Does a GAN leave distinct model-specific fingerprints?

Yuzhen Ding
Yuzhen.Ding@asu.edu
Nupur Thakur
nsthaku1@asu.edu
Baoxin Li
Baoxin.Li@asu.edu

Ira A. Fulton Schools of Engineering
Arizona State University
699 S Mill Ave, Tempe, AZ 85281

Abstract

Generative Adversarial Networks (GANs) have been breaking their own records in terms of the quality of the synthesized images, which could be so high as to make it impossible to distinguish generated images from real ones by human eyes. This has raised threats to security and privacy-sensitive applications, and thus it is important to be able to tell if an image is generated by GANs, and better yet, by which GAN. The task is in a sense similar to digital image forensics for establishing image authenticity, but the literature has inconclusive reports as to whether GANs leave unique fingerprints in the generated images. In this paper, we attempt to develop a comprehensive understanding towards answering this question. We propose a model to extract fingerprints that can be viewed largely as GAN-specific. We further identify a few key components that contribute to defining the fingerprint of the generated images. Using experiments based on state-of-the-art GAN models and different datasets, we evaluate the performance of our model and verify the major conclusions of our analysis.

1 Introduction

With advancements in generative adversarial networks, photo-realistic image generation and manipulation has seen exponential growth. The state-of-the-art generative models ([15, 16, 36]) have made it fairly easy to generate realistic visual contents that exhibit little perceivable artifacts. As these networks are gaining popularity, there is a growing concern of their misuse. For example, they may be used to misguide the public in an election campaign by faking a video of a politician or even to fool an autonomous vehicle into taking actions with catastrophic consequences. And indeed there have been many recent reports of misuse of GANs ([10, 11, 28]). Because of the potential high-impact misuse of such techniques, research on analyzing GAN-generated contents has been drawing increasing attention.

One line of research is on detecting fake imagery generated by GANs, posing as a binary classification problem. On this regard, some leading approaches focused on specific problem domains like deepfakes via face-swapping ([1, 2, 14]). In the meantime, there are also approaches that were tuned to, and thus work only for, specific GAN architectures, e.g., ([27]). Another line of research attempts to answer a more fundamental question: whether GANs leave unique traces in the generated images, analogous to digital fingerprints used in forensic image analysis (where an imaging device is believed to leave a unique and stable mark on the acquired data, due to, e.g., unique sensor noise patterns and/or peculiar in-camera processing units). Although prior research ([23, 33, 35]) has alluded to the existence of such GAN fingerprints, research on this regard is still scarce and existing works appear to be inconclusive, sometimes reporting conflicting answers to the question. Furthermore, even if there might be some "GAN fingerprints" that are associated with the generated images, questions like how to extract them, how reliable they are, and how they depend on various components of the learning process remain largely unexplored.

In this work, we attempt to develop a comprehensive understanding towards answering the above questions. The abundant choices of GAN models make it less attractive to employ an architecture-specific approach. Also, similarities (in architecture and/or learning algorithm) among GAN models suggests that there may be some clustering structure of the potential fingerprints. Furthermore, rich visual contents that may be generated by GANs suggest that GAN fingerprints, if they exist, should present themselves in more abstract forms than image-level features.

For these considerations, we propose to employ a hierarchical Bayesian approach to GAN fingerprint modeling: different level latent representations are used for capturing different level features for the fingerprints. For implementation, we propose to embed this hierarchical modeling inside an architecture similar to variational autoencoder (VAE), leading to a deep model termed BFR-VAE. Through extensive experiments using several popular GAN models, we demonstrate that the proposed model is able to extract GAN fingerprints that are largely model-specific. We will also show how the model is used for analyzing major factors that contribute to defining the fingerprints and thus unveiling interesting insights for understanding the problem.

2 Related Work

The interesting idea of GAN was first proposed in ([13]), where a generator G and a discriminator D are trained simultaneously in an adversarial manner. Many variants ([7, 17]) and relevant tasks ([12, 29]) have been explored since then. Recently, detecting GAN-generated imagery has been researched extensively ([24, 25]). For example, properties of facial landmarks are utilized for fake face detection ([20, 24, 25]). Deep networks ([22, 31]) are also being employed for this task. Though they yield high accuracy for the GANs they have been trained on, the performance may decrease sharply if tested on images from different GAN architectures. Other relevant efforts include those attempting to associate an image to one of the given GAN models. For example, ([3]) proposed an extended attribution problem using generator inversion in a white-box setting for identifying the source GAN model that was used to create the synthetic image.

