Replicating "Image-Adaptive GAN Based Reconstruction"

Antonio Marino, Yilong Song, and Daisuke Yamada

October 14, 2025

Abstract

The image inverse problem is the problem of reconstructing an image given its degraded or compressed observation. Some previous solutions to this problem use generative adversarial networks (GANs) [4], but the representation capabilities of such models cannot capture the full distribution of complex classes of images (e.g., human faces), thus producing sub-optimal results. Our work examines the image-adaptive generative model, proposed in Hussein et al (2020), that purports to mitigate the limited representation capabilities of previous models in solving the image inverse problem [2]. To this end, we implement the model proposed in Hussein et al (2020), which makes generators "image-adaptive" to a specific test sample. This model consists of three successive optimization stages: the non-image-adaptive "compressed sensing using generative models" (CSGM), the image-adaptive step (IA), and the post-processing "back-projection" (BP). Our results demonstrate that the two image-adaptive approaches—IA and BP—can effectively improve reconstructions. Further testing reveals slight biases existing in the model (e.g., skin tones), which we conjecture to be caused by the training dataset on which the model is trained. Finally, to explore more efficient ways of running the model, we test out different numbers of iterations used for CSGM. The results show that we can indeed decrease the number of CSGM iterations without compromising reconstruction qualities.

1 Introduction

An inverse problem consists of inferring parameters or data distributions of a system from its inadequate observations. Particularly, the goal of image inverse problems is to recover an image given its degraded or compressed observation. This task has a variety of applications, such as medical imaging and computer vision [9]. To address the problem, much work has explored the way for exploiting Generative Adversarial Networks (GAN) [4], which can be trained to produce desired data distributions (e.g., faces of humans) from random vectors in a latent space, to solve image inverse problems.

In 2017, Bora and his colleagues proposed "compressed sensing using generative models's" (CSGM) [2] in which they optimized the latent vectors by comparing observations and GAN-produced images. However, the prior methods, including CSGM, can only estimate images belonging to the class on which the model was trained; they suffered from the limited representation capabilities of GAN that could not capture the complexity of images, resulting in significant mismatches. Building on the method proposed in Bora et al (2017), Hussein proposed a new strategy in "Image-Adaptive GAN based Reconstruction" [5] to mitigate the limited representation capabilities of generative models. Under this strategy, known as Image Adaptation, they performed gradient-based optimization on the weights of a GAN to make their GAN image-adaptive at the inference time. In a noise-less scenario, they proposed further reconstructions that strictly enforce compliance of recoveries with their corresponding observations via back-projection (BP).

In this paper, we replicate "Image-Adaptive GANs" and evaluate the robustness of their method [5]. To find more efficient ways of running the model, we explore their method under potentially helpful usage adjustments—such as different numbers of iterations, changes in parameters, and different implementations of image compression. This paper describes the model architecture and the methods roughly sketched above in Section 2, presents the results of our replication in Section 3, and discusses our findings and ideas for further advancement in Section 4 and 5.

2 Image Reconstruction Approaches

The image inverse problem is defined more rigorously as such: given observation $y \in \mathbb{R}^m$ and $m \times n$ measurement matrix A (usually, $m \ll n$) satisfying

$$y = Ax + e, \tag{1}$$

where $x \in \mathbb{R}^n$ is an unknown vector and $e \in \mathbb{R}^m$ represents the noise, how can we reconstruct $\hat{x} \in \mathbb{R}^n$ that complies with x maximally? Hussein et al (2020) claim that such reconstruction can be obtained by undergoing three stages of compliance: CSGM, the image-adaptive step (IA), and BP. A key tool that will be used, as mentioned in the introduction, will be a pre-trained GAN. We discuss GANs and each of these stages in detail below.

2.1 Generative Adversarial Networks (GANs)

In 2014, Goodfellow and his colleagues proposed a new framework for estimating generative models through Generative Adversarial Networks (GANs) [4], in which they simultaneously train two "adversarial" deep-learning models: a generator G that takes in an arbitrary distribution from a latent space and produces desired data distributions, and a discriminator D that estimates the probability that a sample came from the training data rather than G. The two models compete with each other to get progressively better at their respective tasks.

The cost function for GANs uses the two-player minmax function $C(G, D)$ given by

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}(\log D(x)) + \mathbb{E}_{z \sim p_z(z)}(1 - D(G(z))), \tag{2}$$

where x is a sample from a training set and z is a random vector from a latent space. While the goal of D is to maximize V by making $D(x)$ as close to one as possible, the goal of G is to minimize V by making $G(z)$ as close to one as possible and in turn making $D(G(z))$ as close to one as possible. In other words, we train the discriminator to be able to accurately distinguish between the training data distribution and distributions produced by the generator, and make the generator better at recovering the training data distribution.

