Reference-based GAN Evaluation by Adaptive Inversion

Jianbo Wang
The University of Tokyo
jianbowang815@gmail.com

Heliang Zheng
University of Science and Technology of China
zhenghllj@gmail.com

Toshihiko Yamasaki
The University of Tokyo
yamasaki@cvm.t.u-tokyo.ac.jp

Abstract

Common evaluation metrics for generative models, such as the Fréchet Inception Distance (FID), can produce variable results due to their reliance on image sampling. Some methods accelerate the convergence to optimal FID scores by utilizing pre-trained models for constructing discriminators, a process that bears similarity to the operational mechanism of the FID metric itself. This approach can lead to an overestimation of performance. Consequently, while the FID scores may improve, the visual quality of the generated images may deteriorate. To better evaluate the visual quality of a GAN model, we propose a reference-based evaluation metric for GAN by leveraging GAN Inversion. Specifically, our evaluation metric measures the invertibility of a GAN model to invert unseen images from the target distribution. Experimental results demonstrate that our proposed evaluation metric could effectively measure GAN models under the same image content, especially for those trained with pretrained vision models.

1. Introduction

Recently, unconditional generative adversarial networks (GANs) have been able to generate images of remarkable quality. GANs have achieved impressive photo-realism for image synthesis tasks.

Evaluating GANs presents significant challenges due to the complexity and variety of these models. Assessing GANs is difficult because they generate new, synthetic instances of data, which can be hard to quantify and compare objectively. Traditional methods of evaluation, such as Peak Signal to Noise Ratio (PSNR) [10] used in supervised learning, are not directly applicable to GANs. Furthermore, the inherent stochastic nature of these models means that they can produce different outputs even when fed the same input, adding another layer of complexity to their evaluation.

Currently, the main metrics for evaluating GAN models include the Inception Score (IS) [7] and the Frechet Inception Distance (FID) [2]. IS measures the diversity and quality of images generated by a GAN, using a pre-trained Inception model to predict the class labels of generated images. The FID score, on the other hand, compares the distribution of generated images to that of real images, aiming to quantify how similar the two sets of images are. However, these metrics have their limitations. IS can't detect mode collapse, a common problem in GANs where the model generates a limited variety of outputs. FID, while useful in comparing the quality of images, relies heavily on the Inception model and is limited to the visual quality that the model can capture. Both metrics can be influenced by factors such as dataset bias and the specific architecture of the Inception model used.

Recently, Axel et al. proposed Projected GANs [8], which achieve the same FIDs as the previous best methods up to 40 times faster. This is because the model utilizes the utility of pretrained representations to improve and stabilize GAN training. By incorporating a strong backbone from a pre-trained classifier as the discriminator, these GANs leverage the extensive learning and feature extraction capabilities that have already been established in these classifiers. Since the discriminator in Projected GANs is adapted from a pre-trained classifier, similar to how the FID metric operates, there is a risk that the FID scores might not truly represent the quality of the images generated by these GANs. Essentially, this similarity between the GAN's discriminator and the FID's evaluation process could result in inflated FID scores, providing a somewhat misleading view of the GAN's capabilities (as shown in Fig. 1).

To address these challenges, we propose an evaluation method that involves GAN evaluation through GAN inversion. In this approach, two generators are evaluated under the same images, allowing for a more direct comparison of their performance. This method is not only helpful for

StyleGAN2, FID=5

Projected GANs, FID=3

Figure 1. Illustration of visual comparison for StyleGAN2 and Projected GANs after . Projected GANs is better in terms of FID score, but suffers from low visual quality.

human evaluators but also offers a more robust alternative to FID. Unlike traditional metrics, this approach does not work similarly to the loss function, reducing the likelihood of the evaluation metric being manipulated or attacked. Additionally, this method allows for the evaluation of GANs at different granularities, providing a more nuanced understanding of their performance.

2. Related Works

2.1. Evaluation of Generative Adversarial Networks

Qualitative Evaluation of GANs Human evaluators are shown pairs of images and asked to choose the more realistic one, with the performance of a generative model assessed based on the frequency of preference for its images. Scores are averaged across evaluators to reduce variance, but a limitation is the potential evolution of judges' performance over time.

