this section, we first provide a general review to various evaluation metrics on generative models in Section 2.1. Then, we present classification-based quality metrics which was investigated by a few researchers in Section 2.2. Finally, the development of green learning is examined in Section 2.3 since it is highly related to tools adopted by the proposed LGSQE method.

2.1 Metrics for Generative Model Evaluation

The Inception Score (IS) [50] is one of the early-developed metrics. It exploits the Inception-Net pre-trained on ImageNet to calculate the KL-divergence between the conditional label distribution $p(y|x)$ and the marginal one $p(y)$ obtained from all samples, where x and y denote embeddings and labels, respectively. It has several limitations. First, since IS is susceptible to model overfitting [59], it may not generalize well. Second, it is inefficient in dealing with diversity between different modes, known as "mode collapse". To mitigate that, a Modified Inception Score was proposed in [17]. Third, as IS is pre-trained on ImageNet; it may measure image quality object-wise (rather than realistic-wise). Fourth, IS is sensitive to image resolution. Furthermore, rigorously speaking, IS is not a proper distance metric.

The Fréchet Inception Distance (FID) [18] is a popular metric. It employs Inception-V3 to map samples onto an embedding space, where joint Gaussian distributions model real and synthetic samples. FID is computed between the two Gaussians to measure their similarity. Notably, IS measures the quality and diversity of generated samples, while FID measures the distance between distributions. FID improves over IS in intra-class mode dropping and diversity handling between models. The metric has been further enhanced in [37] by introducing the Class-Aware Frechet Distance (CAFD) to increase its robustness. Yet, FID has its weakness. When their dimension is high, the log-likelihood distributions between real and synthesized samples are difficult to capture FID, and several other qualitative ones were compared in [53], which pointed out the weakness of the log-likelihood metric. Moreover, the Gaussian distribution assumption may not be valid [41].

A high-performance evaluation metric should measure the closeness of real and generated samples and the diversity of generated models. The precision-recall metric with a reference data manifold was introduced in [41] to unify these two aspects. Precision quantifies the former, while recall captures the latter. Yet, the metric is impractical in real applications since the reference manifold is not available in most cases. A way to quantify the tradeoff between precision and recall was proposed in [49]. Limitations of the precision-recall metric, such as failure to realize the match between identical distributions and robustness to outliers, were discussed in [43], where more reliable density and coverage metrics were proposed. On top of precision and recall, authenticity was added to form a three-dimensional metric in [1], which can characterize the generalization power of generative models better.

2.2 Classifier-based Metrics

The idea of classifier-based evaluation can date back to [40] and [19]. It employs a classifier to check whether one can differentiate real and synthetic samples easily. The classifier plays the role of a discriminator and its error rate is used for performance assessment. To give an example, the two-sample test [40] uses the k-nearest neighbor (KNN) classifier or the neural network classifier with one hidden layer trained on deep-layer embeddings of a DNN classifier (e.g., ResNet). An alternative is to evaluate class-conditional generative models as presented in [46]. The classifier is trained on synthetic data and used to predict labels of real data to yield the Classification Accuracy Score (CAS). Gu *et al.* [15] use multiple binary classifiers as a regressor to assess the quality of generated images. Their method can measure the quality of individual samples using the classifier's probability prediction as a quality index. However, their method requires generated images at different iterations of a complex CNN network, which are difficult to obtain and computationally expensive. In addition, their evaluation method may fail in the presence of complicated data distributions. The major difference between LGSQE and prior art in classification-based quality metrics is that LGSQE does not employ embeddings from DNNs as features for the classifier. It uses a more intuitive and explainable data processing pipeline to obtain features, leading to a lightweight and mathematically transparent solution.

2.3 Green Learning

Green learning (GL) [30] targets designing an efficient learning system with a small model size and fast training/inference time. It is suitable for mobile/edge computing. GL adopts a modularized design where each module is optimized

independently for implementational efficiency. GL was originated from pioneering research in [27, 28], which attempts to analyze the roles of nonlinear activation and convolution operations in neural networks. Afterward, several joint spatial-spectral transforms such as Saak [29] and Saab transforms [31] were proposed to extract image embeddings without backpropagation. Besides, one can have multiple Saab transforms in cascade to build learning systems such as Pixelhop [11], Pixelhop++ [12], E-Pixelhop [61], etc. More recently, a powerful feature selection tool called the Discriminant Feature Test (DFT) was proposed in [62]. It builds a bridge from image embeddings to discriminant features, which makes GL more mature.

Admittedly, many problems that DL solves well do not have a competitive GL solution since GL is still in its infancy [30]. Yet, there are still a few successful applications of GL. Examples include image classification [11, 12, 61], image enhancement [4], image quality assessment [42, 69], 3D medical image analysis [38], point cloud classification, segmentation, registration [20–22, 36, 65–68], face biometrics [47, 48], texture analysis and synthesis [34, 35, 64], deepfake image/video detection [7–9, 70], graph node classification [57, 58], etc.

3 Proposed LGSQE Method

Generated images have a wide range of resolutions. For example, image resolutions for the 7 datasets in the experiments range from 32×32 to 256×256. Here, we begin with images of lower resolutions and propose a basic LGSQE solution in Sections 3.1–3.3. Then, we define four LGSQE metrics in 3.4. Finally, we present an advanced LGSQE solution that handles images of higher resolutions in Section 3.5. The basic LGSQE method consists of three modules in cascade as shown in Figure 1. Each of them will be detailed below.

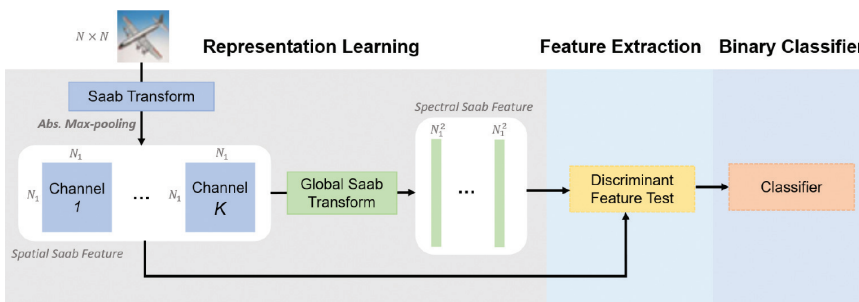


Figure 1: An overview of the LGSQE method, which consists of three modules. Module 1: Representation Learning, Module 2: Discriminant Feature Extraction and Module 3: Binary Classification.

3.1 Module 1: Representation Learning

In the first module, we find effective local and global representations of an image that are potentially useful for real/generated image classification. For the local representation, we extract blocks of dimension $F \times F \times C$ from input images of size $N \times N \times C$ with a stride of S pixels, where $F \times F$ is the filter size and C is the spectral number, i.e., $C = 1$ for gray-scale images and $C = 3$ for color images.

We apply the Saab transform [31] to the blocks and obtain a joint spatial-spectral representation. The Saab transform is a variant of the Principal Component Analysis (PCA) transform. It has two types of transform kernels: 1) the DC kernel, which gives the local average of pixels covered by the filter, and 2) AC kernels, which are data-driven kernels obtained by PCA. The reason to have two kernel types is that PCA can only be applied to zero-mean random vectors. By removing the local block mean, the block residual can be treated as a zero-mean random vector so that PCA can be applied.

The implementation of the Saab transform is summarized below.

1. Flatten the block of dimension $F \times F \times C$ into a long vector. The Saab transform will generate one DC response and $(F^2C - 1)$ AC responses. The DC response is obtained by applying the DC kernel to the block elements. The DC kernel is the constant element vector of unit length. The DC response is nothing but the block mean.

2. We obtain block residuals by removing block means. PCA is used to derive the AC kernels. Kernels associated with larger eigenvalues extract lower frequency components while kernels associated with smaller eigenvalues extract higher frequency ones. We can discard high-frequency components with very small values for dimension reduction. Each component extracted by a kernel is also called a channel. After the Saab transforms, the total number of channels is usually less than $K = F \times F \times C$ due to dimension reduction. The spatial size of the Saab coefficients is equal to $N_1 \times N_1$, where $N_1 = (N - F)/S + 1$.