GANs can be trained to produce distributions of realistic images. The CSGM and IA methods described below take advantage of GANs' generative power and ensure realism in the reconstruction by performing gradient-based optimization to fine-tune the weights of pre-trained GANs.

2.2 Stage 1: CSGM

CSGM (compressed sensing using generative models) was originally proposed in Bora et al (2017). Compressed sensing is a method of signal reconstructing through solving under-

determined linear systems [3]. Despite its name, the method is capable of image reconstruction other than compressed sensing and works as follows. Consider $x \in \mathbb{R}^n$, A and $y \in \mathbb{R}^m$ as introduced earlier. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a differentiable function representing a generative model (i.e. a generator of a pre-trained GAN). We perform gradient-based optimization on z for the loss function

$$\text{loss}(z) = \|AG(z) - y\|_2^2 . \tag{3}$$

$AG(z)$ represents the degraded version of the generated image, and we seek to find $\hat{z}$ that minimizes the difference between this degraded image and the observation y.

2.3 Stage 2: Image Adaptation (IA)

The image adaptive stage (IA), proposed in Hussein et al (2020), uses the $\hat{z}$ initialized by the CSGM stage to further improve the reconstruction. Particularly, instead of only optimizing z by minimizing (3), we simultaneously optimize z and the weights of G—after initializing z to $\hat{z}$—denoted by θ. The new loss function is given by

$$\text{loss}(z, \theta) = \|AG_\theta(z) - y\|_2^2 . \tag{4}$$

CSGM uses a fixed generative model G, whose representation capability is insufficient in covering the entire class of a complex distribution (e.g. human faces), and thus has limited performance when used to reconstruct a sample not belonging to its training samples. The IA stage overcomes such limitation by optimizing θ to make G "adapt" to the specific observation y.

2.4 Stage 3: Back Projection (BP)

After obtaining a reconstructed image, $\hat{x}$, from CSGM and/or IA, back projection (BP) offers a way to further improve the reconstruction through post-processing. In this stage, we enforce compliance of the restoration with our observation y by computing

$$\hat{x}_{bp} = \underset{\tilde{x}}{\text{argmin}} \|\tilde{x} - \hat{x}\|_2^2 , \tag{5}$$

such that $A\tilde{x} = y$. This problem has a closed formula given by

$$\hat{x}_{bp} = A^\dagger(y - A\hat{x}) + \hat{x}, \tag{6}$$

where $A^\dagger := A^T(AA^T)^{-1}$ is the pseudo-inverse of A. Note that if y is noisy, this method is not effective as it amplifies the noise, which we discuss in Section 3.

3 Replication Experiments

We implement the model described in the previous section and apply it to three different types of image degradation. In our experiment, we use the official PyTorch implementation of Progressive Growing GANs (PGGANs) trained on the CelebA-HQ dataset to produce a realistic image of human faces of size 1024×1024 [7]. PGGANs use a progressive training method in which normalized convolutional layers are added throughout training to produce realistic images. The design of PGGANs allows for faster and more stable training by increasing complexity over time and is capable of producing higher-quality images, compared

to other types of GANs [7]. We optimize the loss function (3) and (4) using ADAM optimizer [8] with the learning rate of 0.1 for z optimization in CSGM and the learning rate of 0.0001 and 0.001 for z and θ optimization respectively for IA, as proposed in Hussein et al (2020).

Instead of running the model and averaging over 100 images, as done in (Hussein et al, 2020), we run our experiments over 50 only, and ensure the generalizability of the results by showing 95% confidence intervals. The three tasks and results of our model performance on them are described and shown in the following subsections.

3.1 Measuring Reconstruction Quality

In addition to visual results, we evaluate the performance of different methods (CSGM, CSGM followed by BP, IA, IA followed by BP) using two quantitative measures: PSNR (Peak Signal-to-Noise Ratio) [6] and PS (Perceptual Similarity) [11], as described below. The results are also compared against that of naive reconstruction obtained by $A^{\dagger}y$.

3.1.1 PSNR (Peak Signal-to-Noise Ratio)

The PSNR computes the peak signal-to-noise ratio by naively comparing reconstructed pixels to the actual pixels and taking the absolute difference (MSE). The mathematical representation of the PSNR is as follows:

$$\text{PSNR}(I, J) = 10 \log_{10} \left(\frac{\max(I)^2}{\text{MSE}(I, J)} \right), \tag{7}$$

where $\max(I)$ is equal to 255 (i.e. the maximum difference between two pixel values per channel). $\text{MSE}(I, J)$—Mean Squared Error—is the widely used perceptual metric, computed by

$$\text{MSE}(I, J) = \frac{1}{M \cdot N} \sum_{j=1}^{M} \sum_{i=1}^{N} \|I(i, j) - J(i, j)\|_2^2, \tag{8}$$

where I and J are images of size $N \times M$. Notice that when the pixel values of I and J are more similar, $\text{MSE}(I, J)$ is smaller, which in turn makes our PSNR value bigger. PSNR is a means of measuring the performance of our methods. However, it is too simplistic for capturing nuances of human perception and does not account for whether the differences are perceptually salient to humans. There is thus a need for a more sophisticated quantitative measure that can account for the semantic meanings of images.