Quantitative Evaluation of GANs

The quantitative metrics for GAN evaluation can be divided into Content-Variant Metrics and Content-Invariant Metrics, contingent upon whether the ground truth and input images are matching pairs. The Content-Variant Metrics include Inception Score (IS) and Fréchet Inception Distance (FID). Content-Invariant Metrics include Learned Perceptual Image Patch Similarity (LPIPS), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

- *Inception Score (IS)*: This metric evaluates the quality and diversity of generated images by leveraging a pre-trained Inception-v3 classifier network to compute the average class probability distribution for the generated images [7]. A higher IS indicates superior model performance.
- *Fréchet Inception Distance (FID)*: This metric quantifies the similarity between the feature distributions of real and fake images in a high-dimensional feature space, utilizing a pre-trained Inception-v3 network to extract features from both image sets and computing the FID between their distributions [2]. A lower FID signifies better performance.
- *Perceptual Image Patch Similarity (LPIPS)*: This metric calculates the distance between the feature representations of the two images at multiple layers of a pre-trained network and aggregates these distances to yield a similarity score [12]. A lower LPIPS value indicates better performance.
- *Structural Similarity Index (SSIM)*: This metric assesses the similarity between two images by comparing their pixel values at each point within a local window, considering both contrast and structural similarities within the window regions [10]. A higher SSIM value denotes better performance.
- *Peak Signal to Noise Ratio (PSNR)*: PSNR is a commonly used metric for evaluating the quality of reconstruction in lossy compression codecs, by comparing the similarity between the original and compressed images [10]. A higher PSNR indicates better performance.

2.2. GAN Inversion

GAN inversion is a technique aimed at inverting a given image back into the latent space of a pretrained GAN model [11].

Learning-based GAN Inversion. It involves the use of an encoding neural network that is trained to map an image into a latent code. The key objective here is to optimize the encoder so that it can effectively transform the input image into a latent representation suitable for the GAN to reconstruct the image. This approach is notable for its efficiency in generating latent representations that are both accurate and robust for various images [6, 9, 13].

Optimization-based GAN Inversion. The second cat-

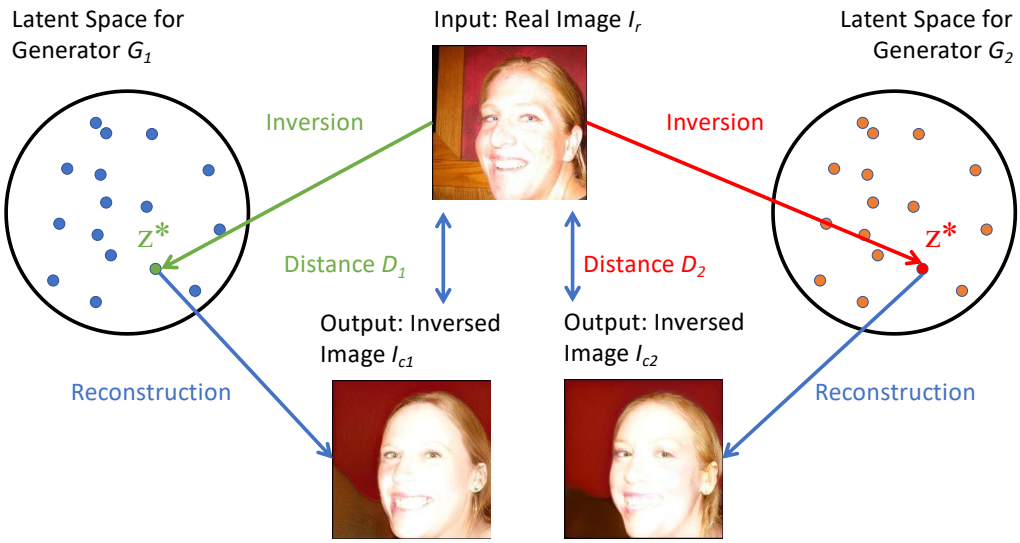

Figure 2. Illustration of the general pipeline for our proposed evaluation method. An image from the real world is utilized as an input to conduct an evaluation of two generative models, namely Generator #1 and Generator #2. This image is individually inverted into the latent spaces corresponding to each of the two generators, resulting in two distinct inverted outputs: Reconstructed Image #1 and Reconstructed Image #2. The quality of each generator is quantitatively assessed by measuring and comparing the distances (Distance #1and Distance #2) between the inverted images and the original real image.