The features extracted from the one-stage Saab transform are called the Hop-1 Saab coefficients. Hop-1 Saab coefficients only provide a local view with a small receptive field. Besides, they are spatial correlations among local-frequency Saab coefficients. Thus, it is desired to conduct the second-stage Saab transform. Generally, we can obtain fine-to-coarse hierarchical features by constructing multi-stage Saab transforms in cascade. The absolute max-pooling can be employed between two consecutive Saab transforms to enlarge the receptive field. The multi-stage transform leads to successive subspace learning. The multi-stage Saab transform can be further simplified using the channel-wise Saab transform (c/w Saab). That is, since there is little

correlation between different channels, we can conduct the Saab transform for an individual channel in the spatial domain only. The input to a regular Saab transform is a 3D tensor (2D spatial plus 1D spectral dimensions) while the input to a c/w Saab transform is a 2D tensor (2D spatial dimension).

3.2 Module 2: Discriminant Feature Extraction

Further dimension reduction at the expense of marginal performance degradation is required to reduce the number of model parameters and, eventually, the overall computational cost. Hence, we apply the Discriminant Feature Test (DFT) [62]. DFT independently computes the discriminant one-dimensional (1D) features and keeps only the most discriminant features for the evaluation module. Specifically, in computing the discriminant power of i^{th} 1D feature, we assign the ground truth labels to samples and compute the feature value range of $[f^i_{min}, f^i_{max}]$. We partition the range into two non-overlapping subsets S^i_L and S^i_R and search for the optimal partition point with the smallest weighted entropy of S^i_L and S^i_R as the DFT loss for a certain feature dimension. The entropy of the left partition is defined by

$$H^i_{L,t} = -\sum_{c=1}^{C} p^i_{L,c} log(p^i_{L,c}), \tag{1}$$

where t is the splitting point and $p^i_{L,c}$ is the probability of samples with class c over all samples in the left partition. The entropy of right partition $H^i_{R,t}$ can be computed similarly. The DFT loss for the $i-th$ feature is defined by

$$L_{DFT} = \min_{t \in T} H^i_t, \tag{2}$$

where T is the set of partition points t, H^i_t indicates the weighted average of entropy in left and right partitions. A lower DFT loss means higher class purity in the subsets, representing a stronger discriminant power. We iteratively apply the DFT to all features to obtain the DFT loss. We sort the features by their DFT loss in ascending order to draw the DFT loss curve. Figure 2 shows the DFT loss curve for MNIST and CIFAR-10 datasets. An obvious elbow point exists in each curve. We select the feature dimensions based on the elbow point as it minimizes the number of kept dimensions while maximizing the classification performance since other dimensions beyond that point are considered less discriminant. According to the curve, we select the first 400 and 800 features for MNIST and CIFAR-10 datasets respectively. We select the feature dimensions for other datasets in the same manner. Notice that we select the features according to their discriminant power. Therefore, depending on their discriminant nature, they may contain low or high-frequency components.

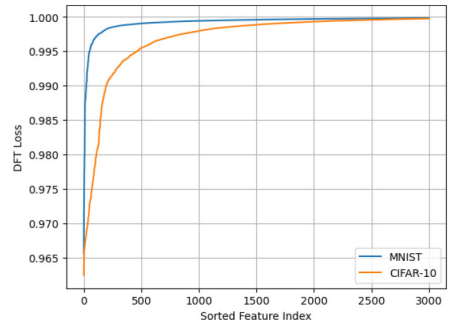


Figure 2: Feature selection curves of MNIST and CIFAR-10 datasets.

3.3 Module 3: Binary Classification

Real/generated data are partitioned into two disjoint sets - one for training and the other for testing. A binary classifier is trained on the union of real and generated training data, where real and generated samples are assigned with labels "0" and "1", respectively. Minor data imbalance is acceptable since it can be easily handled in most binary classifiers. The binary classifier will assign a soft score of $0 \leq d \leq 1$, on each of the tested samples. Usually, we choose a threshold, denoted by $0 \leq t \leq 1$, and make a hard decision depending on whether $0 \leq d < t$ or $1 \geq d \geq t$. The sample is claimed to be "real" (or zero) for the former and "generated" (or one) for the latter. In this work, we adopt the XGBoost(eXtreme Gradient Boosting) classifier [10] due to its high performance and reasonable model size.

3.4 LGSQE Quality Metric

LGSQE can serve as a quality metric for an individual sample or for a generative model with respect to a dataset as elaborated below.

Sample-based Quality Assessment. The soft label indicates the probability of the sample being assigned to the generated class. It serves as the quality evaluation index of each generated sample. If the soft label of a generated sample is closer to zero, it has a higher probability of a real sample (than that of a generated one). Its quality is good. On the other hand, if the soft label is closer to one, it has a higher probability of a generated sample (than that of a real one). Its quality is poor. The same argument applies to real samples. If a real sample has a soft label closer to one, it is an outlier from the distribution of real samples. It looks like a generated (or fake) one. For an ideal generative model, the distributions of real and generated samples should be the same. They are bell-shaped with peaks at 0.5. Then, classifier accuracy is about the chance-level accuracy (i.e., 50%)

Generative-Model-based Quality Assessment. We can evaluate the effectiveness of a generative model by aggregating the quality assessment of its generated samples of a sufficient amount. There are four commonly used performance metrics of a binary classifier. They are accuracy (Acc), precision (Pre), recall (Rec), and the area under the curve (AUC). Acc is the ratio of "correct decision number" over the "total decision number". There are two types of errors: FP (false positive) and FN (false negative). Then, precision and recall are defined by

$$\text{Pre} = \frac{TP}{TP + FP}, \quad \text{Rec} = \frac{TP}{TP + FN}. \tag{3}$$

One can draw the precision-recall curve by varying threshold t from zero to one and then calculate AUC. We choose $t = 0.5$ for the Acc metric.

For a specific dataset and a generative model, we use all four of them of the binary classifier in Module 3 as the LGSQE quality metrics. The Acc and AUC values of an ideal generative model are both equal to 0.5. It is the chance level against a mixed real/generated dataset, where the distributions of real and generated data in the feature space are completely interleaved and cannot be separated. In contrast, if the Acc and AUC values of a generative model are higher, the proposed LGSQE method can differentiate real and generated samples more easily. It means that the generative model is poorer. Similarly, a higher precision/recall value implies a poorer generative model.

It is worthwhile to point out that the precision and recall defined in [49] are different from our definitions as given in (3). Precision and recall are viewed as proxies of quality and diversity metrics, respectively, in [49]. Here, we take real/generated samples as negative/positive classes to compute the precision and recall values. The precision indicates the fraction of images predicted as generated data that are truly generated images. The recall represents the fraction of generated images successfully predicted as generated data.

3.5 Advanced LGSQE for Higher-Resolution Images

Images from the LSUN-Bedroom and the LSUN-Church datasets in Section 4 have a resolution of 256×256. Downsampling images to a smaller size directly may lose important detail information. To address this problem, we propose a two-scale processing pipeline to obtain high-resolution details and low-resolution global layout information jointly as shown in Figure 3. We downsample input images to 128×128 and feed them to two branches to extract both local and global discriminant features that are powerful in differentiating real and synthesized images.

- *Features for Local Details.* We use a sliding window of size 48×48 with stride 40 to extract 9 overlapping subimages from each input image.

These subimages are then fed to Module 1 and Module 2 as shown in Section 3 to find 100 features for each subimage. Then, each image contains $9 \times 100 = 900$ discriminant features, representing the local detail information. The Saab transform parameters are $F = 4$, $S = 3$, and $C = 3$.

- *Features for Global Layout.* We downsample input images to 48×48 and feed them to another branch to derive 900 discriminant features to capture the global information. The Saab transform parameters are $F = 3$, $S = 2$, and $C = 3$.

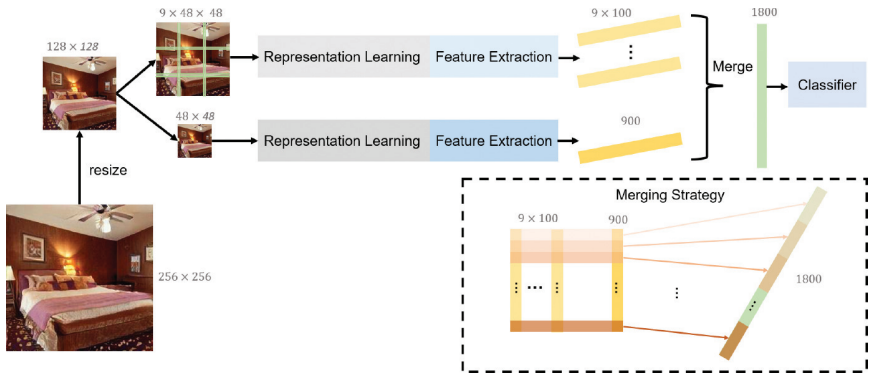


Figure 3: An overview of the advanced LGSQE method for higher-resolution images, which comprises two branches with identical architecture. The features for local details and global layout obtained from two branches are merged together by a merging strategy shown in the lower right of the figure.