3.1.2 PS (Perceptual Similarity)

The PS measures the similarity between two images in a way that better resembles human perception [11]. It is the output of a deep learning model trained using "perceptual loss" functions that measure the similarity of two images. This PS metric is shown to outperform all other metrics, such as PSNR, in the task of image differentiation. PS models can be trained using both supervised and unsupervised learning. While Hussein et al (2020) do not specify which PS model they use, models trained with supervised learning are advantageous. Hence, we use one of the supervised models, AlexNet (supervised), in our experiments [11].

3.2 Super Resolution: Bicubic Interpolation

The task of super-resolution is characterized by reconstructing images that have undergone a down-sampling process that maps a sample area from the original image to each pixel in the resulting images, lowering the resolution. We use Bicubic Interpolation (its official implementation in PyTorch) to down-sample images before attempting super-resolution. Bicubic Interpolation is a method that maps 4×4 pixel grids in the original image to a single pixel in the image output, whose convolutional kernel is typically given by

$$W(x) = \begin{cases} 1.5|x|^3 - 2.5|x|^3 + 1 \text{ for } |x| \leq 1, \\ 0.5|x|^3 - 2.5|x|^2 + 4|x| - 2 \text{ for } 1 < |x| < 2, \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

For super-resolution using bicubic interpolation, our $A^\dagger$ (i.e. naive reconstruction) is given by bicubic up-sampling. The results are shown in Table 1 and 2.

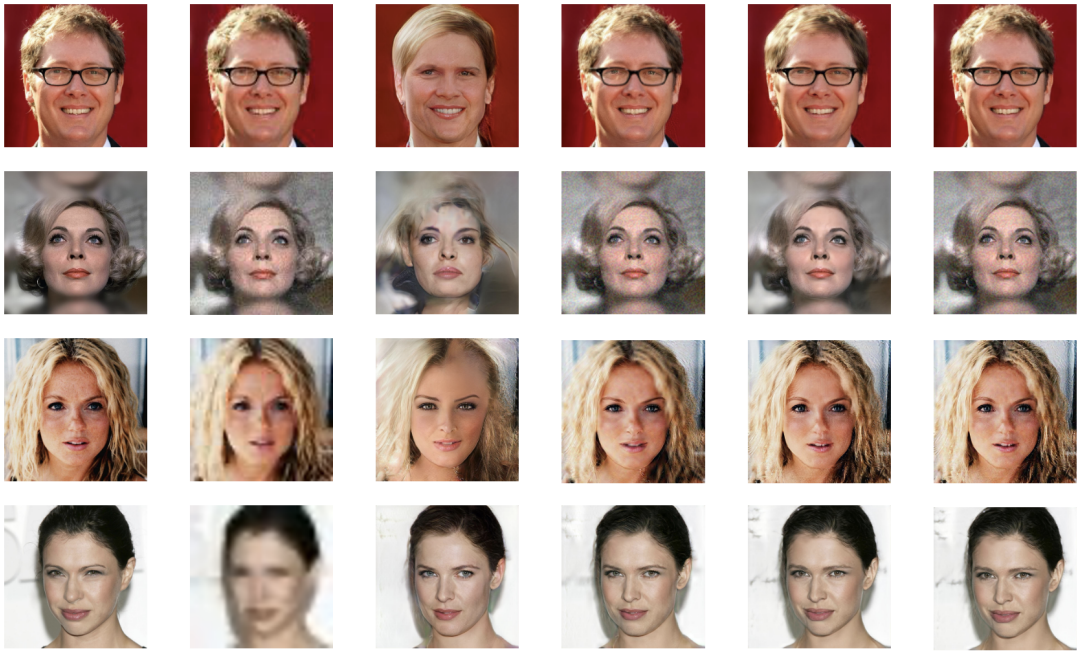

Figure 1: Super-resolution with bicubic kernel. From top to bottom: scale factor 8 with noise level of 0/255, scale factor 8 with noise level of 40/255, scale factor 16 with noise level of 10/255, scale factor 32 with noise level of 0/255. From left to right: original image (x), degraded image (y), bicubic up-sampling, CSGM, CSGM-BP, IA, and IA-BP. Note that CSGM and IAGAN use the pre-trained PGGAN.

