Detecting Overfitting of Deep Generative Networks via Latent Recovery

Ryan Webster, Julien Rabin, Loïc Simon and Frédéric Jurie
Normandie Univ., ENSICAEN, UNICAEN, CNRS, GREYC, France
ryan.webster@unicaen.fr

Abstract

State of the art deep generative networks have achieved such realism that they can be suspected of memorizing training images. It is why it is not uncommon to include visualizations of training set nearest neighbors, to suggest generated images are not simply memorized. We argue this is not sufficient and motivates studying overfitting of deep generators with more scrutiny. We address this question by i) showing how simple losses are highly effective at reconstructing images for deep generators ii) analyzing the statistics of reconstruction errors for training versus validation images. Using this methodology, we show that pure GAN models appear to generalize well, in contrast with those using hybrid adversarial losses, which are amongst the most widely applied generative methods. We also show that standard GAN evaluation metrics fail to capture memorization for some deep generators. Finally, we note the ramifications of memorization on data privacy. Considering the already widespread application of generative networks, we provide a step in the right direction towards the important yet incomplete picture of generative overfitting.

1. Introduction and Related Work

In just a few short years, image generation with deep networks has gone from niche to a center piece of machine learning. This was largely initiated by Generative Adversarial Networks (GANs) [13] and since then incredible progress has been made, from deep convolutional GAN (DCGAN) [32] producing artifacted faces, to progressive GANS (PGGAN) [20] producing faces which are virtually indistinguishable from real ones even to human observers and at high resolution (see Fig. 1). While a large amount of research has proposed new generative models, less research has been devoted to the evaluation of such models. Furthermore, evaluating overfitting of deep generators has been performed via intuitive visual demonstrations, such as training set nearest neighbor search and latent space interpolation [7, 20]. Fig. 1 (last column) illustrates the nearest neighbor (NN) test, where $\text{NN}_{\mathcal{D}}(y)$ is the training dataset

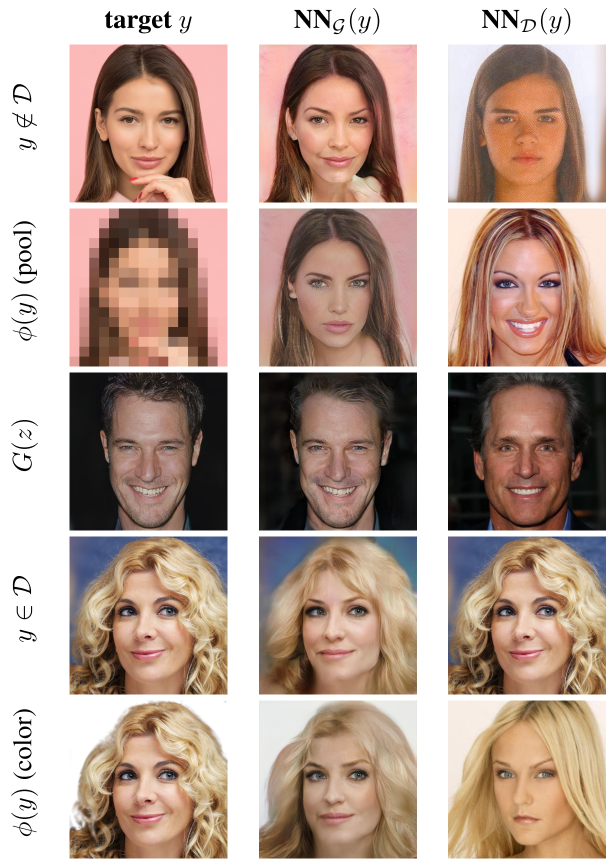


Figure 1: *Rather than inspecting the most similar images $NN_{\mathcal{D}}(y)$ in the training dataset $\mathcal{D}$ for sampled generated images $G(z)$ (row 3), we consider finding the most similar image in the manifold $NN_{\mathcal{G}}(y)$ of generated images (column 2). As seen in the last two rows, $NN_{\mathcal{G}}(y)$ is more meaningful under some transformations. Analysis of the discrepancy between reconstructions of the train set $\mathcal{D}$ and reconstructions outside $\mathcal{D}$ makes it possible to detect overfitting for some generators.*

NN of a few images y. While $\text{NN}_{\mathcal{D}}(y)$ with the Euclidean distance is a common heuristic (last column in Fig. 1), it is purely visual and sensitive to transformations of the training data.

In contrast, we suggest to rely on the opposite methodology by optimizing the latent code $z \sim \mathcal{Z}$ to find the nearest neighbors $\mathrm{NN}_{\mathcal{G}}(y)$ in the manifold of generated faces $\mathcal{G} = \{G(z)\}_{z \sim Z}$ of images from the training ($y \in \mathcal{D}$) and a validation set ($y \notin \mathcal{D}$). Not only is this approach more robust, it provides us with reconstructions errors which can be analyzed for different sets of images. Using this framework that we refer to as *latent recovery*, we propose the following contributions:

- A demonstration of successful latent recovery across a variety of generators. In Section 2 we introduce our optimization procedure and show it is meaningful even if the target image is corrupted.
- Section 3 introduces a novel method to numerically estimate overfitting in deep generators via statistics of recovery errors on test and train sets. Overfitting is undetectable for GANs, which is corroborated visually in Fig. 3 and statistically in Table 1. Overfitting is however detectable in hybrid adversarial losses similar to CycleGAN [42], and easily detectable in non-adversarial generators such as GLO [4]. Finally, we show that standard evaluation metrics do not detect overfitting in some models.