We concatenate local and global features together to form a new feature set of 1,800 elements. We use them to train a binary classifier in Module 3 and apply the trained classifier in the evaluation stage. With this multi-scale representation strategy, LGSQE can be applied to high-resolution image datasets.

4 Experiments

In this section, we conduct experiments on seven datasets and compare the results of various evaluation metrics to demonstrate the validity of the proposed LGSQE method. At the same time, we conduct performance benchmarking of several generative models.

4.1 Experimental Setup

Datasets. We consider seven datasets and use them to train various generative models. The first two are gray-scale images while the last five comprise color images.

- *MNIST* [32]. We generate the same number of training and test images as the official MNIST dataset; namely, 60,000 training examples and 10,000 test examples. The Saab transform parameters are $F = 5$, $S = 2$ and $C = 1$. After removing Saab coefficients of extremely low energy, we apply the DFT to 3,000 remaining representations, select 400 features based on the elbow point of the DFT curve, and feed them to the binary classifier in Module 3.

- *Fashion-mnist* [56]. We adopt the same settings as those in MNIST.

- *CIFAR-10* [26]. Following the training/test split of the CIFAR-10 dataset, we generate 50,000 training images and 10,000 test images. The hyper-parameters of the Saab transform are $F = 3$, $S = 1$ and $C = 3$ (see Section 3.1). We remove Saab coefficients of extremely low energy values ($< 5 \times 10^{-5}$) and conduct DFT on 3,500 remaining representations. Based on the elbow point of the DFT curve, we select 800 features and feed them to the binary classifier in Module 3.

- *STL-10* [13]. To be consistent with evaluation results reported by other papers, we use the Lanczos interpolation to downsample the resolution of input color images from 96×96 to 48×48 in the STL-10 dataset. We generate 50,000 training images and 10,000 test images. The Saab transform parameters are $F = 3$, $S = 2$, and $C = 3$. We select 800 features based on the DFT curve and feed them into the binary classifier in Module 3.

- *LSUN-Bedroom* and *LSUN-Church* [63]. We use various generative models to generate 10,000 training images and 2,000 test images.

- *Celeb-A* [39]. We downsample the resolution of input color images to 48×48 and generate 50,000 training images and 10,000 test images. The Saab transform parameters are $F = 3$, $S = 2$, and $C = 3$. We use DFT to select 800 discriminant features for the binary classifier.

Generative Models. We adopt different generative models for different datasets. They are summarized below.

- The generative models for MNIST and Fashion-MNIST include GAN [14], WGAN [2] and WGAN-GP [16]. All models are trained from scratch.

- The generative models for CIFAR-10 include DCGAN [45], StyleFormer [44], StyleGAN2-ADA [24], Diffusion-StyleGAN2 [55] and StyleGAN-XL [52]. We train DCGAN from scratch to obtain the generative model. We use the weights from the official paper repositories for the remaining models.
- The generative models for STL-10 include Styleformer, Diffusion-StyleGAN2, Diffusion-ProjectedGAN, and E2GAN [54].
- The generative models for Celeb-A include Styleformer and Diffusion-StyleGAN2.
- The generative models for LSUN include Styleformer, Diffusion-StyleGAN2, Diffusion-ProjectedGAN, ProjectedGAN [51], StyleGAN [25] and ProgressiveGAN [23].

Hyper-Parameters of XGBoost. We adopt the XGBoost (extreme gradient boosting) classifier [10] in the third module. The maximum depth of a tree is set to one to reduce the computational cost. Figure 4 shows the classification accuracy as a function of the tree number for the XGBoost classifier. A higher classification accuracy indicates a poorer generative model. The accuracy increases with the tree number at the cost of higher complexity and memory. To cut down the computational cost, we stop adding more trees at the saturation point of the validation data. The chosen tree numbers for MNIST, Fashion-mnist, CIFAR-10, STL-10, Celeb-A, LSUN-Bedroom and LSUN-Church are 650, 650, 1250, 1250, 1250, 3000 and 3000 respectively.

4.2 Results and Analysis

4.2.1 Quality Evaluation of Generated and Real Samples

Figure 5 shows the soft label histograms of generated and real samples for four datasets. Diffusion-StyleGAN2 is used to generate the synthetic images on (a) and (b). WGAN-GP is used on (c) and (d). Since the performance of WGAN-GP is not ideal, most samples have soft label values close to zero for real and one for generated samples. Figure 6 shows the soft label histogram of samples generated by Styleformer. The performance of Styleformer is better than Diffusion-StyleGAN2 on the CIFAR-10 dataset and inferior on the STL-10 dataset since it has more samples of probabilities close to 0.5 (see Figure 5-a) and more samples of probabilities close at the tails of the distribution (see Figure 5(b)).

4.2.2 Visualization of Generated and Real Samples

Several generated image samples trained by the LSUN-bedroom and the LSUN-church datasets are visualized in Figures 7 and 8, respectively. We also show

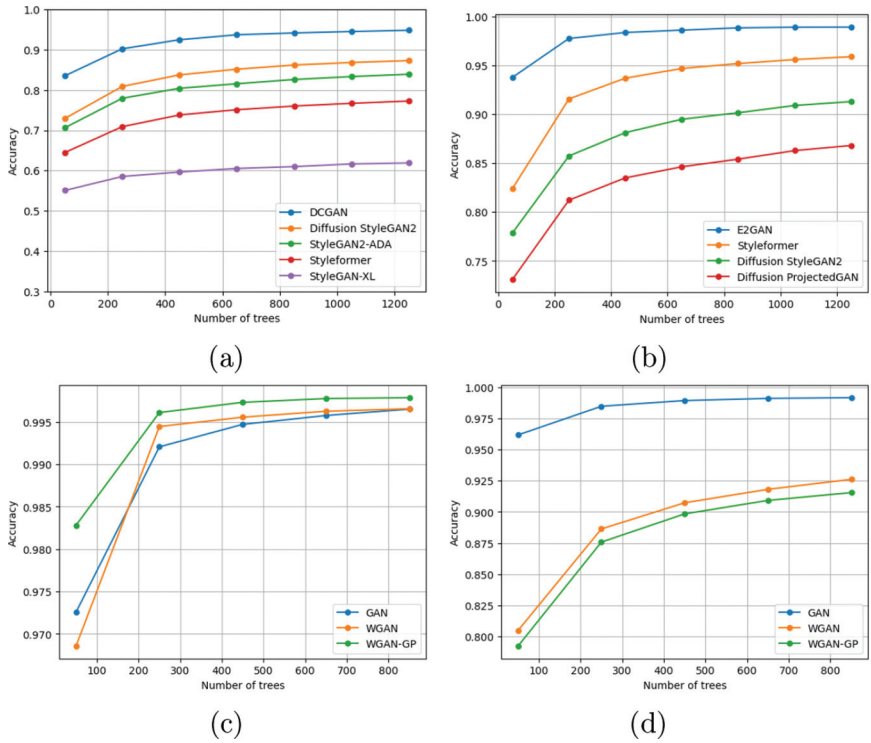


Figure 4: The classification accuracy as a function of tree numbers in the XGBoost classifier for four experiment datasets: (a) CIFAR-10, (b) STL-10, (c) MNIST, and (d) Fashion-mnist.

their associated LGSQE quality indices and the histogram of a large number of generated samples. The former is assigned by the binary classifier. A generated image has a quality index closer to zero if it appears to be a real one viewed by human eyes. The latter indicates the capability of the corresponding generative model. More high- and low-quality images generated by certain generative models against the LSUN-church, LSUN-bedroom, Celeb-A and MNIST datasets are given in Figures 11–18 in the Appendix. Each figure contains 88 generated images. They are evaluated as high- or low-quality images using the proposed LGSQE method.