Table 1: Results for super-resolution with bicubic interpolation. Reconstruction PSNR averaged over 50 images from CelebA-HQ–higher values are better than lower values of PSNR. Images are generated from PGGAN with scale factors 8, 16, and 32, and noise levels 0%, 10%, and 40%. Errors reflect 95% confidence interval of resulting values.

Super Resolution (Bicubic Interpolation) with PSNR						
Scale	Noise	Naive	CSGM	CSGM-BP	IA	IA-BP
8	0%	**29.91** ± 0.84	21.24 ± 0.50	28.07 ± 0.79	27.88 ± 0.76	29.39 ± 0.85
8	10%	**28.05** ± 0.56	21.05 ± 0.49	26.64 ± 0.60	26.59 ± 0.60	27.49 ± 0.60
8	40%	20.54 ± 0.12	18.76 ± 0.26	20.27 ± 0.12	**20.73** ± 0.17	20.35 ± 0.13
16	0%	**26.93** ± 0.67	21.18 ± 0.49	25.25 ± 0.62	25.58 ± 0.74	25.89 ± 0.76
16	10%	**25.92** ± 0.37	20.99 ± 0.37	24.52 ± 0.39	24.77 ± 0.46	24.91 ± 0.46
32	0%	**23.83** ± 0.52	20.77 ± 0.47	22.70 ± 0.52	23.08 ± 0.61	23.13 ± 0.62

Setting aside the fact that naive reconstruction (bicubic up-sampling) appears to yield the best results, it is noticeable that as we progress through the stages of our GAN based reconstruction, the PSNR evaluation generally improves. The results of CSGM are not satisfying both in Figure 1 and Table 1. The PSNR values for CSGM are significantly lower than those of other methods, and there are significant mismatches between the original image and the CSGM reconstruction. The BP step appears very effective in reducing errors and improving results from CSGM. Compared to CSGM, IA yields much better result across all scale factors and noise levels, showing the effectiveness of image adaptation. The post-processing BP improves the IA results slightly when noise is relatively low (i.e. 0% and 10%), and does not when noise is higher (i.e. 40%), as predicted. 95% confidence intervals ensure that CSGM performs reliably worse than naive reconstruction, while IA-BP is only slightly worse. Naive reconstruction, interestingly, does not yield the most visually superior results despite having high (at most times the highest) PSNR score. This motivates us to view the more sophisticated PS ratings of the corresponding results.

Table 2: Results for super-resolution with bicubic interpolation. Reconstruction PS (AlexNet) averaged over 50 images from CelebA-HQ–lower values are better than higher values of PS. Images are generated from PGGAN with scale factors 8, 16, and 32, and noise levels 0%, 10%, and 40%. Errors reflect 95% confidence interval of resulting values.

Super Resolution (Bicubic Interpolation) with PS (AlexNet)						
Scale	Noise	Naive	CSGM	CSGM-BP	IA	IA-BP
8	0%	0.331 ± 0.03	0.349 ± 0.03	0.282 ± 0.02	0.266 ± 0.02	**0.242** ± 0.02
8	10%	0.352 ± 0.03	0.341 ± 0.03	0.299 ± 0.026	**0.262** ± 0.02	0.264 ± 0.02
8	40%	0.526 ± 0.02	0.359 ± 0.03	0.496 ± 0.02	**0.279** ± 0.02	0.467 ± 0.016
16	0%	0.410 ± 0.03	0.353 ± 0.03	0.318 ± 0.02	0.286 ± 0.02	**0.281** ± 0.02
16	10%	0.439 ± 0.02	0.353 ± 0.02	0.339 ± 0.02	**0.290** ± 0.02	0.305 ± 0.02
32	0%	0.447 ± 0.03	0.345 ± 0.02	0.328 ± 0.02	0.310 ± 0.02	**0.308** ± 0.02

As shown in Table 2, IA and IA-BP outperform both CSGM and naive reconstructions when measured by perceptual similarity, and for larger scales, CSGM also outperforms naive reconstruction under this measurement. In particular, naive reconstruction is no longer superior to the rest, which matches our visual evaluation, reaffirming the idea that PS is a more accurate (i.e., closer to human perception) measurement than PSNR.

It should be noted that we have smaller (i.e. better) PS results than Hussein et al (2020), possibly due to differences in implementations of PS: our implementation uses the supervised Alex-NET, discussed in Section 3.1.2, while Hussein et al (2020) do not specify

their implementation. Despite the difference, the trends in the two tables above match results collected in Hussein et al (2020), showing the robustness of their method in the task of super-resolution.