1.1. Related Work

Adversarial losses have seen successful applications in a variety of settings beyond just image generation: unpaired image to image translation in CycleGAN [42], face attribute modification in StarGAN [8] and various image inpainting techniques [17, 39] to name a few. This progress has created a huge need to evaluate generated image quality, which to some degree has not been fully answered [6].

GAN Evaluation Metrics The Fréchet Inception Distance (FID), recently introduced in [16], has become a standard for evaluating the quality of generated GAN images. The FID is computed by computing the Fréchet Distance between features of the Inception network [34] modeled as multivariate gaussians. Furthermore, it was demonstrated to be consistent with human evaluation. In the large scale GAN study [24], FID was used to compare a huge variety of GANs, wherein it was shown auxiliary factors such as hyperparameter tuning can obfuscate true differences between GANs. In [33], notions of precision and recall are introduced for generated images, to help characterize model failure rather than providing a scalar in image quality. While these works have helped to compare GAN image quality, they do not address overfitting of the training set.

Overfitting in Generative Networks For image classification, a model is said to overfit when it performs significantly better on training examples compared to test examples, and said to generalize otherwise. While the exact reasons why deep nets generalize even when over parametrized is an open question, they are certainly not immune to overfitting. In the extreme case, Zhang et al. [40] demonstrated random labels can be perfectly memorized even on the large scale ImageNet database.

Despite this, very little work has gone into defining overfitting for generative models. In [1], the authors defined generalization for GANs in a largely theoretical setting. The formulation was used to suggest a new GAN training protocol rather than provide an evaluation technique. In [2], the support of a GAN generator, in terms of the number of face identities it could produce, was estimated using the birthday paradox heuristic. While crude, it suggested the support of faces could be quite large with respect to the size of the training set. The very recent work of [14] attempts to numerically estimate the notion of overfitting with a Neural Net Distance (NND). That is, they train a neural net to differentiate generated samples from real samples and similar to [18], use the resulting divergence as a measure of quality. Importantly, they are able to show a slight overfitting for some GANs and show that this divergence penalizes a generator trivially memorizing the train set. Unfortunately, this approach requires a massive test set in order to train the NND, which is unrealistic considering successful GANs already require massive train sets, not to mention the NND itself needs to be trained. Furthermore, the NND may favor GANs if the divergence they use resembles or is identical to the GAN under evaluation.

Finally, a new class of generative models has recently been proposed which involves invertible generators [10, 21]. These generators are attractive as they are mathematically well motivated and admit exact log-likelihood estimations, via taking the determinant of each layer jacobian. [28] examined the log likelihoods for such models and showed that while some GAN models generalized nicely to validation samples, out of distribution samples, such as those taken from completely different datasets, yield higher likelihoods. [37] also examined log-likelihood in the generative setting, and both works ultimately cautioned against the use of log-likelihood for generative evaluation.

Memorization and Privacy Beyond these aspects of memorization and practical evaluation of generators lies the important and debated issue of privacy: How to ensure that the data used for training cannot leak by some reverse engineering, such as reconstruction from features [25, 12] ? Because GANs have seen such widespread application, it is imperative that we have better evaluation tools to assess how much these networks have overfit the training data. For example, if a user is using a neural net to inpaint faces as in [22] or to perform super-resolution enhancing [9], it seems necessary to ensure verbatim copies of training images do not appear, due to privacy or even copyright concerns. Indeed, several attacks against machine learning systems have been exposed in the literature [30]. For instance, authors in [12] designed an inversion attack to coarsely recover faces used during the training of a white box facial recognition neural network. More recently, [35] performed a successful membership attack, which is the ability to discern training examples from a model, in a purely black box setting. Very recently [15] explored the potential of membership attacks for GANs and exploited the tendency of the discriminator to overfit the training set.

2. Reconstruction by Latent Code Recovery

This section proposes a methodology for reconstructing the most similar images to target images with an existing generator. Inversion of deep representations has been already addressed in the literature. [25] used a simple optimization procedure to maximize an output class of a VGG-like network. In the seminal works of [29, 36], a similar inversion of deep nets unveiled adversarial examples. In [24], *generative* networks are inverted to study recall, which is the ability of the network to reproduce all images in the dataset and finally [27] used latent recovery of a GAN generator to evaluate its quality.

Other works tackle recovering latent codes directly by training an encoder network to send images back from image space to latent space, such as the BEGAN model [3] or Adversarially Learned Inference (AGI) [11]. In Generative Latent Optimization (GLO) [4], a generative model is trained along with a fixed-size set of latent codes, so that they are known explicitly when training finishes.

In this paper, we will proceed by recovering latent codes via optimization, following [27, 5, 23, 24]. In contrast with [27], we will ultimately be concerned comparing image recovery between train and validation sets.