4.2.3 LGSQE as A Post-processing Tool

LGSQE can be used as a post-processing tool to boost the performance of generative models and improve the performance of downstream tasks (e.g., semantic segmentation). By sorting the generated samples by their soft labels in

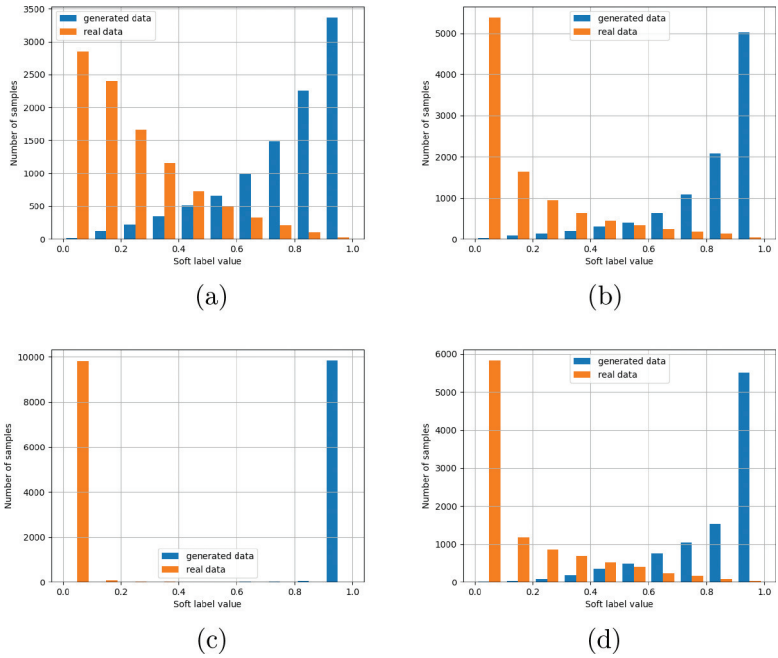


Figure 5: The soft label histograms of generated and real samples on (a) CIFAR-10, (b) STL-10, (c) MNIST, and (d) Fashion-mnist datasets, where the samples are generated by Diffusion StyleGAN2 in (a) and (b) and WGAN-GP in (c) and (d).

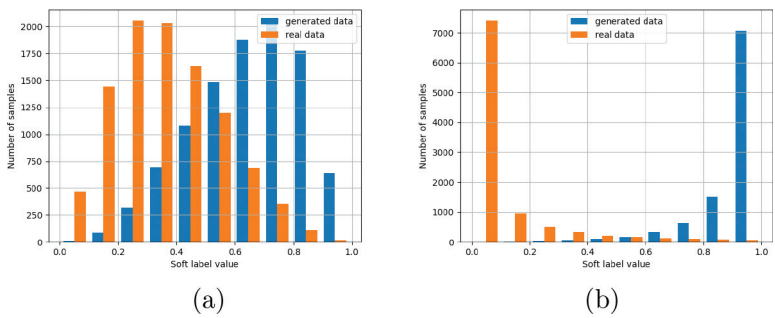


Figure 6: The soft label histograms of generated and real samples on (a) CIFAR-10 and (b) STL-10 datasets with Styleformer as the generative model.

an ascending order and keeping certain percentages of the best quality samples, which have the lowest LGSQE quality indices, one can lower classification accuracy and FID scores of the kept samples against the real samples as shown in Figure 9. We should emphasize that lower classification accuracy and smaller

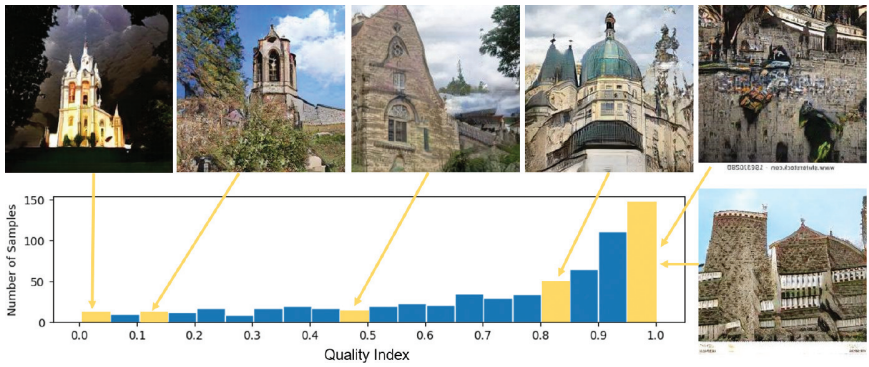


Figure 7: Generated LSUN-church samples from the ProgressiveGAN model and their associated LGSQE quality indices, where a smaller quality index value indicates that the sample is more like a real one. The histogram of a large number of generated samples is also given to show the capability of the ProgressiveGAN model.

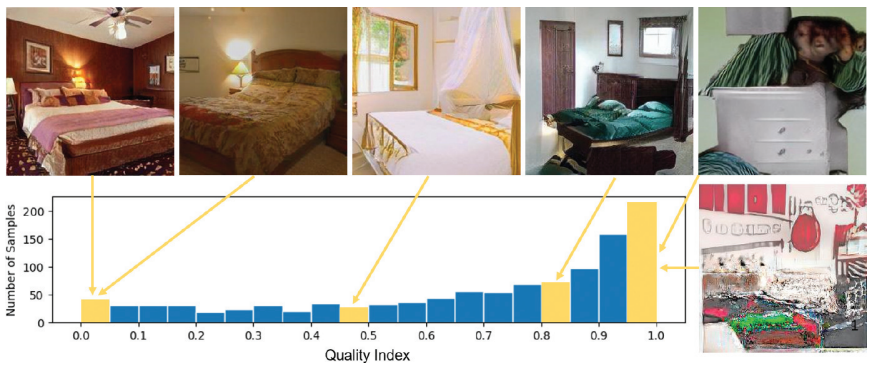
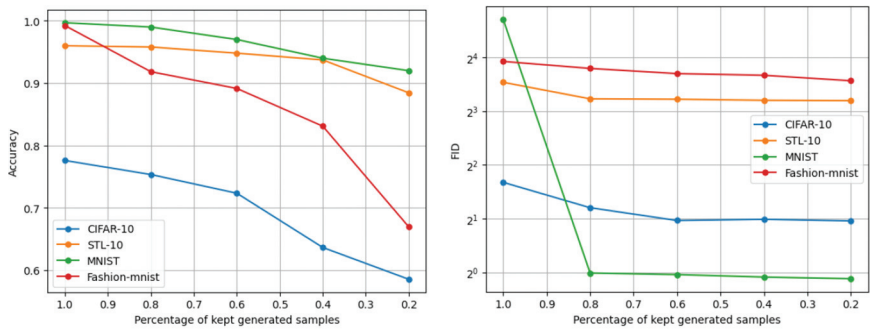


Figure 8: Generated LSUN-bedroom samples from the ProjectedGAN model and their associated LGSQE quality indices, where a smaller quality index value indicates that the sample is more like a real one. The histogram of a large number of generated samples is also given to show the capability of the ProgressiveGAN model.

FID scores imply better quality of generated samples since generated and real samples are difficult to separate.

4.2.4 Quality Evaluation of Generative Models

Apart from being used as a sample-based quality measure, LGSQE can provide evaluation metrics for a generative model by aggregating the quality indices from its generated samples. As discussed in Section 3.4, we use the classification accuracy (Acc), AUC, precision, and recall as four LGSQE evaluation metrics



(a) (b)

Figure 9: The classification accuracy and the FID score are plotted as functions of the percentages of kept samples in (a) and (b), respectively, where samples of the poorest quality are removed first.

for generative models. Since FID is the most popular evaluation metric, we compare the ranking of FID with those of four LGSQE metrics for MNIST, Fashion-mnist, CIFAR-10, STL-10, LSUN-Church, LSUN-Bedroom, and Celeb-A datasets in Tables 1-7, respectively. We arrange generative models based on their FID scores from the largest to the smallest in each table, which correspond to the weakest and the strongest generative models, respectively. As shown in the five tables, the rankings offered by the FID scores of generative models are consistent with those of the four metrics of LGSQE. These experiments demonstrate the effectiveness of the proposed LGSQE method in measuring the power of generative models.

Table 1: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 3 generative models (WGAN, GAN, and WGAN-GP) for the MNIST dataset.

MNIST dataset	FID	Acc	AUC	Precision	Recall
WGAN	32.37	0.997	0.999	0.996	0.998
GAN	26.56	0.996	0.998	0.994	0.997
WGAN-GP	26.12	0.864	0.931	0.870	0.868

Table 2: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 3 generative models (GAN, WGAN, and WGAN-GP) for the Fashion-mnist dataset.