3.3 Compressed Sensing: Fast Fourier Transformation (FFT)

Compressed Sensing is a signal process of reconstructing a signal from its compressed form in the frequency domain. Compression, on the other hand, is about making a given signal less expensive to store—lessening the amount of information stored. Our experiment uses a compression algorithm that involves the fast Fourier transform (FFT) described as follows.

Given the original signal (i.e. image) x in the time domain, we apply the fast Fourier transform on it, obtaining x' that lives in the frequency domain. We then apply a random mask on x' that randomly zeroes out a certain percentage (i.e. $1-$compression rate) of its elements to obtain compression y. As hinted, y is a vector in the frequency domain consisting of complex numbers. When minimizing the loss function (3), we store y and $AG(z)$ as multi-variable real vectors. Having y in the frequency domain also provides us with a convenient pseudo-inverse $A^{\dagger}$: the inverse fast Fourier transform (IFFT), which projects a vector in the frequency domain back to its counterpart in the time domain.

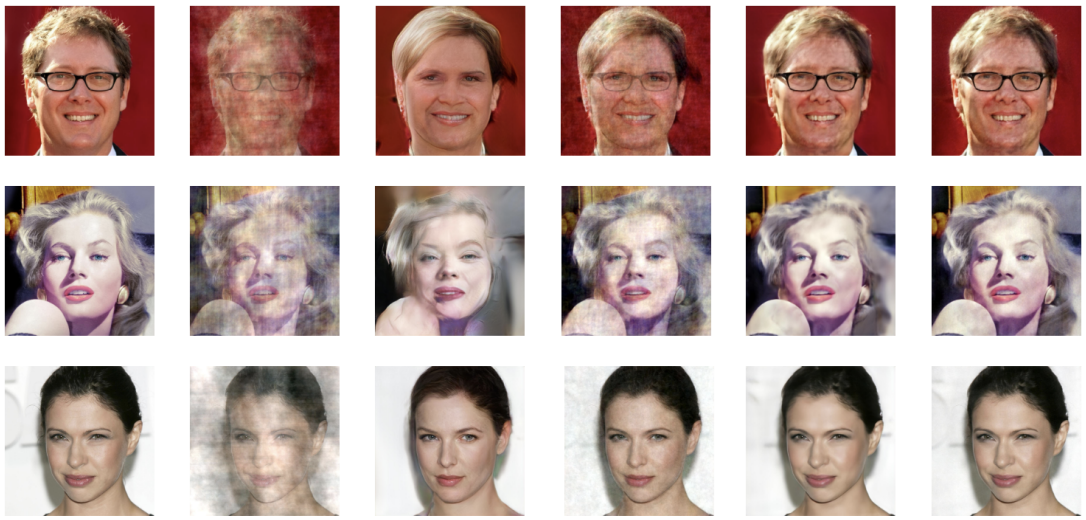

Figure 2: Compressed Sensing with FFT. From top to bottom: compression rate 0.3 with noise level of 0/255, compression rate 0.5 with noise level of 0/255, compression rate 0.5 with noise level of 10/255. From left to right: original image (x), IFFT image, CSGM, CSGM-BP, IA, and IA-BP. Note that CSGM and IAGAN use the pre-trained PGGAN.

Table 3: Results for compressed sensing with the FFT compression. Reconstruction PSNR averaged over 50 images from CelebA-HQ. Images are generated from PGGAN with compression ratios 30% and 50%, and noise levels 0% and 10%. Errors reflect 95% confidence interval of resulting values.

Compressed Sensing (FFT) with PSNR						
Ratio	Noise	IFFT	CSGM	CSGM-BP	IA	IA-BP
0.3	0%	14.35 ± 0.55	18.608 ± 0.76	19.73 ± 0.87	21.85 ± 1.10	**22.30** ± 1.17
0.3	10%	14.58 ± 0.48	18.97 ± 0.59	20.09 ± 0.65	22.46 ± 0.99	**22.76** ± 1.00
0.5	0%	16.92 ± 0.54	20.45 ± 0.56	23.54 ± 0.68	26.02 ± 0.92	**27.83** ± 1.10
0.5	10%	16.59 ± 0.58	20.14 ± 0.58	22.82 ± 0.70	25.83 ± 0.98	**26.73** ± 0.95

Table 4: Results for compressed sensing with the FFT compression. Reconstruction PS averaged over 50 images from CelebA-HQ. Images are generated from PGGAN with compression ratios 30% and 50%, and noise levels 0% and 10%. Errors reflect 95% confidence interval of resulting values.