2.1. Latent Code Recovery with Euclidean Loss

We explore recovery with a euclidean loss and find it is effective at recovering latent codes for a variety of GAN methods. Here, we consider the following *latent recovery* optimization problem

$$z^\star(y) \in \arg\min_z \|\phi(G(z)) - \phi(y)\|_2^2 \qquad (\mathrm{NN}_\mathcal{G})$$

where G is a deep generative network, z is the input latent vector and y is the target image. Using a solution z^* of Problem ($\mathrm{NN}_\mathcal{G}$), we denote by $\mathrm{NN}_\mathcal{G}(y) = G(z^*)$ the Nearest Neighbor recovery of a given image y in the set of generated images, as opposed to the usual NN search in a dataset $\mathcal{D}$: $\mathrm{NN}_\mathcal{D}(y) = \arg\min_{x\in\mathcal{D}} \|x - y\|$. In this work, we consider mostly ϕ as the identity, but other operators are discussed in the next paragraph and for applications such as super resolution in Section 4. Fig. 1 illustrates the difference between the two NN searches on a few examples.

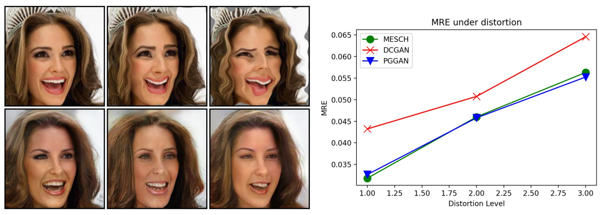
Experimental Validation In every experiment, we employ LBFGS and noted it converges roughly 10x faster than SGD (successful recovery requiring approximately 50 iterations as opposed to 500 in [5, 23]). Although Eq. ($\mathrm{NN}_\mathcal{G}$) is highly non-convex, the proposed latent recovery optimization works well, as shown in Fig. 1 and Fig. 3. In particular for generated images $y = G(z)$, where $\mathrm{NN}_\mathcal{G}(y) = z$, a global minimum (verbatim copy) is consistently achieved (see third row of Fig. 1). Every network analyzed in this document appeared to be able to verbatim recover generated images, an observation also noted by [23] and exemplified by the tight distribution of errors near zero in Fig. 4. Note that we also considered the widely used perceptual loss [19] by taking ϕ to be VGG-19 features, with either no improvement or even degradation of visual results (see supplementary). Furthermore, we did not see any difference in the statistical results of Section 3 for perceptual losses.

2.2. Latent Code Recovery Under Distortion

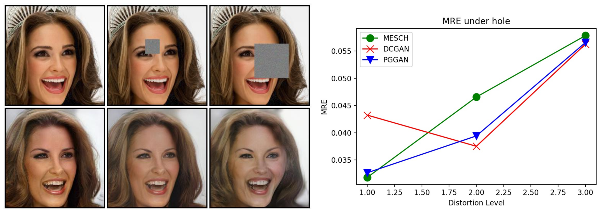
It should be noted that Eq. ($\mathrm{NN}_\mathcal{G}$) by itself may not be meaningful for some generators. For example, if the generator is invertible, errors will zero regardless of the target image. To verify that Eq. ($\mathrm{NN}_\mathcal{G}$) is meaningful, we want to make sure the error is lower for images inside the considered manifold and large for those outside. To do this, we follow [16] (used there to motivate the FID) wherein we test Eq. ($\mathrm{NN}_\mathcal{G}$) response to various distortions. We choose ϕ to be one of the three distortions that are illustrated in Fig. 2:

- **Smooth Vector Field Warp (Fig. 2a)** Following [16, 41] we warp training images by bilinear interpolation with a smooth 2D vector field $V_{\sigma_d} = V * g$, which is obtained from the Gaussian smoothing g of a Gaussian random vector field $V(x, y) \sim \mathcal{N}(0, \sigma_d^2)$;
- **Corruption Noise Patches (Fig. 2b)** As [22], we corrupt training images by replacing patches of various sizes with fixed Gaussian noise with variance σ_d^2;
- **Additive Noise (Fig. 2c)** We add noise to each training image with $X_n = X + W_d$, where W_d is sampled from a Gaussian distribution $W_d \sim \mathcal{N}(0, \sigma_d^2)$.

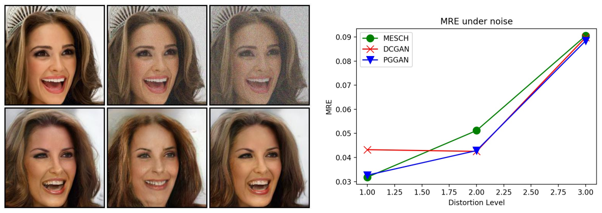
Experimental Validation Fig. 2 demonstrates a few facts about latent recovery. By inspection of recovered images, it appears robust enough to recover faces semantically similar to the ground truth even if the image has been heavily distorted. It also demonstrates the precision of the network, for example the three networks highlighted will reject images only slightly outside the manifold. In Table 1, we can see that not all networks share the same specificity. For example, the GLO networks can recover distorted images with similar MRE's to training images, which means the networks are less precise. This is coupled with a lower FID of the network, for example see Fig. 5.



(a) Deformation by smooth diffeomorphism (warping)



(b) Unsupervised inpainting (face completion)



(c) Additive white noise

Figure 2: *Median recovery error (MRE, see Eq. (4)) for 1800 test images on various GAN generators (PG-GAN [20], MESCH [26] and DCGAN [32]) under various distortions ϕ in latent recovery optimization ($\text{NN}_{\mathcal{G}}$) (see text for details).*

3. Using Latent Recovery to Assess Overfitting

In this section, we train a variety of generative models with a training and validation split. Then, we analyze the difference between image recovery using Eq. ($\text{NN}_{\mathcal{G}}$), between training and validation images.