Fashion-mnist dataset	FID	Acc	AUC	Precision	Recall
GAN	62.22	0.991	0.999	0.990	0.991
WGAN	26.58	0.926	0.980	0.920	0.931
WGAN-GP	15.19	0.915	0.969	0.917	0.913

Table 3: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 5 generative models (DCGAN, Diffusion-StyleGAN2, StyleGAN2-ADA, Styleformer, and StyleGAN-XL) for the CIFAR-10 dataset.

CIFAR-10 dataset	FID	Acc	AUC	Precision	Recall
DCGAN	47.7	0.950	0.990	0.954	0.946
Diffusion-StyleGAN2	3.19	0.877	0.948	0.879	0.876
StyleGAN2-ADA	2.92	0.842	0.919	0.843	0.842
Styleformer	2.82	0.776	0.859	0.773	0.782
StyleGAN-XL	1.85	0.622	0.680	0.616	0.649

Table 4: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 4 generative models (E2GAN, StyleFormer, Diffusion-StyleGAN2, and Diffusion-ProjectedGAN) for the STL-10 dataset.

STL-10 dataset	FID	Acc	AUC	Precision	Recall
E2GAN	25.4	0.989	0.999	0.985	0.993
StyleFormer	15.2	0.960	0.991	0.947	0.975
Diffusion-StyleGAN2	11.6	0.914	0.973	0.907	0.924
Diffusion-ProjectedGAN	6.91	0.871	0.946	0.846	0.906

Table 5: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 5 generative models (StyleFormer, ProgressiveGAN, Diffusion-StyleGAN2, Diffusion-ProjectedGAN, and ProjectedGAN) for the LSUN-church dataset.

LSUN-Church dataset	FID	Acc	AUC	Precision	Recall
StyleFormer	7.99	0.998	0.999	0.997	0.998
ProgressiveGAN	6.42	0.904	0.971	0.912	0.893
Diffusion-StyleGAN2	3.17	0.892	0.963	0.896	0.887
Diffusion-ProjectedGAN	1.85	0.874	0.946	0.876	0.872
ProjectedGAN	1.59	0.861	0.946	0.876	0.840

Table 6: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 5 generative models (ProgressiveGAN, Diffusion-StyleGAN2, Diffusion-ProjectedGAN, ProjectedGAN, and StyleGAN) for the LSUN-bedroom dataset.

LSUN-Bedroom dataset	FID	Acc	AUC	Precision	Recall
ProgressiveGAN	8.34	0.894	0.962	0.894	0.893
Diffusion-StyleGAN2	3.65	0.876	0.948	0.873	0.880
StyleGAN	2.65	0.867	0.944	0.866	0.869
ProjectedGAN	1.52	0.845	0.925	0.852	0.835
Diffusion-ProjectedGAN	1.43	0.824	0.910	0.818	0.833

Table 7: Comparison of 5 evaluation metrics (FID, Acc, AUC, Precision, and Recall) on 2 generative models (Styleformer and Diffusion-StyleGAN2) for the Celeb-A dataset.

Celeb-A dataset	FID	Acc	AUC	Precision	Recall
StyleFormer	3.66	0.975	0.997	0.955	0.998
Diffusion-StyleGAN2	1.69	0.942	0.946	0.944	0.987

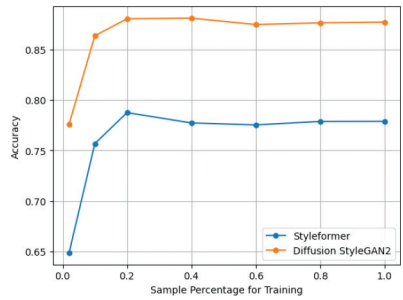


Figure 10: Training on a partial dataset for LGSQE.

4.2.5 Weak Supervision

The behavior of the LGSQE metric is relatively stable against the number of training samples. We show the classification accuracy metric of LGSQE as a function of the percentages of the total samples for two generative models, Styleformer and Diffusion-StyleGAN, with respect to the CIFAR-10 dataset in Figure 10 (a). The accuracy metric remains at the same level from 20% to 100% of samples. It means that LGSQE can provide the same evaluation performance with only 20% training samples. Such a phenomenon is observed for all generative models in all datasets. For comparison, since other evaluation metrics rely on features extracted from complex networks such as Inception-V3, more generative samples are needed to avoid over-fitting. Furthermore, since the filters used in LGSQE are directly obtained from the evaluation dataset, they do not have any bias with a pre-trained dataset such as the ImageNet.

4.3 Comparison of Model Sizes and Evaluation Time

In this subsection, we examine the efficiency of generative model quality evaluation methods by comparing their model sizes (in terms of the number of model parameters) and computational complexity (in terms of floating-point operations per pixel, or FLOPs/pixel). Most state-of-the-art evaluation methods extract features from large networks pre-trained by the ImageNet (e.g., Inception-v3, VGG-16, or ResNet-34). We compare the computational complexity and the model sizes of the entire LGSQE method and the feature

Table 8: Comparison of computational complexity and model sizes of LGSQE and pretrained networks used for feature extraction in DL-based evaluation metrics.

Complexity		FLOPs/pixel	No. of Parameters
VGG-16		102.97K	138.36M
Inception-V3		22.37K	25M
ResNet-34		24.45K	21.8M
Entire LGSQE	MNIST / Fashion-mnist	137.82	0.95M
	CIFAR-10	142.30	2.89M
	STL-10 / Celeb-A	69.46	3.17M
	LSUN	69.62	3.76M

extraction modules of DL-based methods in Table 8. We see a huge gap in both computational complexity and model sizes. Clearly, LGSQE is much more efficient.

We also measure the evaluation time with a computer with Intel(R) Xeon(R) CPU E5-2620 v3 at 2.40 GHz without code optimization. It is not surprising that LGSQE is much faster than other methods. Take the CIFAR-10 dataset as an example. It takes 122 minutes for the FID computation on 10,000 pairs of actual and generated images. In contrast, it takes LGSQE 2-3 minutes to achieve the same task.

5 Conclusion and Future Work

A lightweight quality evaluation method for generated samples and generative models, called LGSQE, was proposed in this paper. Compared with deep-learning-based evaluation metrics, LGSQE offers discriminant features that can accurately differentiate generated from real samples. LGSQE is more transparent to users due to its modularized design. Users can adjust hyper-parameters to fine-tune each module. Furthermore, it has a smaller model size and shorter evaluation time. In the future, we would like to use LGSQE as a discriminator to boost the performance of light-weight generative models such as NITES [34], TGHop [35], PAGER [3], and GENHOP [33].

Acknowledgements

This project was sponsored by US DoD LUCI (Laboratory University Collaboration Initiative) fellowship and US Army Research Laboratory. The authors also acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources.

Appendix

Figure 11: Illustration of 88 high-quality images generated by Diffusion-PojectedGAN trained on LSUN-church.

Figure 12: Illustration of 88 low-quality images generated by Diffusion-PojectedGAN trained on LSUN-church.