Compressed Sensing (FFT) with PS (Alex)						
Ratio	Noise	IFFT	CSGM	CSGM-BP	IA	IA-BP
0.3	0%	0.355 ± 0.017	0.375 ± 0.028	0.321 ± 0.022	0.317 ± 0.030	**0.238** ± 0.024
0.3	10%	0.436 ± 0.012	0.365 ± 0.024	0.377 ± 0.016	**0.307** ± 0.026	0.310 ± 0.018
0.5	0%	0.260 ± 0.014	0.346 ± 0.025	0.233 ± 0.017	0.277 ± 0.026	**0.145** ± 0.018
0.5	10%	0.358 ± 0.010	0.350 ± 0.025	0.318 ± 0.012	0.280 ± 0.027	**0.238** ± 0.012

The numerical results collected for compressed sensing in Table 3 and 4 match Hussein et al (2020) in patterns and trends, and also match our expectations. The effect of noise on the back projection step is also clear. From results in the first two rows—compressed sensing with a compression ratio of 0.3, we notice that BP worsens the reconstruction when noise is present (for both CSGM and IA). This is not exactly the case for the latter two rows (with a compression ratio of 0.5), but the extent to which BP improves the reconstruction differs depending on the presence of noise: without noise, the PS rating from IA to IA-BP is decreased from 0.277 to 0.145, while with 10% noise, the change is from 0.280 to 0.238.

3.4 Deblurring: Gaussian Blur

Deblurring is the process that recovers a sharp image from its blurred state. The original image x is convoluted with a blur kernel of a specific size to generate our observation y. We examine the scenario in which we applied the blurring operator A of Gaussian kernel of size 9×9 (its official implementation in PyTorch). In this case, there is no efficient way to implement the pseudo-inverse $A^{\dagger}$. Hence, we do not examine the BP post-processing. The results are shown below in Figure 3 and Table 5 and 6.

Figure 3: Deblurring with Gaussian Blur with 9×9 kernel. From top to bottom: noise level of 0/255 and noise level of 10/255. From left to right: original image (x), CSGM, and IA. Note that CSGM and IAGAN use the pre-trained PGGAN.

Table 5: Results for Deblurring with Gaussian Kernel. Reconstruction PSNR averaged over 50 images from CelebA-HQ. Images are generated from PGGAN with kernels of size 9 and noise levels 0% and 10%. Errors reflect 95% confidence interval of resulting values.

Deblurring (Gaussian Kernel) with PSNR			
Kernel	Noise	CSGM	IA
9	0%	21.55 ± 0.48	**28.78** ± 0.74
9	10%	21.37 ± 0.45	**27.63** ± 0.59

Table 6: Results for Deblurring with Gaussian Kernel. Reconstruction PS (AlexNet) averaged over 50 images from CelebA-HQ. Images are generated from PGGAN with kernels of size 9 and noise levels 0% and 10%. Errors reflect 95% confidence interval of resulting values.

Deblurring (Gaussian Kernel) with PS			
Kernel	Noise	CSGM	IA
9	0%	0.348 ± 0.02	**0.282** ± 0.03
9	10%	0.342 ± 0.03	**0.281** ± 0.03

We see that the IA method unmistakably outperforms CSGM. Note that, interestingly, the average perceptual similarity of reconstructed images with higher noise levels is better than that with 0% noise, suggesting that the GAN-based optimization stages are not necessarily negatively affected by low noise levels in the task of deblurring.

4 Robustness

In addition to testing and verifying the method proposed in Hussein et al (2020) through replication, we further examine the method's robustness by assessing its performance on different facial features, and with an adjusted number of iterations for CSGM. By running our model on different facial features, we uncover potential biases residing within the model. By testing different numbers of CSGM iterations, we seek to strike a balance between time efficiency and reconstruction quality.

4.1 Evaluating Performance on Different Facial Features

We divide the images from CelebA-HQ into subsets (each containing 10 images) based on six preselected features: the presence of objects such as glasses and ornaments, dark and light hair colors, the absence of hair, and dark and light skin colors. This allows us to measure the effect of the innate biases residing within the training dataset, CelebA-HQ, on which PGGAN is trained. We focus on the task of super-resolution with bicubic interpolation with scale factor 16 with no noise and the PS results of the reconstructions. We compare the result of these features against that of the general test samples. The result is shown below in Table 7 and Figure 4.

Table 7: Results for super-resolution with bicubic interpolation. Reconstruction PS (AlexNet) averaged over 10 images with particular characteristics (General, Objects, Dark Hair, Light Hair, No Hair, Dark Skin, and Light Skin) from CelebA-HQ. Images are generated from PGGAN with scale factors 16 and noise levels 0%. Errors reflect 95% confidence interval of resulting values.