egory focuses on the direct optimization of the latent code to achieve accurate reconstruction of the target image. It involves iteratively tweaking the latent code to minimize the difference between the generated image and the input image. This method is particularly useful when precision in image reconstruction is paramount, as it allows for fine-grained control over the generated output [1, 4].

3. Approach

If a given image can be successfully inverted, meaning a latent code is identified that enables the generator to produce an image identical to the given one, it demonstrates the generator's capability to recreate that specific image. Extending this concept to multiple images, if a model can effectively invert a set of test images, it indicates the model's proficiency in generating a diverse range of images. Such an ability to accurately invert a collection of images from a test set suggests a high level of competency in the generator. This proficiency in inverting a spectrum of images is a strong indicator of the quality and robustness of the generator within the GAN framework. Building upon this concept, we propose an evaluation methodology based on GAN inversion. This approach leverages the inversion process as a metric to assess the capabilities of a GAN's generator.

3.1. Inversion Process

We introduce a method shown in Fig. 2 for assessing the efficacy of generative models G, utilizing a dataset comprising N test images. The detailed pipeline can be found in Alg. 1. The procedure commences by inverting each image in the dataset $I_r^1, I_r^2, \ldots, I_r^N$ into the latent space of the generative model G. Inverting an image I^i into the latent space of G is represented as

$$\mathbf{C}^i = G^{-1}(I^i), \tag{1}$$

where $\mathbf{C}^i$ denotes the inverted representation of I_r^i in the latent space, and G^{-1} symbolizes the inverting operation applied by the model G. This inverting, executed via an optimization process, generates a set of inverted codes $\{\mathbf{C}_r^1, \mathbf{C}_r^2, \ldots, \mathbf{C}_r^N\}$.

Finally, the reconstructed image I_C^i corresponding to the optimized latent vector $\mathbf{C}$ is obtained through

$$I_C^i = G(\mathbf{C}^i). \tag{2}$$

In this expression, I_C^i represents the final reconstructed image, achieved by passing the optimized latent vector $\mathbf{C}^i$ through the generative model G. Subsequently, the approach involves a comparative analysis, where the LPIPS [12] metric is employed to compute the distances between each corresponding pair of the original and the reconstructed images. The distance for each pair is denoted

Algorithm 1: The evaluation process of the proposed method.

Input: N Testing Images $\{I_r^1, I_r^2, \ldots, I_r^N\}$, testing Generators G

Output: Evaluation Score for G, namely S

Invert each image in $\{I_r^1, I_r^2, \ldots, I_r^N\}$ into latent space for G by optimization process, obtaining inverted code $\{\mathbf{C}_r^1, \mathbf{C}_r^2, \ldots, \mathbf{C}_r^N\}$

Feed $\{\mathbf{C}_r^1, \mathbf{C}_r^2, \ldots, \mathbf{C}_r^N\}$ to G as inputs, obtaining reconstructed output $\{I_c^1, I_c^2, \ldots, I_c^N\}$

Given two sets of images $\{I_c^1, I_c^2, \ldots, I_c^N\}$ and $\{I_r^1, I_r^2, \ldots, I_r^N\}$, calculate the LPIPS distances between corresponding pairs of images as $\{D^1, D^2, \ldots, D^N\}$, where each D^i represents the distance between I_c^i and I_r^i.

The score S is then defined as the average of these distances: $S = \frac{1}{N}\sum_{i=1}^{N} D^i$

as D^i, representing the degree of similarity between I_c^i and I_r^i.