Recent literature also reported that GANs produce distinct artifacts in the generated images, which is why most detection methods fail to generalize over a large number of GAN models. This drove researchers in the direction of finding if GAN-generated images have unique digital fingerprints just like real images do due to the imaging process. [23] used fingerprint extraction similar to PRNU (photo-response non-uniformity) pattern and revealed that every GAN architecture leaves some peculiar fingerprints in the generated images. However, the approach suggests that different random initializations could lead to different fingerprints, making it questionable to link a fingerprint uniquely to an architecture.

In [35], a method was proposed to train an external classifier on top of a GAN to extract

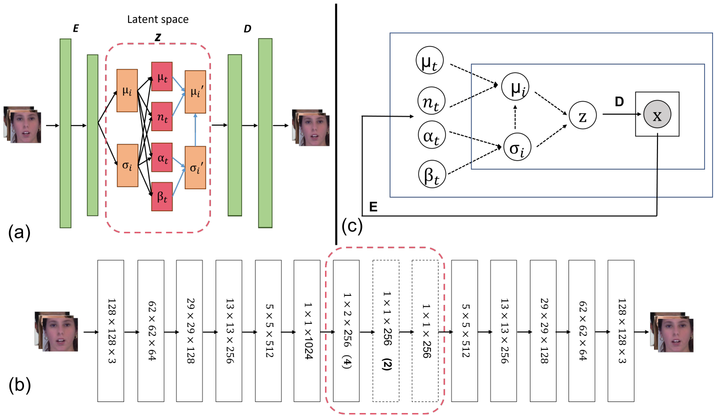


Figure 1: (a) Overview of BFR-VAE. The red dashed line block z represents the latent space. The two latent layers are represented by orange and red in z. (b) The size of output feature maps for each layer with () showing the number of duplicate blocks. The sampling process is shown as the black dashed line blocks. The input is the GAN images. (c) Bayesian view model structure.

the fingerprint that is supposedly unique to the underlying GAN. This approach also attempts decouple the GAN fingerprint from the image fingerprint. Another more recent work [33] reported an interpretable GAN fingerprint using yet another GAN-like framework. Another parallel line of research is finding the causes of artifacts in the generated images and fixing them like in [18]. Small changes to the network architecture are made to eliminate aliasing artifacts which produced high visual quality images, equivariant to sub-pixel translations like rotation etc. It remains to be explored whether such aliasing artifacts may be linked model-defining properties of a GAN model, if they may be eliminated.

While having some success in distinguishing real versus GAN-generated images, these works fall short of conclusively answering the question of what defines GAN fingerprints. For example, if the fingerprint-attributing model is explicitly based on an external classifier, it casts some doubt on whether the extracted fingerprint is indeed inherent to the underlying GAN. Also, if the uniqueness of the fingerprint depends on unrealistic assumptions like knowing the initialization in GAN training, the fingerprint would have little practical use.

Our approach aims at overcoming the common limitations of the existing methods. By employing an end-to-end learning approach with embedded hierarchical Bayesian modeling in a latent space of the fingerprints, we intend to support not only indexing a GAN (in relation to existing ones) but also modeling distinct new instances of the same GAN.

3 Learning to Extract GAN Fingerprints

To facilitate the discussion and presentation of our research on answering the questions related to GAN fingerprints as articulated previously, we pose the following clearly defined task: Given several groups of GAN-generated images, with each group being from a distinct GAN model/architecture, to extract a representation that is unique to each group. If the task is feasible and such a representation exists, it is called the fingerprint of the underlying GAN.

As mentioned above, our key idea is to model the (potential) GAN fingerprint using a hierarchical Bayesian approach in a latent space, allowing distribution-based abstraction above image-level features. This allows the probabilistic association of an image sample to each distribution (fingerprint), i.e., achieving soft membership assignment, which helps avoid overfitting under limited training samples. Purely from a Bayesian modeling point of view, this idea would be akin to topic modeling that can be solved by well-established approaches like Latent Dirichlet Allocation (LDA) ([6]). However, the feature space for the fingerprints needs to be first defined and then representations in such a space can be extracted, before we can apply such an approach, which essentially goes back to the original problem of extracting and defining fingerprints, making it not a real solution. To this end, we propose a learning-based approach whereby the hierarchical Bayesian modeling is defined in the latent space and embedded in a deep model so that both the representation and the hierarchical modeling of the representation are learned in an end-to-end fashion.