Super Resolution (Bicubic Interpolation) with PS (AlexNet)				
Group	CSGM	CSGM-BP	IA	IA-BP
General	0.353 ± 0.03	0.318 ± 0.02	0.286 ± 0.02	0.281 ± 0.02
Object	0.327 ± 0.03	0.290 ± 0.03	0.260 ± 0.03	0.254 ± 0.03
Dark Hair	0.333 ± 0.05	0.304 ± 0.05	0.266 ± 0.04	0.262 ± 0.04
Light Hair	0.329 ± 0.06	0.288 ± 0.06	0.273 ± 0.05	0.267 ± 0.05
No Hair	0.324 ± 0.08	0.284 ± 0.08	0.268 ± 0.08	0.262 ± 0.08
Dark Skin	0.374 ± 0.08	0.347 ± 0.08	0.312 ± 0.07	0.309 ± 0.07
Light Skin	0.343 ± 0.07	0.307 ± 0.07	0.282 ± 0.06	0.276 ± 0.06

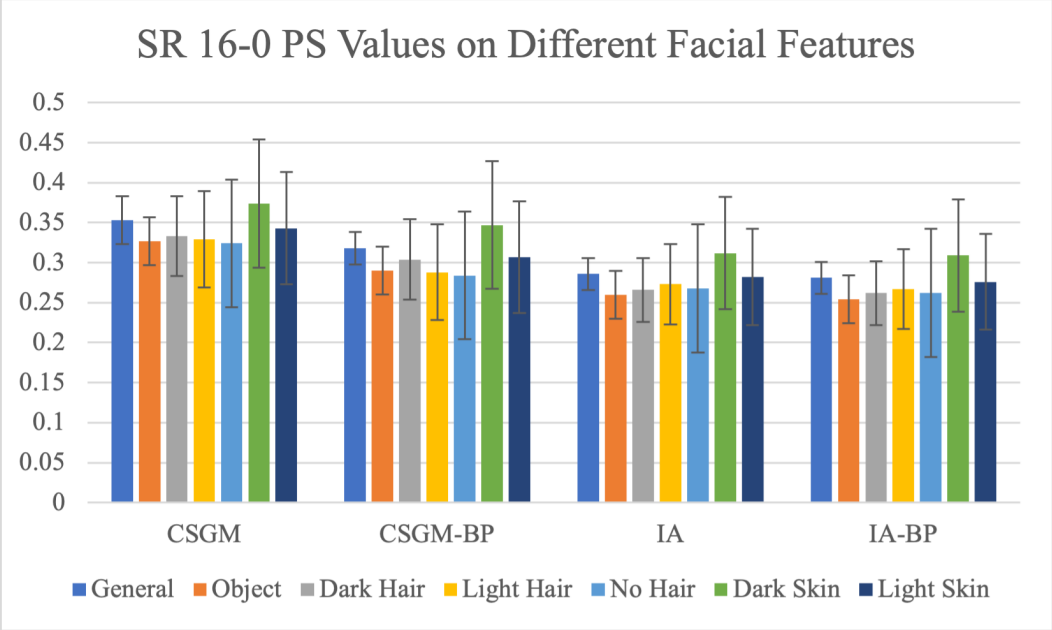

Figure 4: Graph representing the results for super-resolution with bicubic interpolation. Reconstruction PS (AlexNet) averaged over 10 images with particular characteristics (General, Objects, Dark Hair, Light Hair, No Hair, Dark Skin, and Light Skin) from CelebA-HQ. Images are generated from PGGAN with scale factors 16 and noise levels 0%. Errors reflect 95% confidence interval of resulting values.

There are noticeable, albeit slight biases present in our model, particularly between dark and light skin colors. We conjecture, however, that such bias can be removed by updating the dataset on which the PGGAN is trained. Although the presence of hair appears to introduce more complexity to the reconstruction and therefore should yield worse reconstructions, our data shows that there is no evidence supporting this hypothesis.

4.2 Finding "Optimal" Ratio of CSGM in IA

As ensured in previous sections, reconstruction can be obtained by undergoing three stages of compliance: CSGM, IA, and BP. While Hussein et al (2020) do clearly state the number of iterations used for CSGM and IA in the various tasks in their experiments, it offers no direct discussion of what ratio, or numerical interval, is "optimal" for the number of iterations of the CSGM and IA stage, taking into account both efficiency in time and quality of reconstruction. In other words, we question how much z initialization (i.e., CSGM) is necessary in the presence of the joint optimization on z and the weight of PGGAN (i.e., IA). To this end, we observe the final performance of our model (reconstructions after the three stages of compliance) under varying numbers of iterations of CSGM relative to IA.

Table 8: Results for super-resolution with bicubic interpolation. 95% confidence intervals for PSNR and PS (AlexNet) of reconstruction averaged over 50 images from CelebA-HQ. Images are generated from PGGAN with scale factor 8 and noise level 0%.