3.1. Training Protocols

We summarize the details of each generative model below, in terms of their training procedure and purpose within this work.

GAN Generative Adversarial Networks (GAN) involve a stochastic training procedure which simultaneously trains a discriminator and a generator. The original GAN [13] optimization problem writes

$$\max_D \min_G \mathbb{E}_{z\sim\mathcal{N}(0,1), x\sim p_{data}}[\mathcal{L}_{adv}(D,G,z,x)] \quad (1)$$

where $\mathcal{L}_{adv}(D,G,z,x) = \log(D(x)) + \log(1-D(G(z)))$. We examine three prominent GANs in the literature. First is DCGAN [32], as it is one of the most widely used GAN architectures and with still decent performance across a variety of datasets [24]. Then we study two state-of-the-art GANs for high resolution generation; progressive growing of GANs [20], which we refer to as PGGAN and the zero centered gradient penalty Resnet presented in [26], which we refer to as MESCH. We train these three GANs on CelebA-HQ with a training split of the first 26k images and the first 70k images of LSUN bedroom and tower. We chose these splits to preserve the quality of each method, as GAN quality significantly degrades with small dataset sizes.

Generative Latent Optimization (GLO) The recently introduced Generative Latent Optimization (GLO) creates a mapping from a fixed set of latent vectors to training images. The GLO objective is as follows

$$\min_G \sum_{(z_i,x_i)} \mathcal{L}_{rec}(G(z_i), x_i) := \|G(z_i) - x_i\|_2^2 \quad (2)$$

where $x_i \in \mathcal{D}$ refers to training images, $z_i \sim \mathcal{N}(0,1)$ samples a Gaussian distribution and the pairs (z_i, x_i) are drawn once and for all before training begins[1]. Because we know the latent distribution is Gaussian, we can easily sample the network after it is trained.

AutoEncoder Finally, we train a vanilla autoencoder on CelebA-HQ with the objective:

$$\min_{G,E} \sum_{x_i\in\mathcal{D}} \mathcal{L}_{rec}(G, E, x_i) \quad (3)$$

Hybrid Losses We consider a generative model combining both the adversarial loss Eq. (1) with euclidean auto encoding loss (3) which we refer to as AEGAN.

Concerning models trained with a reconstruction loss (GLO, AEGAN and AE), we selected these architectures for a theoretical perspective, as they offer interesting windows into how generators can memorize. In particular, we will study the impact of the training set size N on the overfitting inclination. For example, while we were unable to train a good quality GAN with a small set of images (say 256), GLO converges extremely quickly in such a case. See the GLO-256 network in Fig. 3 (4th row) where memorization is immediately apparent. As a result, we will refer respectively to GLO-N, AEGAN-N and AE-N, to account for this size. Besides, for both the AE and AEGAN models, we forgo optimization in Eq. ($\text{NN}_{\mathcal{G}}$) and use the encoder E (3) to recover the latent vector, as is natural for autoencoder models.

In the next paragraphs, we will proceed to show that it is possible for generative networks to memorize in the sense that validation and training sets have significantly different recovery error distributions.

3.2. Comparison of recovery errors

Figure 4 shows the histograms of recovery errors on train ($\mathcal{D}$ in green) and validation ($\mathcal{T}$ in red) datasets from

[1] Contrary to [4], we do not optimize the latent space and found no negative impact on the reconstruction capacity and generation quality.