The overall performance score S of the generative model G is determined by averaging these distances. This score provides an insightful quantitative measure of the model's ability to faithfully reconstruct images from their corresponding latent representations, thus serving as a pivotal benchmark for evaluating the model's reconstruction accuracy.

3.2. Optimization Methods

3.2.1 Vanilla Optimization Process

To achieve this inverting, an optimization process is utilized, formalized as

$$\mathbf{C}_r = \arg\min_{\mathbf{C}_r} \mathcal{L}(G(\mathbf{C}_r), I). \tag{3}$$

Here, $\mathbf{C}_r$ signifies the optimized latent vector that best represents the image I_r within the latent space of G. The function $\mathcal{L}$ represents a loss function, which quantifies the discrepancy between the reconstructed image $G(C_r)$ from a latent vector $\mathbf{C}_r$ and the original image I_r. The optimization aims to find the latent vector $\mathbf{C}_r$ that minimizes this loss, thereby ensuring the closest possible representation of I_r in the latent space. The whole pipeline is outlined in Alg. 2.

3.2.2 Improved Optimization Process

We introduce an improved optimization strategy designed to effectively generate latent codes from a testing image using a GAN and its associated mapping network. The process is encapsulated in Alg. 3, which initiates with the sampling

Algorithm 2: Algorithm of Vanilla Optimization Process.

Input: Testing image I_c, generator G, mapping network M, the number of random sampled vectors T.

Output: Latent code $\mathbf{C}_r$

Sample T random vectors $\mathbf{z}_{1:T}$ and initialize them with a normal distribution.

$\mathbf{w}_{1:T} = M(\mathbf{z}_{1:T})$

$\overline{\mathbf{w}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{w}_i$, where $\overline{\mathbf{w}}$ denotes the average.

Initialize $\mathbf{C}_r = \overline{\mathbf{w}}$

for *number of iterations* **do**

  Update $\mathbf{C}_r$ by minimizing:

  $\mathbf{C}_r = \arg\min_{C_r} \mathcal{L}(G(\mathbf{C}_r), I_r)$.

return $\mathbf{C}_r$

Algorithm 3: Algorithm of Improved Optimization Process.

Input: Testing image I_c, generator G, mapping network M, the number of random sampled vectors T, the number of coefficients learned n_C.

Output: Latent code $\mathbf{C}_r$

Sample T random vectors $\mathbf{z}_{1:T}$ and initialize them with a normal distribution.

$\mathbf{w}_{1:T} = M(\mathbf{z}_{1:T})$

Calculate the mean vector: $\overline{\mathbf{w}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{w}_i$

Compute the covariance matrix: $C = \frac{1}{T-1}\sum_{i=1}^{T}(\mathbf{w}_i - \overline{\mathbf{w}})(\mathbf{w}_i - \overline{\mathbf{w}})^T$

Compute the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ of C

Sort the eigenvectors $\mathbf{v}_i$ by descending eigenvalues λ_i

Let PC be the matrix of the first n_C sorted eigenvectors: $PC = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_C}]$

Initialize $\mathbf{w}_a = \mathbf{0} \in \mathbb{R}^{1\times n_c}$

for *number of iterations* **do**

  $\mathbf{C}_r = \overline{\mathbf{w}} + \mathbf{w}_a \times \mathrm{PC}$

  Update $\mathbf{w}_a$ by minimizing:

  $\mathbf{w}_a = \arg\min_{\mathbf{w}_a} \mathcal{L}(G(\mathbf{C}_r), I_r)$.

return $\mathbf{C}_r$

of T random vectors $\mathbf{z}_{1:T}$ from a standard normal distribution. These vectors are subsequently processed through a mapping network M, yielding a series of $\mathbf{w}$-vectors, $\mathbf{w}_{1:T}$, located within the GAN's latent space.