3.1 Bayesian Fingerprint-Reading VAE Overview

We term our approach Bayesian Fingerprint-Reading Variational AutoEncoder (BFR-VAE), whose structure is schematically illustrated in Figure 1. Through explicit hierarchical Bayesian modeling of the (potential) fingerprint in a latent space, BFR-VAE is different from other recent VAE variants like NVAE [32], FactorVAE [19] or TCVAE [8], in that the key objective of these techniques is still for better generative performance. Empirical studies also show a vanilla VAE lacks the capacity for GAN attribution (more details in the supplemental).

BFR-VAE has the following benefits: 1) It is an end-to-end learning framework and thus avoiding the difficult task of defining fingerprint features; 2) With a hierarchical Bayesian structure, the higher-level abstraction of the potential fingerprint features is more likely to capture model-specific information; 3) Besides distinguishing different fingerprints, it can identify potential correlations among them (in terms of distributions).

The following notations will be used in the subsequent discussion: the given GAN image is x; the potential GAN fingerprint is represented by (the parameters of) a Gaussian distribution $N(\mu_i, \sigma_i)$, and the hyper-parameters that govern the GAN fingerprints are represented by a Gamma distribution $Ga(\alpha_t, \beta_t)$ and a Gaussian distribution $N(\mu_t, n_t)$.

3.2 Learning in BFR-VAE

To better understand the process of extracting the fingerprint, we now look at the BFR-VAE from a Bayesian point of view (Figure 1(c)). Recall that for a typical topic-modelling model that is used for analyzing a collection of documents, two hierarchical distributions are employed. The higher-level distribution is a prior that determines the topic proportion whereas the lower-level distribution is a document-word distribution that decides the words appearing in the document. When learning the model, the prior (higher-level distribution) together with the likelihood (lower-level distribution) leads to the posterior which is the observation (the words). Then the posterior will become the prior for the updates in the next iteration. Moreover, the conjugate distributions used in the two levels simplify the iterative updates.

Analogous to topic modelling, we treat the fingerprint as the lower-level distribution which regulates the features for recovering the images. The potential components that govern the fingerprint are considered as the higher-level distribution. By the same token, conjugate distributions are used.

The latent space z and the decoder D can be viewed as the generating process of a sample/image x', namely $p(x'|z)$: First, we draw a sample (μ_i', σ_i') for GAN fingerprint from the prior distributions $Ga(\alpha_t, \beta_t)$ and $N(\mu_t, n_t)$, which are learned from the encoder E. Then, a sample drawn from $N(\mu_i', \sigma_i')$ forms the input e_{latent} for the decoder. Lastly, the decoder constructs the new sample x'. The process can be expressed as:

$$p(x'|z) = p(D(e_{latent}|z)) = D(\int N(e_{latent}|\mu_i', \sigma_i')N(\mu_i'|\mu_t, n_t\sigma_i')Ga(\sigma_i'|\alpha_t, \beta_t)\mathrm{d}\mu_i'\mathrm{d}\sigma_i') \quad (1)$$

Considering the input x and the latent space z which generates x, we would like to compute $p(z|x) = p(x|z)p(z)/p(x)$ to infer the characteristics of z. However, directly computing $p(x)$ is intractable. Thus we use an approximate posterior probability $q(z|x)$, namely, the encoder E in BFR-VAE, as a surrogate for $p(z|x)$. $q(z|x)$ can be expressed as:

$$q(z|x) = q(\mu_i, \sigma_i|x) \propto \int Ga(\sigma_i|E(x))N(\mu_i|\sigma_i, E(x))\mathrm{d}x \quad (2)$$

To make $q(z|x)$ similar to $p(z|x)$, the KL divergence is applied and our goal is to minimize the following objective function L_e:

$$\begin{aligned} L_e = KL(q(z|x))||p(z|x)) &\propto -(E_{q(z|x)}logp(z|x) - KL(q(z|x)||p(x))) \\ &\propto MSE(x, x') + KL(q(z|x)||p(x)) \end{aligned} \quad (3)$$

Thus, the learning process of BFR-VAE can be considered as inferring the sample x' from the given sample x with the goal to maximize $p(x'|x)$ since the corresponding samples/images all come from the same fingerprint distribution. Therefore, it is equivalent to maximizing $p(x'|z) + p(z|x)$, as $p(x'|z)$ is a constant once z is determined. Hence, only maximizing $p(z|x)$ is required and it eventually leads to minimizing Eq. (3).