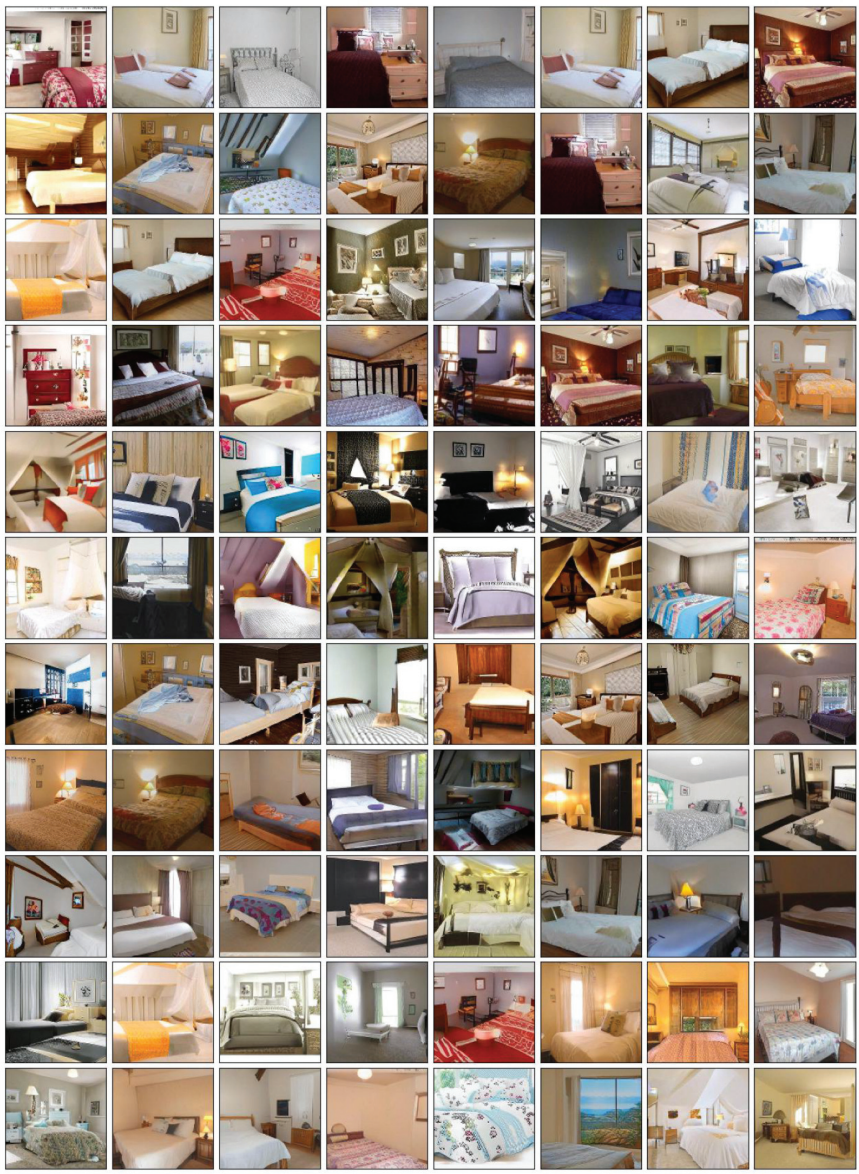


Figure 13: Illustration of 88 high-quality images generated by PojectedGAN trained on LSUN-bedroom.

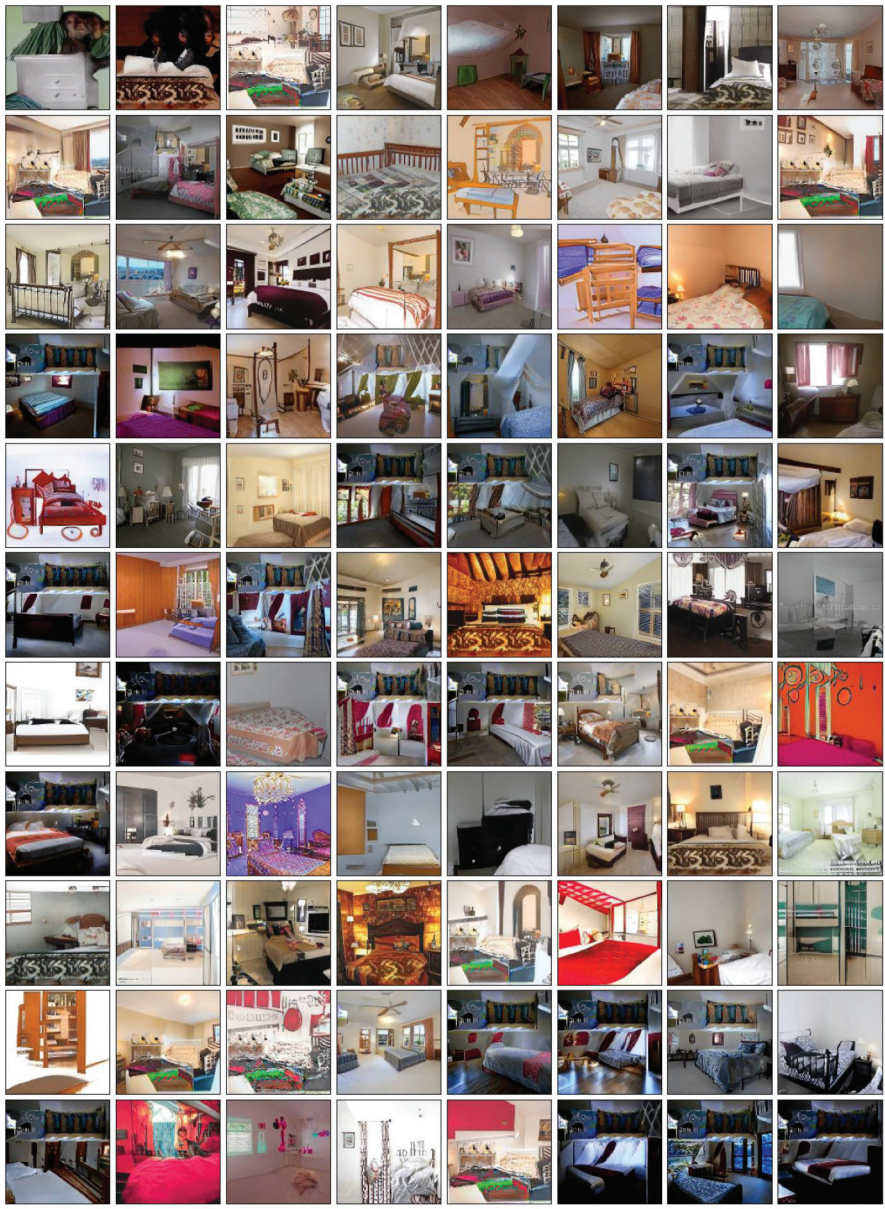


Figure 14: Illustration of 88 low-quality images generated by PojectedGAN trained on LSUN-bedroom.

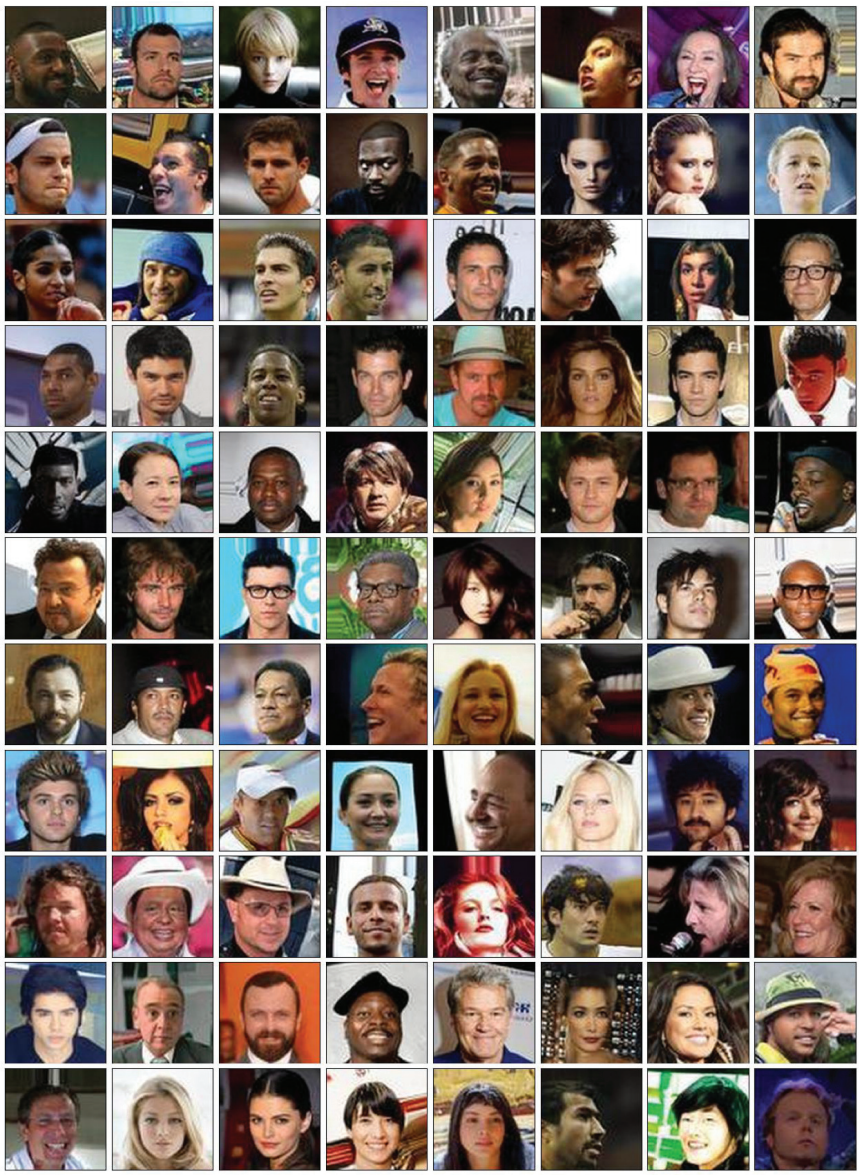


Figure 15: Illustration of 88 high-quality images generated by Diffusion-StyleGAN2 trained on CelebA.