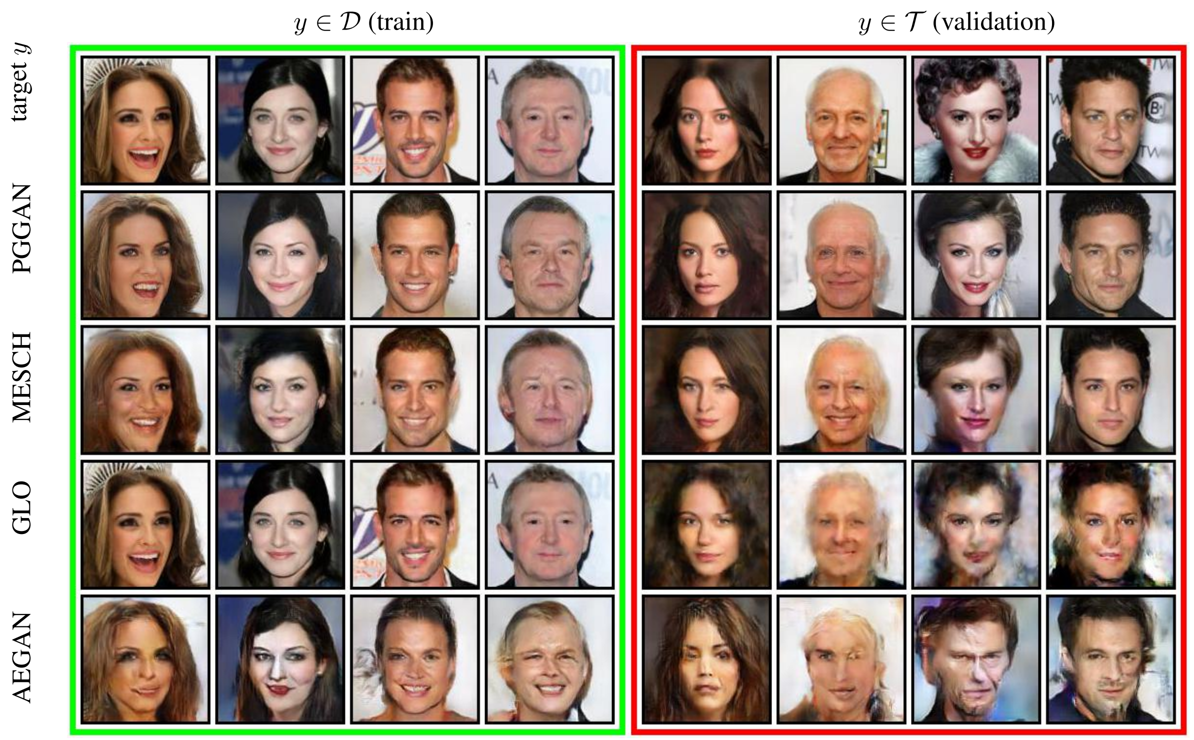


Figure 3: *Latent recovery of training images from $\mathcal{D}$ (left, green frame) and test images from $\mathcal{T}$ (right, red frame) for 128×128 images of Celeba-HQ [20]. From top row to bottom are first **target** images, and then recovery from Progressive GANs [20] (**PGGAN**), 0-GP resnet GAN [26] (**MESCH**), a **GLO** network [4], and finally a Cycle-GAN like network [42] (**AEGAN**). While GLO obviously shows some memorization of training examples, it is hard to visually assess when overfitting happens for other methods, as discussed in Section 3 (with additional details on architectures and training).*

CelebA-HQ, for various generators. For the sake of readability, the distribution of errors for generated images (yellow) from $\mathcal{G}$ and distorted images (blue) are only displayed for PGGAN and MESCH. Confirming visual inspection from Fig. 1, observe that the recovery errors for generated images (in yellow) are quite low. Increasing the number of iterations and using several random initializations improve results, but have not been used to reduce computation costs.

Now we are going to consider the distribution of recovery errors for test and train. For GLO-N and AEGAN-N generators with $N \in \{128, 1024, 8192\}$, the difference is clear, and is decreasing with the number of training images N. For very small datasets of $N = 128$, the train and validation error distributions are disjoint. On the other hand, pure GAN models can not be successfully trained with small datasets. We therefore only trained PGGAN and MESH with full datasets and in both cases, the difference of recovery error distribution between the train (green) and validation (red) set is barely noticeable. Further statistical analysis in the next paragraph shows indeed that such a small gap is very likely for two samples drawn from the same law, demonstrating generalization.

3.3. Statistical Analysis

In light of the previous results, we propose two simple definitions to measure and detect overfitting without relying on histograms or image inspection. First, to summarize the distribution of errors to a single value, we consider the **Median Recovery Error** (MRE), defined for a generator G and a dataset Y as

$$\text{MRE}_G(\mathcal{Y}) = \text{median}\left\{\min_z \|y_i - G(z)\|^2\right\}_{y_i \in \mathcal{Y}} \quad (4)$$

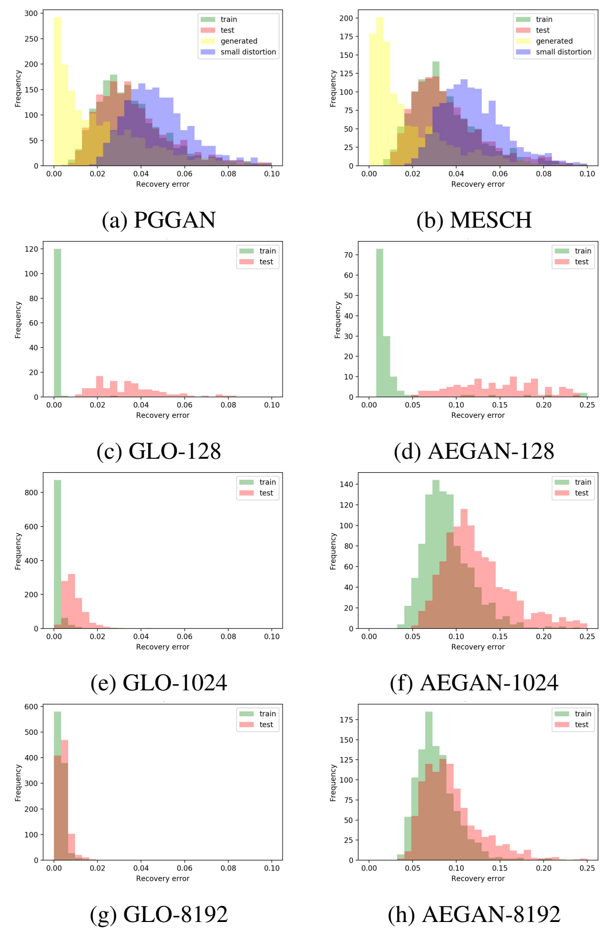
Table 1 reports such values for other deep generators and other datasets.

Then to measure the distance between two distributions, that is to estimate to which extent the generator overfits the training set, we simply compute the normalized **MRE-gap** between validation $\mathcal{T}$ and train $\mathcal{D}$ dataset, which writes

$$\text{MRE-gap}_G = \left(\text{MRE}_G(\mathcal{T}) - \text{MRE}_G(\mathcal{D})\right) / \text{MRE}_G(\mathcal{T}) \quad (5)$$

These values are reported[2] in Table 1.