We further impose a triplet loss ([30]) on the latent layer such that the samples from the same GAN have parameters (μ_i', σ_i') as close as possible and as distant as possible for samples from different GANs. Consider 3 adversarial samples x_1, x_2, x_3 where the first two belong to the same GAN and the last one is from other GANs. Then, the loss will be:

$$L_t = \sum_{p}^{p \in \{\mu_i', \sigma_i'\}} ||p_1 - p_2||_2^2 - ||p_1 - p_3||_2^2 \quad (4)$$

where $p_{1,2,3}$ represents the parameters in layer (μ', σ') for sample $x_{1,2,3}$.

Accordingly, BFR-VAE is trained end-to-end with the loss $L = \lambda_1 L_e + \lambda_2 L_t$ to be minimized ($\lambda_{1,2}$ are the weights for each loss component). With GAN images as inputs, the potential GAN fingerprint is represented by samples drawn from layer $Ga(\alpha_t, \beta_t)$ and $N(\mu_t, n_t)$. We also study the importance and effect of each loss component on fingerprint extraction. Please refer to supplemental for ablation study results.

4 Simulation-Based Experiments

As GAN fingerprints (if they exist) from real data may be too subtle to be visually inspected, we first evaluate our method by using a simple (i.e., assuming additive fingerprints) but illustrative simulation, where we explicitly introduce two-dimensional sine waves as the underlying "fingerprints". We use $3,000$ celebA ([21]) images with a resolution of 128×128 as the base images and evenly split them into 3 sets. Each set has 800 and 200 training and test images, respectively. Figure 2 shows the fingerprints and a few simulated images. Also, a Gaussian noise with L_2 norm being 0.1 is added to the fingerprints before adding them to

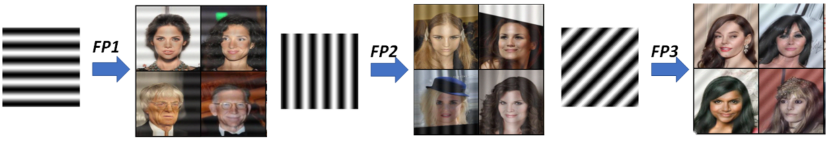
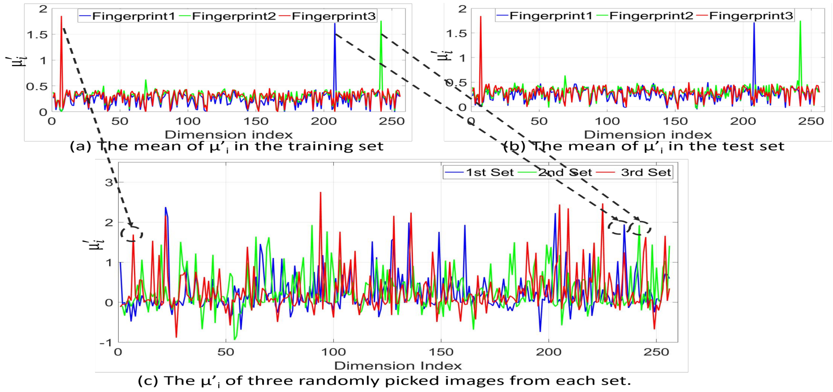


Figure 2: The additive "fingerprints" are sine waves with different orientations. The fingerprints with a L_2 norm being 0.2 are added to the celebA images.



(a) The mean of μ'ᵢ in the training set

(b) The mean of μ'ᵢ in the test set

(c) The μ'ᵢ of three randomly picked images from each set.

Figure 3: Visualizing the GAN fingerprint extracted in the simulation. Blue, green and red curves represent the first, the second and the third sets, respectively.

each image, which acts as data augmentation and simulate fingerprint variations. BFR-VAE is trained on the training set, where the optimizer is Adam with a learning rate of $6e-4$.

Next, we extract the fingerprint (μ_i', σ_i') (two $256 \times 1 \times 1$ tensors) for both the training and test data. Without loss of generality, we take μ_i' for illustration. First, the mean of μ_i' is calculated. As shown in Figure 3 (a) and (b), it is clear that, statically, each set has a unique pattern (represented by the peaks in each curve) and the pattern is shared among the training and test images. Note that, when visualizing the latent representation of any single image (Figure 3 (c)), it would be unclear to tell which part of the μ_i' vector has a dominant contribution to the underlying fingerprint. This is the nature of a distribution-based representation (but the likelihood of any single sample may still be assessed under a learned distribution). To further verify if the peak corresponds to the fingerprint, we modify the value of μ_i' of a single image in two ways: 1) removing the peak by setting the value of the peak index of the mean curve (marked in black dashed line) to the average of all other indexes; and 2) maintaining the peak but altering other values by replacing the rest 255 dimensions by the average. As such, both methods lead to a new vector, then a sample drawn from it is used as the input for the decoder to generate new images.