Central to our method is the application of Principal Component Analysis (PCA) to the set of $\mathbf{w}$-vectors. We begin by computing the mean vector $\overline{\mathbf{w}}$ and the covariance

matrix C of the $\mathbf{w}$-vectors as follows:

$$\overline{\mathbf{w}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{w}_i, \tag{4}$$

$$C = \frac{1}{T-1}\sum_{i=1}^{T}(\mathbf{w}_i - \overline{\mathbf{w}})(\mathbf{w}_i - \overline{\mathbf{w}})^T. \tag{5}$$

Subsequent to the computation of $\overline{\mathbf{w}}$ and C, we derive the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$. These eigenvectors are sorted based on the descending order of their corresponding eigenvalues, and the principal components are extracted by selecting the first n_C eigenvectors to construct the matrix PC:

$$PC = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_C}]. \tag{6}$$

The optimization of the latent code $\mathbf{C}_r$ commences with the initialization of the coefficient vector $\mathbf{w}_a$ as a zero vector in $\mathbb{R}^{1\times n_C}$. Iteratively, over a number of iterations, we update $\mathbf{C}_r$ by computing:

$$\mathbf{C}_r = \overline{\mathbf{w}} + \mathbf{w}_a \times PC. \tag{7}$$

Here, $\mathbf{w}_a$ is refined by minimizing the loss function $\mathcal{L}$, which quantifies the difference between the generated image $G(\mathbf{C}_r)$ and the target testing image I_r. This iterative refinement of $\mathbf{C}_r$ aims to align the generated image more closely with the target image.

By leveraging the principal components within the latent space, our approach ensures that modifications to the latent code are directed along the most significant axes of variation. This strategy leads to a more streamlined and effective optimization process, particularly within the intricate latent spaces typical of GANs.

Impact of Principal Component Selection. The number of principal components denoted as n_c can balance fidelity and computational efficiency.

Higher n_c: Improved representation enhances detail and accuracy in latent space, thus improving inversion precision. However, there is a risk of overfitting; with limited data, it may capture noise, thereby reducing generalization.

Lower n_c: Model simplification focuses on key variations, which speeds up computation and possibly reduces overfitting. Nonetheless, it risks information loss, potentially losing crucial details that impact inversion quality.

4. Experiments and Discussions

4.1. Experimental Setup.

In the experimental segment, our approach involved randomly selecting 1,000 images from the FFHQ validation dataset [3] to constitute our evaluation set. We conducted assessments on both Projected GAN and StyleGAN models. More precisely, we utilized a ProjectedGAN model that underwent 183,456 iterations of training on the FFHQ training dataset. Additionally, we evaluated StyleGAN models that were trained for 1,008, 5,846, and 20,160 iterations on the same dataset. For clarity and precision in our comparisons, these models are designated as StyleGAN at 1,008 iterations (StyleGAN-1008), StyleGAN at 5,846 iterations (StyleGAN-5846), and StyleGAN at 20,160 iterations (StyleGAN-20160).

Throughout our testing procedures, we consistently applied a PCA with a principal component ratio of 0.9 (has the same meaning with n_c) and employed the Adam optimizer with hyperparameters set to 0.90 and 0.99. The number of random sampled vectors T is set to be 100,000.

4.2. Comparison Results

We compares StyleGAN models at different training iterations with a Projected GAN model trained for 183,456 iterations. Fig. 3 highlights contradictory evaluations between FID scores and visual quality using pre-trained discriminator features. Despite Projected GAN's lowest FID (FID equal to 3), suggesting superior image quality, its visual output quality is comparable to StyleGAN-1008 (FID equal to 27.15), as seen in Fig. 4. Our scoring aligns more consistently with visual quality assessments (see Fig. 4).

4.3. Ablations

To verify the effect of each component in Sec. 3, we conduct an ablation study by visually comparing the generated results without each of them.

Different Dataset The images used for testing are sourced from the FFHQ test set, while the models tested were trained on the training set of the FFHQ dataset. Although these models have not been exposed to the test images previously, we also present results on other facial datasets for a comprehensive evaluation. Fig. 5 demonstrates the inverting results and scores of four models on the CeleA dataset [5]. It is observable that, despite the change in dataset, the relative performance ranking of the models, as assessed by our score, remains consistent with the results obtained on the FFHQ dataset. This consistency underscores the effectiveness of our proposed evaluation method across different datasets.