[2]Notice that other metrics could have been used, such as the Wasserstein distance.



(a) PGGAN (b) MESCH (c) GLO-128 (d) AEGAN-128 (e) GLO-1024 (f) AEGAN-1024 (g) GLO-8192 (h) AEGAN-8192

Figure 4: *Histograms of recovery errors on train $\mathcal{D}$ and validation $\mathcal{T}$ datasets from CelebA-HQ showing that overfitting is not happening for PGGAN and MESCH generators on the training dataset, but is for GLO-N and AEGAN-N when training for a small dataset $N \leq 8192$.*

Instead of using an empirical threshold to automatically assess if the amount of overfitting is significant regarding the size of the training set, we rely on a statistical test. We compute the p-value of the Kolmogorov-Smirnov test (KS) which measures the probability that two random samples drawn from the same distribution have a larger discrepancy, defined as the maximum absolute difference between cumulative empirical distributions, than the one observed.

Such p-values are displayed in Table 1, and a threshold of 1% is used to detect overfitting (values are highlighted). To show the consistency between the two proposed metric, we also highlight the values of MRE-gap that are above 10%.

Observe that the results are mostly confirming previous empirical evidence: memorization is strongly correlated to the number of images seen during training. We also see that the same overfitting occurs on different datasets (CelebA-HQ and LSUN), and for autoencoder (AE). At $N = 26000$ on CelebA-HQ and $N = 32768$ on LSUN bedroom, overfitting is no longer detectable.

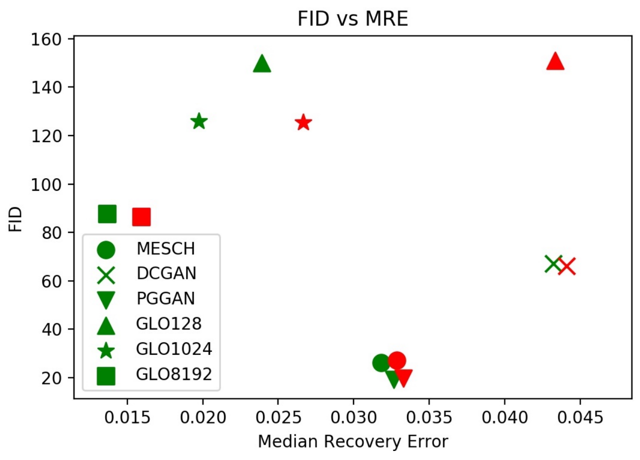


Figure 5: *Comparison of FID versus Median Recovery Error (MRE) for various models computed over training images (in green) and validation images (in red). FID does not detect memorization in GLO models.*

However, using the proposed statistics (p-value, normalized MRE-gap) is much more practical to detect overfitting than only inspecting histograms and easier to threshold than MRE itself. It also illustrates that such statistical principle overrules empirical evaluation, as memorization is indeed sometimes quite hard to tell from simple visual inspection, such as for the AEGAN generator in Fig. 3.

3.4. FID Does Not Detect Memorization

The FID is the standard GAN evaluation metric for images [16], so it is natural to ask whether this metric can be used to detect memorization. Figure 5 displays FID scores computed between generated and training images (in green) and generated and test images (in red). While the median recovery error (MRE) is able to detect memorization in GLO models, the FID is not sensitive to this (this fact was also noted by [14]). We do not suggest replacing the FID, but rather using MRE to provide a more complete picture of generator performance. Besides, other metrics such as the precision recall introduced in [33] can be considered as well to tackle more subtle statistical biases such as mode dropping versus mode invention.

4. Discussion and Future Work

4.1. Notes on Applications

Recently, GANs have seen wide application to various face generation tasks, such as face attribute modification [8], generative face completion [22] and face super-resolution [9]. In a similar vein, deep image prior [38], recovers images by first fixing a random latent vector, then

Table 1: *Kolmogorov-Smirnov (KS) p-values, normalized median error difference (MRE-gap) Eq.* (5)*, and Median recovery errors (MRE) Eq.* (4) *for a variety of generators. Highlighted values indicate generators for which overfitting of the training set has been detected: (in blue) with the KS test using* 1% *threshold on p-value, (in green) using* 10% *threshold on MRE-gap.*