Figure 4 depicts the resulted images. In each box (different colors correspond to the curves in Figure 3) , the left image (Original) is generated from the unmodified latent vector whereas the second (Peak2Mean) and third (Peak&Mean) ones are the images generated from the vectors modified using the two methods mentioned above, respectively. Compared with the Original, the Peak2Mean image eliminates the underlying "fingerprint" to a large extent (e.g., the background does not show obvious periodic waves) whereas the Peak&Mean image basically maintains the fingerprint. It is worth mentioning that, in both cases, the images are slightly distorted and blurred, which indicates that each dimension of the μ_i vector

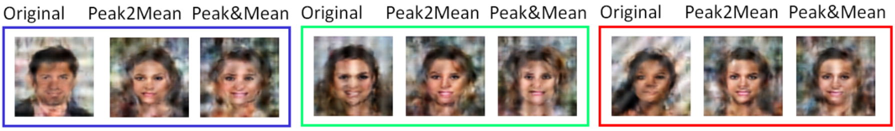
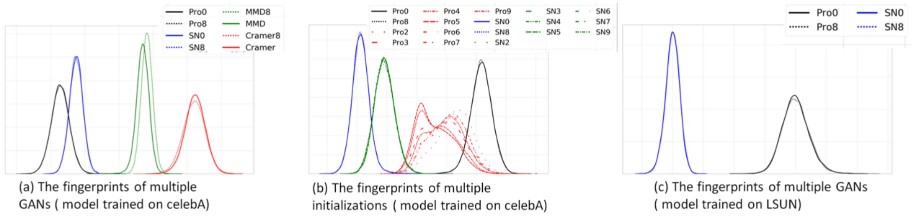


Figure 4: The images generated from the modified "fingerprints".



(a) The fingerprints of multiple GANs (model trained on celebA)

(b) The fingerprints of multiple initializations (model trained on celebA)

(c) The fingerprints of multiple GANs (model trained on LSUN)

Figure 5: Visualizing the distribution of the extracted fingerprint in 1-D for different GANs. Same color denotes the same GAN while different line patterns represent different initializations of a GAN. (a) BFR-VAE trained and tested on 4 GANs, each with 2 initializations (celebA) (b) BFR-VAE trained on 2 GANs, each 2 initializations and tested on images generated by 7 unseen initializations of each GAN. (c) BFR-VAE trained on 2 GANs with 2 initializations (LSUN).

contains some image information (our model does not attempt to disentangle the "fingerprint" from the image content). Nevertheless, these simulation experiments show that the proposed BFR-VAE model can indeed identify the underlying fingerprint efficiently.

5 Experiments with Real Datasets

In this section, we evaluate BFR-VAE using real datasets under various settings. First, we describe the experimental setup followed by the experiment results that evaluate our model for extracting reliable fingerprints under different configurations. These experiments also show how different factors affect the GAN fingerprints. We also include comparison with baselines and experiments regarding single images in the supplemental.

5.1 Experimental Setup

We describe the GAN models, datasets and evaluation criteria used for our experiments here.

GANs used: We use ProGAN [15], Spectral Normalization GAN (SNGAN) [26], Maximum Mean Discrepancy GAN (MMDGAN) [5], CramerGAN [4], StyleGAN [16] and StarGAN [9] to evaluate our approach because 1) they are state-of-the-art networks; 2) each of them represents difference in either network architecture or their loss functions. To present the results, we denote different GAN models and initialization seeds as *GANx*, where *GAN* is the type of GAN and *x* is the initialization seed used.

Datasets: We use CelebA ([21]) and LSUN ([34]) bedroom scene dataset, since they are widely used for GAN-related benchmarks and significantly different from each other. For CelebA, we use a total of 182,637 generated images (162,770 for training and the rest for testing). For LSUN, 200,000 and 20,000 images are used in training and testing, respectively. We follow the experiment configurations described in [35], including the image resolution, GAN training protocol, etc. To show that the GAN imagery used have a good quality, we present the Fréchet Inception Distance (FID) score for the generated images in the supplemental.