Different Optimization Strategies As mentioned in Sec. 3.2, we explored two distinct optimization approaches: Vanilla Optimization and Improved Optimization. To ensure that the optimized latent code resides on the generator's manifold and to enhance efficiency by reducing the number of parameters needed for optimization (via PCA), we adopted the Improved Optimization strategy. Fig. 6 compares the outcomes of these two methods applied to the StyleGAN model. It reveals that different optimization

Figure 3. The visual comparison of the inverting process and their corresponding quantitative evaluations for the four models on FFHQ dataset. From top to bottom, each row represents the real image I_c, followed by the visual outcomes after inverting for StyleGAN-1008, StyleGAN-5846, StyleGAN-20160, and Projected GAN, respectively. Notably, Projected GAN exhibits the lowest FID value, indicating superior visual quality. Conversely, StyleGAN-20160 has the lowest value on our score, which also denotes the highest visual quality.

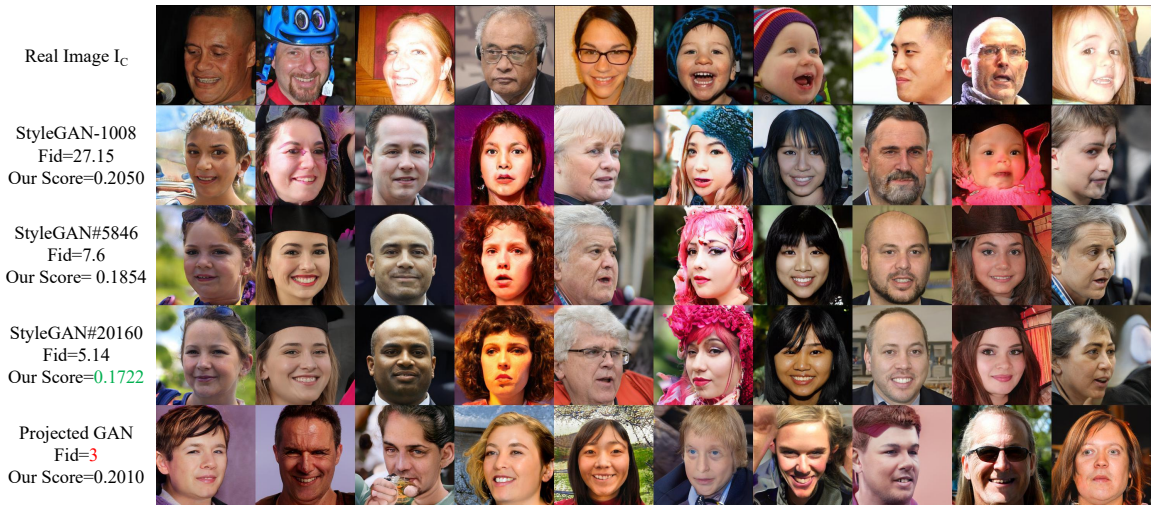

Figure 4. The visual comparison of visual outcomes generated by four models. From top to bottom, each row sequentially displays the real image I_c, followed by the visual effects of images generated by StyleGAN-1008, StyleGAN-5846, StyleGAN-20160, and Projected GAN. It is important to note that due to the uniform latent space architecture employed by the StyleGAN models, we utilized the same latent code for image generation. This approach facilitates a more pronounced comparison of the visual effects across different iterations.

strategies significantly impact the visual quality of model inversion. Generally, Vanilla Optimization yields lower scores, indicating that the inverting results are closer to the original image. Similarly, our scoring system consistently ranks model performance in the same order across different optimization strategies.

Different PCA Settings In Sec. 3.2.2, we discussed how the number of principal components selected in PCA impacts the effectiveness of the Improved Optimization strategy for image inverting. As illustrated in Fig. 7, rather than directly setting the number of components n_c , we employ the parameter $PCA@X$. Here, the number of components is selected such that the explained variance exceeds the threshold percentage X. This approach allows us to evaluate GANs under varying levels of granularity. For higher values of X, the computed scores are lower, which is rea-

Figure 5. The visual comparison of the inverting process and their corresponding quantitative evaluations for the four models on CelebA dataset.