Super Resolution (Bicubic Interpolation)			
CSGM iteration	IA iteration	PSNR (after IA)	PS (after IA)
0	300	29.03 ± 0.68	0.276 ± 0.02
900	300	29.31 ± 0.81	**0.237**± 0.02
1800	300	**29.41** ± 0.81	0.242 ± 0.02

Our results show that the 1800-300 combination outperforms 0-300, suggesting that z initialization (CSGM) is necessary to some extent. However, the 1800-300 result is comparable to that of 900-300, which implies that considering time efficiency, the number of iterations for CSGM could perhaps be reduced without compromising the quality of reconstructions. After running pairwise permutation tests comparing the mean difference in PS between the 900-300 and 1800-300 combinations, we are 95% confident that the PS for the 1800-300 is between -0.0195 to 0.00988 greater than that of the 900-300 combination for any given image on average. This suggests any difference between the two is quite small. Likewise, assuming there is no difference in performance between the 900-300 and 1800-300 combinations, there is roughly a 52.2% chance of measuring a mean difference in performance between the models as extreme or more than what we observed. This suggests that in some cases, we do not have evidence that adding more iterations of CSGM will achieve performance that is practically discernible from models with fewer iterations of CSGM.

5 Discussion

Our work examines the image-adaptive generative model, proposed in Hussein et al (2020), that purports to mitigate problems caused by the limited representation capabilities of previous models in solving the image inverse problem. To this end, we implement the model proposed in Hussein et al (2020) and evaluate the robustness of their methodologies.

Our results in Section 3 show that the two approaches, IA and BP, can effectively improve reconstructions. Particularly, the image adaptation technique (i.e., IA) is very effective in enhancing the generator's capability to estimate the specific test sample. Discussion in Section 4.1 reveals slight biases in the model (e.g. reconstruction of faces with darker skin tone is worse) that cannot be adjusted by the IA step. We conjecture that the biases are caused by the training dataset for the PGGAN and thus able to be ameliorated by updating the dataset. Finally, to explore more efficient ways of running the model, we tested out our conjecture that the number of iterations used for CSGM can be lowered, and find that it indeed can be lowered to an extent without sacrificing model performance. However, future work can further explore the optimal number of iterations for IA. We conjecture that performing too many IA iterations will produce worse reconstructions, due to "overfitting" of the model, and hence hinder its ability to generate a realistic image of human faces.

Questions remain regarding how to further generalize the reconstruction model. In our work, the reconstruction focuses on a particular class of image data: human faces, but the method can reasonably be applied to the reconstruction of new classes of image data, and, by extension, the reconstruction of other types of signal, such as degraded audio signals. Exploring in such directions may yield valuable insights into the generality of this optimization strategy. Further insights could also be gained by utilizing CSGM, IA, and BP on different types of generative models such as boundary equilibrium GANs (BEGANs) [1] and recurrent GANs (RGANs) [10].

Moreover, although images generated through these techniques are visually appealing, individual images still require a timescale of minutes—roughly 55 seconds per image on Google Colab's High Access RAM Persistence-M GPUs—to be produced. If real-time image reconstruction is to be performed on video data, much work remains at improving the speed at inference time.

Furthermore, understanding the harmful ways this technology might be used could mitigate its potential negative impact. For instance, image reconstruction GANs that use this technique to enhance security camera footage and identify criminals might show bias, incorrectly reconstruct faces, and lead to false imprisonment. Conversely, if this technology continues to improve in accuracy, oppressive governments might be able to accurately target and harass peaceful protesters caught on video. Knowing this, researchers could respond by clearly stating the limitations of the models they release to the public as well as employers and restricting access to powerful models via APIs that can refuse to perform potentially harmful reconstructions. Thus, conducting future studies on generalizing to new data and models, improving speed and performance, and potential drawbacks of these optimization techniques could improve our overall understanding.

6 Acknowledgement

We would like to thank Professor Anna Rafferty for supervising our project. Without her guidance, expertise, and patience, our project could never proceed as it did, and certainly would not be nearly as fruitful.

References

[1] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[2] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[3] Yaniv Erlich. Cs: Compressed genotyping, dna sudoku - harnessing high throughput sequencing for multiplexed specimen analysis. In *Paris Machine Learning*, pages 1–1. Nuit Blanche community, 2009.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[5] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Image-adaptive gan based reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3121–3129, 2020.

[6] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[7] 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.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Otmar Scherzer. Scale-space methods and regularization for denoising and inverse problems. *Advances in imaging and electron physics*, 128:446–530, 2003.

[10] Yuqiang Sun, Lei Peng, Huiyun Li, and Min Sun. Exploration on spatiotemporal data repairing of parking lots based on recurrent gans. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 467–472. IEEE, 2018.

[11] 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.