GAN model	Pro0	Pro8	SN0	SN8
Pro0	0 / 1e02	**8.92e-02 / 1e2**	27.52 / 22.37	27.55 / 22.36
Pro8	**8.92e-02 / 1e02**	0 / 1e2	27.39 / 22.56	27.38 / 22.55
SN0	27.52 / 22.37	27.39 / 22.56	0 / 1e02	**3.70e-02 / 1e02**
SN8	27.55 / 22.36	27.38 / 22.55	**3.70e-02 / 1e02**	0 / 1e02
MMD0	30.37 / 20.56	30.38 / 20.60	31.01 / 20.41	31.05 / 20.43
MMD8	32.76 / 16.37	32.72 / 16.40	33.02 / 16.40	33.08 / 16.42
Cramer0	34.62 / 14.90	34.60 / 14.96	34.22 / 15.52	34.33 / 15.45
Cramer8	34.95 / 14.97	34.93 / 15.04	34.64 / 15.54	34.74 / 15.47
Pro4	8.02 / 80.18	**8.01 / 80.22**	27.52 / 30.52	27.56 / 30.51
SN4	28.92 / 27.39	28.79 / 27.57	**7.03 / 86.61**	7.08 / 86.58
Style	17.42 / 51.87	17.35 / 52.04	10.51 / 74.42	10.59 / 74.49
Star	21.20 / 42.25	21.07 / 42.41	10.42 / 74.38	10.46 / 74.37
GAN model	**MMD0**	**MMD8**	**Cramer0**	**Cramer8**
Pro0	30.37 / 20.56	32.76 / 16.37	34.62 / 14.90	34.95 / 14.97
Pro8	30.38 / 20.60	32.72 / 16.40	34.60 / 14.96	34.93 / 15.04
SN0	31.01 / 20.41	33.02 / 16.40	34.22 / 15.52	34.64 / 15.54
SN8	31.05 / 20.43	33.08 / 16.42	34.33 / 15.54	34.74 / 15.47
MMD0	0 / 1e02	**0.59 / 98.97**	27.54 / 23.19	27.80 / 23.08
MMD8	**0.59 / 98.97**	0 / 1e02	29.56 / 18.90	29.29 / 20.00
Cramer0	27.54 / 23.19	29.56 / 18.90	0 / 1e02	**0.38 / 99.30**
Cramer8	27.80 / 23.08	29.29 / 20.00	**0.38 / 99.30**	0 / 1e02
Pro4	14.68 / 69.69	16.77 / 65.41	20.86 / 45.24	21.15 / 45.64
SN4	23.00 / 41.97	25.07 / 36.77	16.15 / 58.55	16.57 / 58.70
Style	17.80 / 62.93	19.80 / 58.52	19.76 / 52.39	19.96 / 53.35
Star	16.82 / 64.36	18.71 / 60.31	18.13 / 55.21	18.32 / 56.41

Table 1: $JSD/correlation$ between extracted fingerprints for images generated using different GANs. BFR-VAE is trained on celebA images generated by ProGANx (Prox), SNGANx (SNx) MMDGANx (MMDx) and CramerGANx (Cramerx) where x=$\{0,8\}$ is the initialization seed used for training and x = 4 is the unseen initialization seed. *Note all the values have 1e-02 as a multiplication factor. The most similar fingerprints (excluding itself) are in **bold** and the closest fingerprints across different GANs are underscored.

Evaluation criteria: We use both Jensen Shannon-Divergence (JSD) and correlation coefficient between two distributions (representing two fingerprints) as the evaluation criteria. Both JSD and correlation coefficient lie in the range of 0 to 1. For JSD, 0 and 1 represent the same and disparate distributions respectively. On the other hand, for correlation coefficient, 0 and 1 represent two independent and correlated distributions respectively. As an illustration, we also provide Figure 5 which contains visualization of the fingerprint distributions in 1-D, in which the dimensions are reduced using principal component analysis (PCA).

5.2 Experimental Results

We analyze the extracted GAN fingerprints under different conditions like different GAN models, initializations, datasets and image transformations and discuss the impact of these factors on the fingerprint.

Different GANs: We train BFR-VAE using GAN celebA images generated from four GAN models (i.e., ProGAN, SNGAN, MMDGAN, and CramerGAN) and consider two different initializations for each GAN. Figure 5 (a) illustrates the GAN fingerprints in 1-D. We observe that different GANs have distinct distributions that are almost independent of initializations. This indicates that each GAN has its own unique "fingerprint" and the initialization has minimum impact. Moreover, we calculate the JSD and correlation coefficient between GANs, as shown in Table 1. We notice that the same GAN with different initializations results in less JSD value and higher correlation coefficient than the case of different GANs, which suggests the same conclusion.