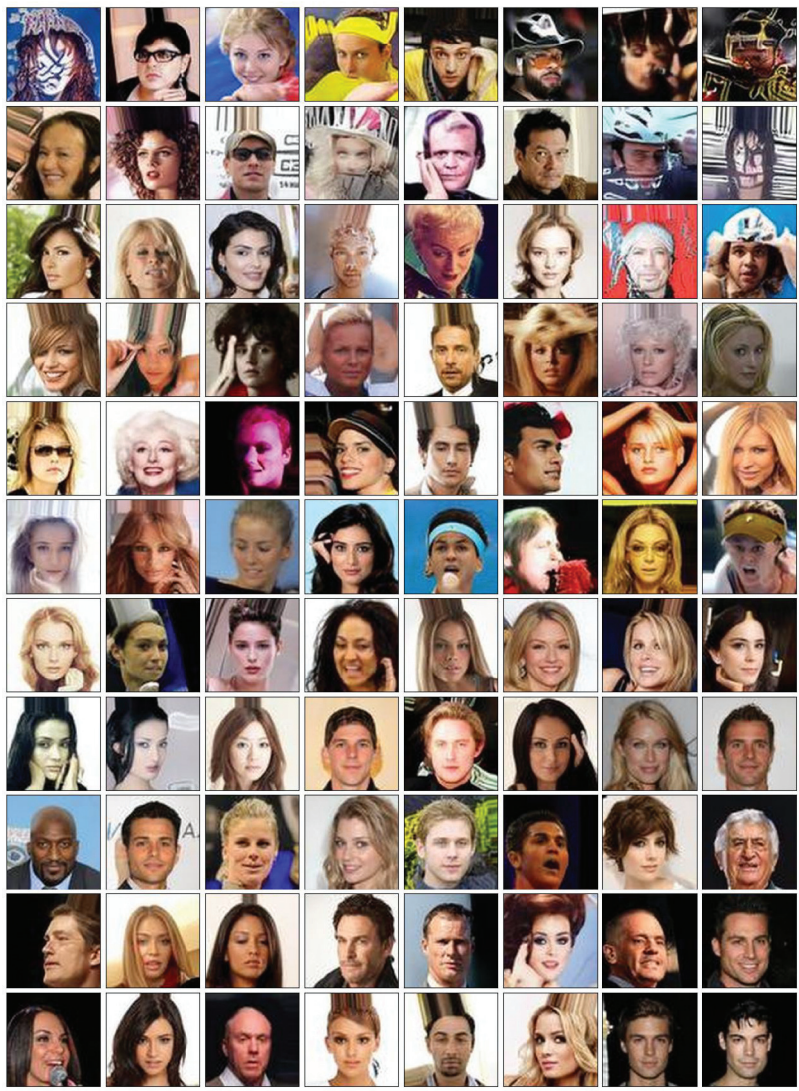


Figure 16: Illustration of 88 low-quality images generated by Diffusion-StyleGAN2 trained on CelebA.

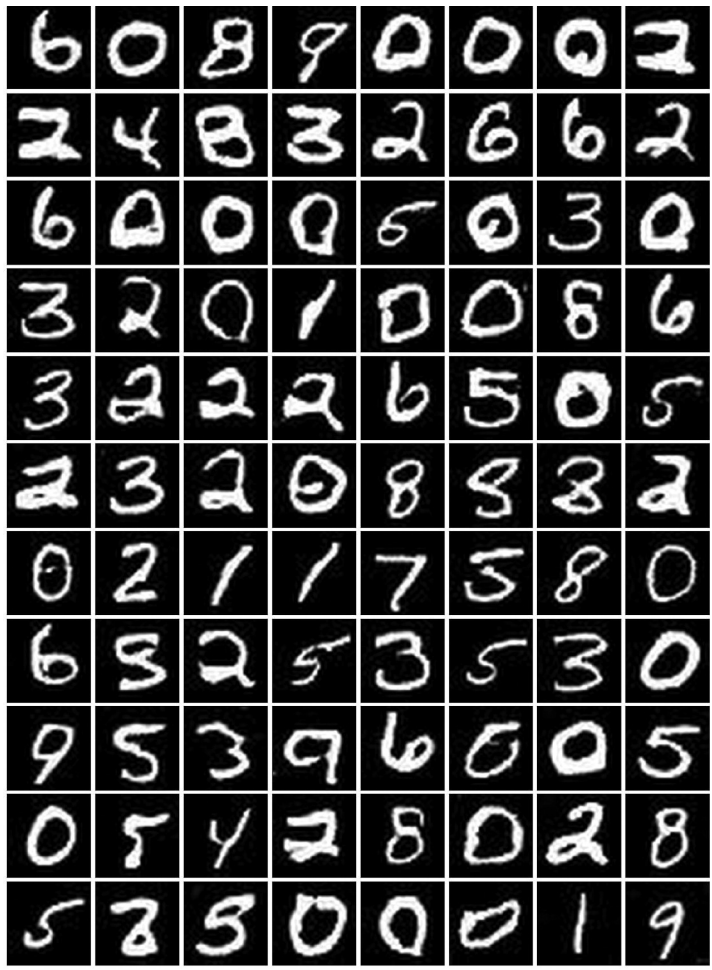


Figure 17: Illustration of 88 high-quality images generated by WGAN-GP trained on MNIST.

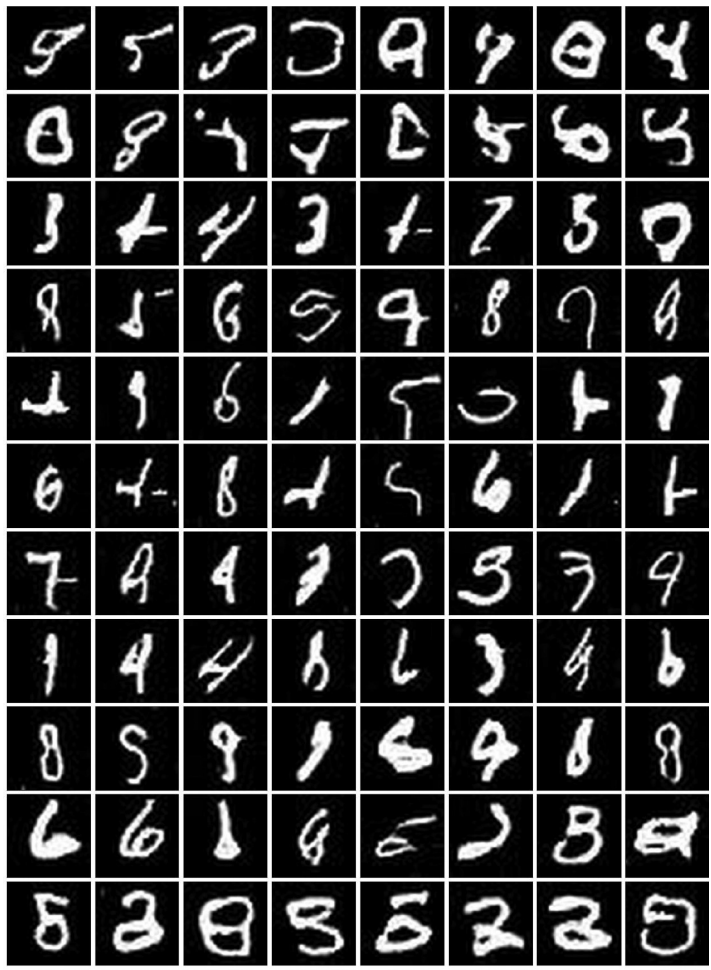


Figure 18: Illustration of 88 low-quality images generated by WGAN-GP trained on MNIST.

References

[1] A. Alaa, B. Van Breugel, E. S. Saveliev, and M. van der Schaar, "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models," in *International Conference on Machine Learning*, PMLR, 2022, 290–306.

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, 214–23.

[3] Z. Azizi and C.-C. J. Kuo, "PAGER: Progressive Attribute-Guided Extendable Robust Image Generation," *arXiv preprint arXiv:2206.00162*, 2022.