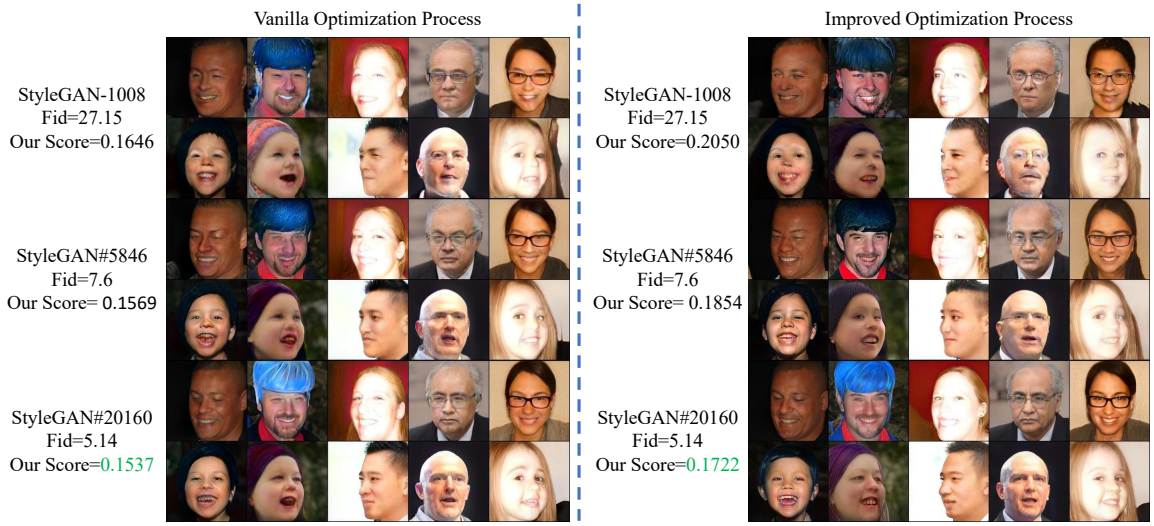

Figure 6. The visual comparison for different optimization strategies.

sonable since more features are retained during the inverting process. Additionally, our scoring system consistently ranks the performance of models in the same order across different PCA settings.

Different Optimizers During the image inverting process, the choice of optimizer can significantly impact the results. In Fig. 3, we selected Adam as the default optimizer. In Fig. 8, we compared the outcomes of using the SGD optimizer against the Adam optimizer. We observed that employing the SGD optimizer enhanced the visual appeal of the inverted results, yet these results exhibited greater divergence from the original image. This can be attributed to the fact that SGD performs more meticulous adjustments during the optimization process, aiding in the precise matching of certain low-level features of the image. However, this approach may lead to an increased disparity in the overall high-level information relative to the original image.

5. Conclusion

We presented a novel approach for evaluating GANs through GAN inversion, offering an insightful and robust metric beyond the conventional FID. Our methodology capitalizes on the inversion capability of GANs, providing a