Another test was conducted on two sets of CelebA data generated from ProGAN and SNGAN with initializations that were never used in training. The results are presented in Table 1. The unseen ProGAN initialization shows higher correlation to seen ProGAN than other GANs. Similar trend is observed for unseen SNGAN as well.

Furthermore, we explore whether BFR-VAE can identify unseen GANs with fingerprints extracted from unseen GANs (i.e. StyleGAN and StarGAN). Table 1 depicts low similarity between unseen GANs and seen GANs, which indicates BFR-VAE can distinguish new/unseen GANs.

	Pro0	**Pro8**	**SN0**	**SN8**
Pro2	4.54 / 91.41	**4.54 / 91.44**	27.42 / 29.03	27.51 / 29.02
Pro3	**9.24** / 77.34	9.27 / **77.37**	28.44 / 29.36	28.49 / 29.34
Pro4	8.02 / 80.18	**8.01 / 80.22**	27.52 / 30.52	27.56 / 30.51
Pro5	13.70 / 61.16	**13.69 / 61.20**	29.26 / 27.94	29.30 / 27.92
Pro6	6.40 / 85.19	**6.39 / 85.22**	27.40 / 30.66	27.46 / 30.64
Pro7	**6.79** / 85.26	6.81 / **85.29**	27.68 / 30.15	27.71 / 30.13
Pro9	12.40 / 65.66	**12.40 / 65.69**	28.80 / 28.41	28.85 / 28.40
SN2	28.20 / 28.80	28.05 / 28.99	**6.20 / 88.26**	6.26 / 88.23
SN3	27.91 / 29.10	27.76 / 29.29	**6.03 / 88.58**	6.10 / 88.56
SN4	28.92 / 27.39	28.79 / 27.57	**7.03 / 86.61**	7.08 / 86.58
SN5	27.99 / 29.12	27.85 / 29.31	**6.09 / 88.45**	6.11 / 88.42
SN6	28.39 / 28.38	28.21 / 28.57	**6.50 / 87.72**	6.57 / 87.70
SN7	28.50 / 28.46	28.35 / 28.64	6.57 / **87.84**	**6.49** / 87.81
SN9	28.49 / 28.41	28.32 / 28.60	**6.50 / 87.83**	6.57 / 87.80

Table 2: $JSD/correlation$ between extracted fingerprints for images generated using different GANs. BFR-VAE is trained with celebA images generated by ProGANx (Prox) and SNGANx (SNx) where x=$\{0,8\}$ is the initialization seed used for training and x=$\{2,3,4,5,6,7,9\}$ is the unseen initialization seed. *Note all the values have 1e-02 as a multiplication factor. The most similar fingerprints (excluding itself) are in **bold** and the closest fingerprints across different GANs are underscored.

Different Initializations: The above experiments mainly focus on seen data. To further investigate the impact of unseen initializations, we train BFR-VAE using ProGAN and SNGAN with only 2 initializations and test on other 7 different initializations. The solid/dashed black and blue lines in Figure 5 (b) represent the ProGAN and SNGAN initializations used in the training respectively whereas the red and green represent the unseen initializations for ProGAN and SNGAN, respectively. We notice that 1) the initializations used in training are almost overlapping with each other, and 2) the unseen initializations are closer to the ones from the same GAN model (i.e., red and green curves are close to black and blue ones, respectively). The JSD value and correlation coefficient are presented in Table 2. It shows high similarity exists for the same GAN and low similarity for different GANs.

Different Datasets: Here, we show how BFR-VAE performs on other dataset, namely, LSUN. We follow the similar training protocol as for celebA - train using images from ProGAN and SNGAN with two different initializations. Figure 5 (c) depicts the distributions