[4] Z. Azizi, X. Lei, and C.-C. J. Kuo, "Noise-Aware Texture-Preserving Low-Light Enhancement," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 443–6.

[5] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, 179, 2019, 41–65.

[6] A. Borji, "Pros and cons of GAN evaluation measures: New developments," *Computer Vision and Image Understanding*, 215, 2022, 103329.

[7] H.-S. Chen, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop++: An Enhanced Lightweight Deepfake Detector," *arXiv preprint arXiv:2205.00211*, 2022.

[8] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[9] H.-S. Chen, K. Zhang, S. Hu, S. You, and C.-C. J. Kuo, "Geo-DefakeHop: High-Performance Geographic Fake Image Detection," *arXiv preprint arXiv:2110.09795*, 2021.

[10] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, 785–94.

[11] Y. Chen and C.-C. J. Kuo, "Pixelhop: A successive subspace learning (SSL) method for object recognition," *Journal of Visual Communication and Image Representation*, 2020, 102749.

[12] Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, "Pixelhop++: A small successive-subspace-learning-based (ssl-based) model for image classification," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3294–8.

[13] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2011, 215–23.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 27, 2014.

[15] S. Gu, J. Bao, D. Chen, and F. Wen, "Giqa: Generated image quality assessment," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, Springer, 2020, 369–85.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, 30, 2017.

[17] S. Gurumurthy, R. Kiran Sarvadevabhatla, and R. Venkatesh Babu, "Deligan: Generative adversarial networks for diverse and limited data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 166–74.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, 30, 2017.

[19] D. J. Im, H. Ma, G. Taylor, and K. Branson, "Quantitatively evaluating GANs with divergences proposed for training," *arXiv preprint arXiv:1803.01045*, 2018.

[20] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "R-PointHop: A Green, Accurate and Unsupervised Point Cloud Registration Method," *arXiv preprint arXiv:2103.08129*, 2021.

[21] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "Unsupervised point cloud registration via salient points analysis (SPA)," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 5–8.

[22] P. Kadam, Q. Zhou, S. Liu, and C.-C. J. Kuo, "PCRP: Unsupervised Point Cloud Object Retrieval and Pose Estimation," *arXiv preprint arXiv:2202.07843*, 2022.

[23] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[24] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Advances in Neural Information Processing Systems*, 33, 2020, 12104–14.

[25] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, 4401–10.

[26] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] C.-C. J. Kuo, "The CNN as a guided multilayer RECOS transform [lecture notes]," *IEEE signal processing magazine*, 34(3), 2017, 81–9.

[28] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, 41, 2016, 406–13.

[29] C.-C. J. Kuo and Y. Chen, "On data-driven saak transform," *Journal of Visual Communication and Image Representation*, 50, 2018, 237–46.

[30] C.-C. J. Kuo and A. M. Madni, "Green Learning: Introduction, Examples and Outlook," *arXiv preprint arXiv:2210.00965*, 2022.

[31] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design," *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), 1998, 2278–324.

[33] X. Lei, W. Wang, and C.-C. J. Kuo, "GENHOP: An Image Generation Method Based on Successive Subspace Learning," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2022, 3314–8.

[34] X. Lei, G. Zhao, and C.-C. J. Kuo, "NITES: A Non-Parametric Interpretable Texture Synthesis Method," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2020, 1698–706.

[35] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "TGHop: an explainable, efficient, and lightweight method for texture generation," *APSIPA Transactions on Signal and Information Processing*, 10, 2021.

[36] S. Liu, M. Zhang, P. Kadam, and C.-C. J. Kuo, *3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods*, Springer.

[37] S. Liu, Y. Wei, J. Lu, and J. Zhou, "An improved evaluation framework for generative adversarial networks," *arXiv preprint arXiv:1803.07474*, 2018.

[38] X. Liu, F. Xing, C. Yang, C.-C. J. Kuo, S. Babu, G. E. Fakhri, T. Jenkins, and J. Woo, "VoxelHop: Successive Subspace Learning for ALS Disease Classification Using Structural MRI," *arXiv preprint arXiv:2101.05131*, 2021.

[39] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[40] D. Lopez-Paz and M. Oquab, "Revisiting classifier two-sample tests," *arXiv preprint arXiv:1610.06545*, 2016.

[41] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," *Advances in neural information processing systems*, 31, 2018.

[42] Z. Mei, Y.-C. Wang, X. He, and C.-C. J. Kuo, "GreenBIQA: A Lightweight Blind Image Quality Assessment Method," *arXiv preprint arXiv:2206.14400*, 2022.

[43] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *International Conference on Machine Learning*, PMLR, 2020, 7176–85.

[44] J. Park and Y. Kim, "Styleformer: Transformer based generative adversarial networks with style vector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 8983–92.

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

[46] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," *Advances in neural information processing systems*, 32, 2019.

[47] M. Rouhsedaghat, Y. Wang, X. Ge, S. Hu, S. You, and C.-C. J. Kuo, "Facehop: A light-weight low-resolution face gender classification method," in *International Conference on Pattern Recognition*, Springer, 2021, 169–83.

[48] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C. J. Kuo, "Low-resolution face recognition in resource-constrained environments," *Pattern Recognition Letters*, 149, 2021, 193–9.

[49] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," *Advances in neural information processing systems*, 31, 2018.

[50] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, 29, 2016.

[51] A. Sauer, K. Chitta, J. Müller, and A. Geiger, "Projected gans converge faster," *Advances in Neural Information Processing Systems*, 34, 2021, 17480–92.

[52] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, 2022, 1–10.

[53] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[54] Y. Tian, Q. Wang, Z. Huang, W. Li, D. Dai, M. Yang, J. Wang, and O. Fink, "Off-policy reinforcement learning for efficient and effective gan architecture search," in *European Conference on Computer Vision*, Springer, 2020, 175–92.

[55] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-GAN: Training GANs with Diffusion," *arXiv preprint arXiv:2206.02262*, 2022.

[56] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[57] T. Xie, R. Kannan, and C.-C. J. Kuo, "GraphHop++: New Insights into GraphHop and Its Enhancement," *arXiv preprint arXiv:2204.08646*, 2022.

[58] T. Xie, B. Wang, and C.-C. J. Kuo, "GraphHop: An Enhanced Label Propagation Method for Node Classification," *arXiv preprint arXiv:2101.02326*, 2021.

[59] J. Yang, A. Kannan, D. Batra, and D. Parikh, "Lr-gan: Layered recursive generative adversarial networks for image generation," *arXiv preprint arXiv:1703.01560*, 2017.

[60] Y. Yang, H. Fu, and C.-C. J. Kuo, "Design of supervision-scalable learning systems: Methodology and performance benchmarking," *arXiv preprint arXiv:2206.09061*, 2022.

[61] Y. Yang, V. Magoulianitis, and C.-C. J. Kuo, "E-pixelhop: An enhanced pixelhop method for object classification," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2021, 1475–82.

[62] Y. Yang, W. Wang, H. Fu, and C.-C. J. Kuo, "On Supervised Feature Selection from High Dimensional Feature Spaces," *arXiv preprint arXiv:2203.11924*, 2022.

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

[64] K. Zhang, H.-S. Chen, Y. Wang, X. Ji, and C.-C. J. Kuo, "Texture Analysis Via Hierarchical Spatial-Spectral Correlation (HSSC)," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, 4419–23.

[65] M. Zhang, P. Kadam, S. Liu, and C.-C. J. Kuo, "GSIP: Green Semantic Segmentation of Large-Scale Indoor Point Clouds," *arXiv preprint arXiv:2109.11835*, 2021.

[66] M. Zhang, P. Kadam, S. Liu, and C.-C. J. Kuo, "Unsupervised Feedforward Feature (UFF) Learning for Point Cloud Classification and Segmentation," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 144–7.

[67] M. Zhang, Y. Wang, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop++: A Lightweight Learning Model on Point Sets for 3D Classification," *arXiv preprint arXiv:2002.03281*, 2020.

[68] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop: An Explainable Machine Learning Method for Point Cloud Classification," *IEEE Transactions on Multimedia*, 2020.

[69] X. Zhang, S. Kwong, and C.-C. J. Kuo, "Data-Driven Transform-Based Compressed Image Quality Assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9), 2020, 3352–65.

[70] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum, and C.-C. J. Kuo, "A-PixelHop: A Green, Robust and Explainable Fake-Image Detector," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 8947–51.