		KS p-value	MRE-gap	MRE			
		train vs val		train	val	generated	distortion
CelebA-HQ	DCGAN	9.43e-01	1.79e-02	4.95e-02	5.04e-02	3.68e-03	5.69e-02
	MESCH	4.55e-01	6.96e-03	3.40e-02	3.43e-02	1.77e-02	4.63e-02
	PGGAN	2.22e-01	2.22e-02	3.31e-02	3.39e-02	1.78e-02	4.65e-02
	GLO-128	**0.00e+00**	9.70e-01	9.94e-04	3.30e-02	5.10e-05	9.32e-03
	GLO-1024	**0.00e+00**	7.59e-01	1.95e-03	8.08e-03	1.29e-03	4.46e-03
	GLO-8192	**2.25e-18**	1.75e-01	3.00e-03	3.64e-03	1.04e-03	3.20e-03
	GLO-26000	2.12e-01	3.69e-02	4.27e-03	4.44e-03	4.08e-04	4.43e-03
	AE-128	**0.00e+00**	9.68e-01	3.36e-03	1.06e-01	N/A	1.80e-02
	AE-1024	**0.00e+00**	9.35e-01	4.19e-03	6.45e-02	N/A	1.80e-02
	AE-8192	**0.00e+00**	7.60e-01	8.04e-03	3.34e-02	N/A	1.67e-02
	AEGAN-128	**0.00e+00**	9.02e-01	1.54e-02	1.57e-01	N/A	2.82e-02
	AEGAN-1024	**0.00e+00**	2.68e-01	8.52e-02	1.16e-01	N/A	8.69e-02
	AEGAN-8192	**3.17e-27**	1.61e-01	7.42e-02	8.84e-02	N/A	7.55e-02
	AEGAN-26000	1.25e-01	1.85e-02	9.96e-02	1.01e-01	N/A	1.00e-01
LSUN	DCGAN (tower)	7.02e-02	1.36e-02	7.96e-02	8.07e-02	1.49e-02	7.31e-02
	DCGAN (bedroom)	3.65e-01	5.34e-03	7.06e-02	7.10e-02	7.03e-02	7.09e-02
	GLO-8192 (bedroom)	**6.70e-06**	1.70e-01	5.45e-03	6.56e-03	5.37e-04	5.01e-03
	GLO-32768 (bedroom)	2.62e-01	5.40e-02	6.58e-03	6.25e-03	8.40e-04	5.44e-03
MNIST	DCGAN	2.41e-01	8.85e-02	3.00e-02	2.75e-02	6.89e-03	-
	GLO-1024	**0.00e+00**	6.78e-01	2.86e-04	8.88e-04	1.49e-03	-
	GLO-16384	3.48e-01	6.45e-03	8.72e-04	8.77e-04	1.41e-03	-
	AEGAN-16384	7.43e-02	2.29e-02	4.56e-02	4.67e-02	N/A	-
CIFAR10	DCGAN	5.40e-01	3.65e-03	2.29e-01	2.28e-01	1.30e-03	-
	GLO-1024	**0.00e+00**	5.84e-01	2.77e-03	6.67e-03	8.53e-04	-
	GLO-16384	3.48e-01	6.45e-03	8.72e-04	8.77e-04	1.41e-03	-

optimizing over the parameters of a randomly initialized generator. We apply ($\text{NN}_\mathcal{G}$) to face inpainting and super resolution for two reasons; first it shows off-the-shelf GAN generators are well suited for a variety of downstream tasks, which is also noted in [39] and second it provides additional visual insight into the observations of the previous section.

Figure 6 shows the progressive GAN generator [20] applied to face inpainting (ϕ is a mask) and super-resolution (ϕ is a 64x pooling). While the face inpainting is artifacted, we note that the results are decent without any post processing and similar to those presented in [22] (while being non-feedforward). As for super-resolution, we obtain results at least on par with [9]. An intriguing property of the images is that the recovery is semantically accurate, in terms of attributes such as gender, facial features and pose, whilst recovering a face that appears to be a different identity. This happens despite the use of images that the PGGAN generator [20] was trained on, which is in accordance with the observations of Sec. 3. Put in another way, we believe the fact that PGGAN has generalized well to CelebA-HQ, also means that it will not be able to verbatim recover an identity found in the dataset. For some applications this could be seen as inadequate, such as the domain translation network of StarGAN [8], wherein a user wants to retain identity but change facial features. On the other hand, if a face dataset is considered private or copyrighted, not verbatim copying any training image can be seen as a benefit of the algorithm. Quantifying whether GAN generators really do generalize with respect to identity, using a face identification network like VGG-Face [31], is an interesting issue that we leave for future work.

4.2. Future Work

Our work is a part of a growing body of research concerned with overfitting of deep generative models [15, 14]. For example, [14] takes a perspective of GAN evaluation, arguing that evaluation with a neural net distance can penalize trivial memorization of the dataset whereas FID cannot. We have a similar perspective, albeit with the goal of merely detecting overfitting and with a simpler approach.

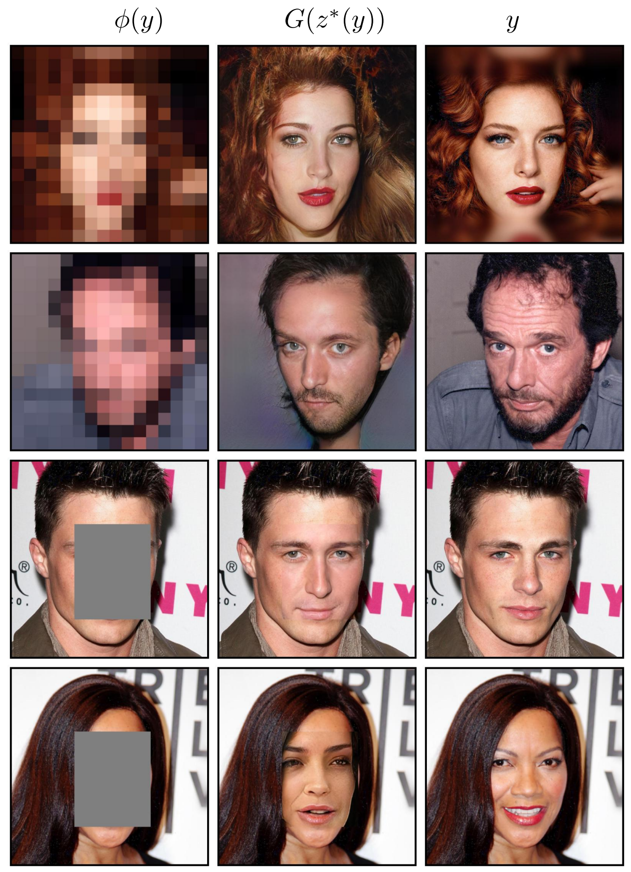


Figure 6: *Off the shelf application of Eq.* (NN$_{\mathcal{G}}$) *with a 1024×1024 generator PGGAN. From left to right: transformed image $\phi(y)$, recovered image $G(z^*(y))$ and ground truth image y. The first two rows are image super-resolution (ϕ is a mask) and next two are image inpainting of images downsampled by a factor of 64 (ϕ is an average pooling).*