	Pro	**SN**	**MMD**	**Cramer**
Pro(noise)	**5e-2 / 99.98**	27.35 / 22.83	30.58 / 19.49	34.15 / 15.63
SN(noise)	27.09 / 23.12	**5e-2 / 99.99**	31.29 / 19.30	33.90 / 16.13
MMD(noise)	30.84 / 19.39	31.81 / 18.58	**0.10 / 99.98**	27.97 / 22.00
Cramer(noise)	34.26 / 15.72	34.16 / 15.87	28.20 / 22.09	**5e-2 / 99.99**
Pro(blur)	14.85 / 60.92	14.89 / 61.07	**14.63 / 60.41**	15.32 /58.50
SN(blur)	14.17 / 62.69	**13.69 / 64.63**	13.84 / 62.50	14.54 / 60.53
MMD(blur)	19.23 / 60.06	19.42 / 60.08	**17.27 / 64.41**	17.58 / 62.44
Cramer(blur)	16.79 / 64.03	17.40 / 62.30	17.87 / 58.39	**16.10 /63.60**
Pro(JPEG)	**11.78 / 70.92**	17.52 / 51.85	14.83 / 58.77	15.52 / 57.90
SN(JPEG)	16.63 / 54.48	**9.27 / 81.24**	16.01 / 55.21	16.62 / 54.60
MMD(JPEG)	19.82 / 56.75	21.78 / 51.26	1**4.82 / 71.02**	17.50 / 62.35
Cramer(JPEG)	17.07 / 62.36	19.60 / 54.68	17.38 / 57.81	**14.40 / 68.80**

Table 3: $JSD/correlation$ between extracted fingerprints for GAN images with different types of transformations. The image transformation applied is indicated in (). Largely speaking, the images from the same GAN have lower JSD and higher correlation than those belonging to different GANs. *Note all the values have 1e-02 as a multiplication factor. The most similar fingerprints are **highlighted**.

of the fingerprints for this setting, which are distinguishable. However, if we use the trained BFR-VAE to test on an unseen dataset (e.g., trained on LSUN and test on celebA or vice versa), it cannot discriminate different GAN models. This implies the datasets can significantly affect the fingerprint.

Different Image Transformations: To investigate whether the fingerprints extracted by BFR-VAE are robust to image transformations, we apply standard Gaussian noise with a L_2 norm of 0.1, Gaussian blur with a kernel size of 5 and JPEG compression with a quality factor of 75 to celebA images respectively, and evaluate the extracted fingerprints with distorted images as input. The BFR-VAE model used is trained with the original images generated by 4 GANs of 2 initialization seeds. The average performance of both initialization seeds is reported in Table 3, although images transformation indeed deviate the fingerprints from the original ones, the fingerprints of the same GAN exhibit stronger correlation and less JSD than that of different GANs. This indicates that the fingerprint extracted by BFR-VAE is robust to common image transformations.

6 Conclusion

We presented a novel architecture BFR-VAE, which embeds hierarchical Bayesian modeling into a VAE, for modeling and extracting GAN-specific fingerprints. With various state-of-the-art GAN models, we evaluated our approach under realistic settings with both simulation and real datasets to demonstrate its effectiveness. Through our experiments, we may conclude that the GANs indeed appear to leave unique fingerprints that are independent of the initialization (as opposed to what reported in the literature). Also, we found that the GAN fingerprints extracted by our model are dependent on the dataset used in training. Using our model, we also analyzed several key factors defining the GAN fingerprints.

Acknowledgement: The work was supported in part by a grant from ONR. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR.

References

[1] Deepfake detection challenge. 2020. https://www.kaggle.com/c/deepfake-detection-challenge.

[2] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 660–661, 2020.

[3] Michael Albright and Scott McCloskey. Source generator attribution via inversion. In *CVPR Workshops*, volume 7, 2019.

[4] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[5] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[8] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[10] CNN. 2019. https://www.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes.

[11] CNN. 2019. https://www.cnn.com/2019/02/18/tech/dangerous-ai-text-generator/index.html.

[12] Yuzhen Ding, Nupur Thakur, and Baoxin Li. Advfoolgen: Creating persistent troubles for deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 142–151, 2021.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[14] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing convolutional traces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 666–667, 2020.

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021.

[19] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[20] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*, 2018.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[22] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 384–389. IEEE, 2018.

[23] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511. IEEE, 2019.

[24] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92. IEEE, 2019.

[25] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018.

[26] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[27] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 43–47, 2018.

[28] NewYorker. 2018. https://www.newyorker.com/magazine/2018/11/12/in-the-age-of-ai-is-seeing-still-believing.

[29] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580, 2018.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[31] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. Detecting both machine and human created fake face images in the wild. In *Proceedings of the 2nd international workshop on multimedia privacy and security*, pages 81–87, 2018.

[32] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[33] Tianyun Yang, Juan Cao, Qiang Sheng, Lei Li, Jiaqi Ji, Xirong Li, and Sheng Tang. Learning to disentangle gan fingerprint for fake image attribution. *arXiv preprint arXiv:2106.08749*, 2021.

[34] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[35] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7556–7566, 2019.

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.