Intermediate Layer Optimization for Inverse Problems using Deep Generative Models

Giannis Daras [*1] **Joseph Dean** [*1] **Ajil Jalal** [1] **Alexandros G. Dimakis** [1]

Abstract

We propose **I**ntermediate **L**ayer **O**ptimization (**ILO**), a novel optimization algorithm for solving inverse problems with deep generative models. Instead of optimizing only over the initial latent code, we progressively change the input layer obtaining successively more expressive generators. To explore the higher dimensional spaces, our method searches for latent codes that lie within a small l_1 ball around the manifold induced by the previous layer. Our theoretical analysis shows that by keeping the radius of the ball relatively small, we can improve the established error bound for compressed sensing with deep generative models. We empirically show that our approach outperforms state-of-the-art methods introduced in StyleGAN-2 and PULSE for a wide range of inverse problems including inpainting, denoising, super-resolution and compressed sensing.

1. Introduction

We study how deep generators can be used as priors to solve inverse problems like inpainting, super-resolution, denoising and compressed sensing from random projections. Image reconstruction methods can be either supervised (Pathak et al., 2016; Richardson et al., 2020; Yu et al., 2018) or unsupervised (Menon et al., 2020; Bora et al., 2017; Pajot et al., 2019), see the recent survey (Ongie et al., 2020) for a unified presentation. Such inverse problems naturally appear in many applications including medical imaging, single pixel reconstruction and other domains (Lustig et al., 2007; 2008; Chen et al., 2008; Duarte et al., 2008; Qaisar et al., 2013; Hegde et al., 2009).

We focus on unsupervised image reconstruction techniques that rely on a pre-trained generator, building on the general framework introduced in CSGM (Bora et al., 2017). The central optimization problem that appears in unsupervised image reconstruction is the inversion of a deep generative model, i.e. finding a latent code that explains the measurements. This can be performed for different generators, e.g. DCGAN or more recently the powerful StyleGAN-2 (Karras et al., 2019; 2020) as shown in the excellent results obtained by PULSE (Menon et al., 2020). Unfortunately, inverting a generator with even 4 layers is NP-hard (Lei et al., 2019) so approximate inversion methods are needed.

The CSGM framework (Bora et al., 2017) used gradient descent to minimize the measurement mean squared error (MSE) and showed good empirical performance for numerous inverse problems including inpainting and compressed sensing with random Gaussian measurements using DCGAN. However, this *does not* work as well for deeper generators e.g. BigGAN as discussed in Daras et al. (2020). PULSE (Menon et al., 2020) improved the CSGM framework focusing specifically on super-resolution, by refining the latent space optimization and using the StyleGAN-2 (Karras et al., 2019; 2020) generator.

We propose a novel optimization method for solving general inverse problems using a technique we call **Intermediate Layer Optimization** (ILO). Our method adaptively changes which layer is optimized, moving from the initial latent code to intermediate layers closer to the pixels. By optimizing intermediate layers we expand the range of the generator to better satisfy the measurements. This has to be done carefully since intermediate layers can produce non-realistic images and therefore inversion must be regularized.

1.1. Our Contributions

1. We propose a novel optimization method for solving general inverse problems by adaptively changing which layer variables are optimized. Our method extends PULSE (Menon et al., 2020) beyond super-resolution, to all inverse problems with differentiable forward operators.
2. To avoid over-expanding the range of the generator to non-realistic images, we only search for latent codes within a small l_1 ball around the manifold induced by the previous layer. Conceptually, our method generalizes the framework introduced in Dhar et al. (2018); instead of allowing sparse

[*]Equal contribution [1]The University of Texas at Austin. Correspondence to: Giannis Daras <giannisdaras@utexas.edu>, Joseph Dean <josephdean98@utexas.edu>, Ajil Jalal <ajiljalal@utexas.edu>, Alexandros G. Dimakis <dimakis@austin.utexas.edu>.

deviations only in the image space, we allow small deviations from the manifold of any layer of the generator.
3. We theoretical analyze our framework by establishing sample complexity and error bounds. We show that by restricting the radius of the latent searches, we can improve the established error bound of CSGM (Bora et al., 2017).
4. Experimentally, our method significantly outperforms the previous state-of-the-art techniques for solving inverse problems with deep generative models for a wide range of tasks including inpainting, denoising and super-resolution.
5. To illustrate the power of inverse problems with general differentiable forward operators, we use a classifier as a measurement process. Specifically, we show how we can use a classifier to bias generators to produce human images that look like ImageNet classes like frogs, corals and goldfishes. Our method uses gradients from classifiers trained to achieve robustness to adversarial attacks as proposed in (Santurkar et al., 2019), but guiding generative latent codes as opposed to pixels directly.
6. We open-source all our code to encourage further research in this area: https://github.com/giannisdaras/ilo. A demo of our code is available under this URL.