Unlike our approach, the NND [18, 14] was able to detect slight overfitting of some GANs, however, the massive size of the validation set and the fact that the NND must be retrained for every generator under evaluation make the analysis computationally burdensome. Additionally, the architecture choice of the NND may be biased to reflect the GAN loss function and not be universal across models, such as the reconstruction based GLO model considered in this work. On the other hand, we present a simple, computationally tractable solution requiring modestly sized (here, just a few thousand) validation images. We found the reconstruction based generator GLO to be interesting from a theoretical perspective. For example, optimization unveiled strong overfitting; training images are nearly verbatim recovered and validation images are blurry (e.g. Fig. 3, row 4), whereas FID on GLO samples was insensitive to this difference. Furthermore, a major advantage of our work is the visual interpretability of our results. For example, one can see in Fig. 3, the visual reconstructions do in fact reveal visual quality of train versus validation samples. Namely, both training and validation images are well reconstructed for the GAN methods. We believe further work must be done to synthesize the results of [14, 15] and our work. One promising area would be to explore other loss functions. We experimented briefly with perceptual losses (see supplementary) but leave this possibility open. We also think recovery could be guided with a learned NND loss. Another interesting direction is analysis of local overfitting on image patches. Preliminary experiments can be found the in supplementary material, which also show generalization of GAN generators. Finally, Eq. (NN$_{\mathcal{G}}$) had mixed success for more complex datasets such as LSUN in terms of visual quality. We think that some datasets lead to more complex latent space with many local minima and direct the reader to the supplementary material for more details on optimization.

Conclusion In this work, we studied overfitting of deep generators through latent recovery. We saw that a simple Euclidean loss was effective at recovering latent codes and recovers plausible images even after image transformations. We used this fact to study whether a variety of deep generators memorize training examples by asking if the network can generate validation samples. Our statistical analysis revealed that overfitting was undetectable for GANs, but detectable for hybrid adversarial methods like AEGAN and non-adversarial methods like GLO, even for training sets of moderate sizes. Due to the ever-growing concerns on privacy or copyright of training data and the already widespread application of generative methods, we provide methodology that is a step in the right direction towards analysis of generative overfitting.

Acknowledgments This work was supported by fundings from *Région Normandie* under grant *RIN NormanD'eep*. This study has also been carried out with financial support from the French State, managed by the French National Research Agency (ANR GOTMI) (ANR-16-CE33-0010-01).

References

[1] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 224–232. PMLR, 06–11 Aug 2017. 2

[2] S. Arora, A. Risteski, and Y. Zhang. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018. 2

[3] D. Berthelot, T. Schumm, and L. Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 3

[4] P. Bojanowski, A. Joulin, D. Lopez-Pas, and A. Szlam. Optimizing the latent space of generative networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 600–609. PMLR, 10–15 Jul 2018. 2, 3, 4, 5

[5] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. *Thirty-fifth International Conference on Machine Learning (ICML)*, 2018. 3

[6] A. Borji. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*, 2018. 2

[7] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 1

[8] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint*, 2018. 2, 6, 7

[9] R. Dahl, M. Norouzi, and J. Shlens. Pixel recursive super resolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5449–5458, Oct 2017. 2, 6, 7

[10] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017. 2

[11] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *ICLR*, 2017. 3

[12] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015. 2, 3

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 4

[14] I. Gulrajani, C. Raffel, and L. Metz. Towards GAN benchmarks which require generalization. In *International Conference on Learning Representations*, 2019. 2, 6, 7, 8

[15] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019. 3, 7, 8

[16] 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. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 2, 3, 6

[17] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 2

[18] D. J. Im, A. H. Ma, G. W. Taylor, and K. Branson. Quantitatively evaluating GANs with divergences proposed for training. In *International Conference on Learning Representations*, 2018. 2, 8

[19] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 3

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *Sixth International Conference on Learning Representations (ICLR)*, 2018. 1, 4, 5, 7

[21] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10236–10245, 2018. 2

[22] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 2, 3, 6, 7

[23] Z. C. Lipton and S. Tripathi. Precise recovery of latent vectors from generative adversarial networks. *ICLR*, 2017. 3

[24] M. Lui, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2, 3, 4

[25] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 2, 3

[26] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3478–3487, 2018. 4, 5

[27] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017. 3

[28] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. 2

[29] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015. 3

[30] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. Towards the science of security and privacy in machine learning. *3rd IEEE European Symposium on Security and Privacy*, 2018. 3

[31] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 7

[32] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1, 4

[33] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018. 2, 6

[34] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 2

[35] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017. 3

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 3

[37] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, Apr 2016. 2

[38] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018. 6

[39] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017. 2, 7

[40] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 2

[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 2, 5