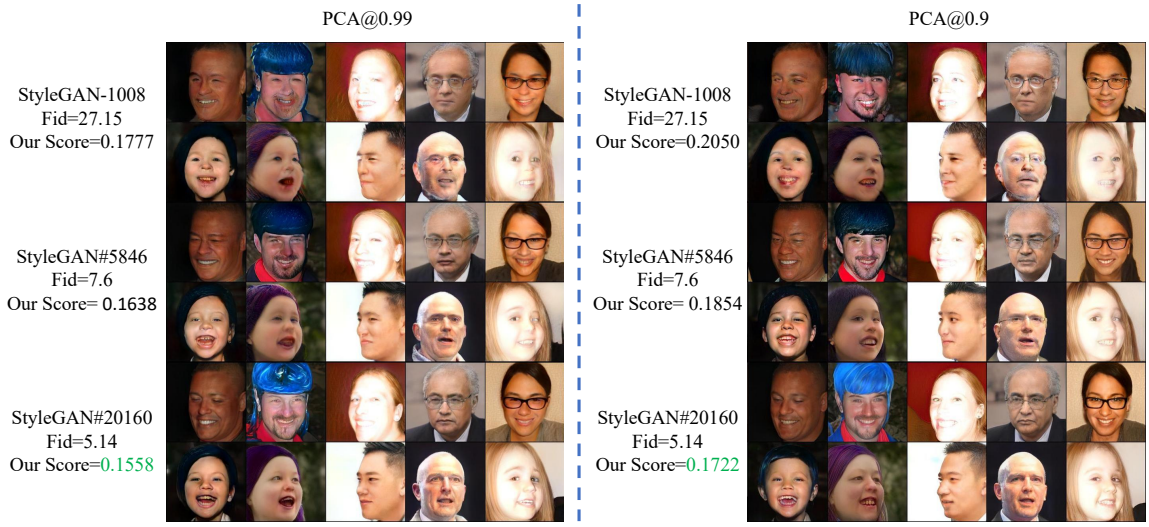

Figure 7. The visual comparison for different PCA Settings.

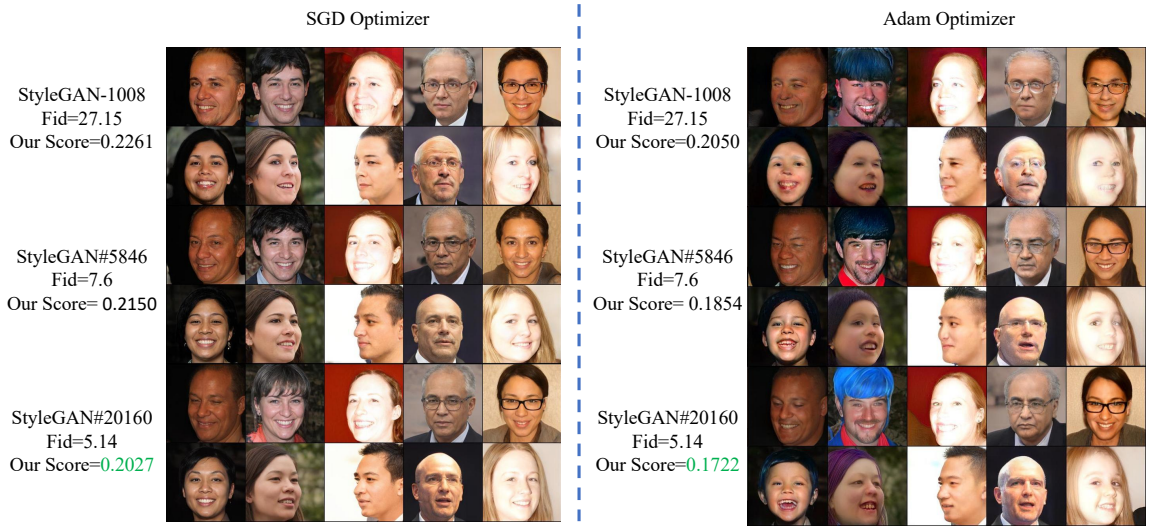

Figure 8. The visual comparison for different optimizers.

direct and comprehensive assessment of their performance across various datasets and optimization strategies. Our experiments, conducted on multiple models including Projected GAN and StyleGAN variants, demonstrated the efficacy of our evaluation method. By comparing these models under different conditions – such as varying PCA settings, optimization methods, and the use of different optimizers – we highlighted the nuances of GAN performance in image synthesis tasks. However, we also acknowledge certain limitations. The time-consuming nature of the inverting process, especially for high-resolution images, indicates the need for more efficient optimization techniques.

Acknowledgements

T. Yamasaki was supported by SPS KAKENHI Grant Number JP22H03640 and Institute for AI and Beyond of The University of Tokyo.

References

[1] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, 30(7): 1967–1974, 2018. 3

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 1, 2

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 5

[4] 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 Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3

[5] 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*, 2015. 5

[6] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2

[7] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 1, 2

[8] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021. 1

[9] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics*, 40(4):1–14, 2021. 2

[10] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1, 2

[11] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3121–3138, 2022. 2

[12] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2, 3

[13] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476. 2017. 2