2. Algorithm

2.1. Setting

The key step in our approach is to decompose pre-trained generative models as compositions of feed-forward neural networks. Given a (pre-trained) generative model $G(z) \in \mathbb{R}^n$ that produces images from latent codes $z \in \mathbb{R}^k$, we decompose it as a $G = G_2 \circ G_1$ where $G_1 : \mathbb{R}^k \to \mathbb{R}^p$ and $G_2 : \mathbb{R}^p \to \mathbb{R}^n$. As usual, the latent vectors $z^k \in \mathbb{R}^k$ were sampled according to a simple distribution P_z, typically Gaussian and independent.

Our observations are formed by a known measurement matrix

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

where $A : \mathbb{R}^{m \times n}$ where $x \in \mathbb{R}^n$ is the real image we want to recover. We emphasize that our algorithm can be applied when the measurement process is a general differentiable operator $y = \mathcal{A}(x)$ but our theory only applies to linear inverse problems. Since we will be working with latent vectors in different layers we indicate the dimension as a superscript, so z^k denotes an initial latent vector in $\mathbb{R}^k$ and z^p an intermediate vector in $\mathbb{R}^p$.

2.2. Approach

Our approach is described in Algorithm 1. The first step of our method is the same as in CSGM (Bora et al., 2017); we optimize over a k-dimensional latent code, z^k, which is the input of the first layer of the generator. In practice, to obtain the solution of line 1 of Algorithm 1, we pick an initial z^k from the latent distribution of the generator and we optimize the loss function $||AG(z^k) - Ax||$ using gradient descent. Once we solve this optimization problem, we obtain a solution, $\hat{z}^k$, that we map to the p-dimensional space using G_1. By doing that, we get an intermediate latent representation, $\hat{z}^p = G_1(\hat{z}^k)$.

From that point onwards, our algorithm proceeds in rounds. At the beginning of each round, we optimize on the p-dimensional input space of G_2 but we only allow solutions that lie within an l_1 ball centered at $\hat{z}^p$. Intuitively, we allow deviations from the range of G_1 to increase the expressitivity of the model, but we restrict those deviations to avoid overfitting on the measurements (see Experiments section).

Once we obtain the solution of line 4 of Algorithm 1, i.e. once we find the latent code, $\tilde{z}^p$, that best explains the measurements and lies inside an l_1 ball of the previous latent, we project this solution back to the range of the generator. To do that, we search for the latent code z^k such that $G_1(z^k)$ is as close as possible to $\tilde{z}^p$ (line 5 of Algorithm 1). This problem is solved by initializing a latent vector z^p to $\hat{z}^p$ and then minimizing using gradient descent the loss $||G_1(z^k) - \tilde{z}^p||$. The solution of this problem forms a new $\hat{z}^k$ vector which is in turn projected again to the intermediate code $\hat{z}^p = G_1(\hat{z}^k)$. Our algorithm attempts to explore the set we call the *extended range:* the range of vectors realizable by the previous layer, dilated by an l_1 ball of sparse deviations. Within this set we would like to find the latent vector that best explains the measurements.

We emphasize that our theoretical analysis provides performance bounds for the global optimum in this extended range, while our algorithm is based on projected gradient descent for a non-convex problem and therefore can be stuck in local optima. It may be possible to prove that such local optimization algorithms obtain global minima under generator weight assumptions as achieved in the pioneering work of Hand & Voroninski (2018); Hand et al. (2018) for CSGM, but this remains open for future work.

3. Theoretical Analysis

3.1. Preliminaries

We begin our theoretical discussion by revisiting some important elements of the theory of compressed sensing with deep generative models.

Definition 1 (S-REC (Bora et al., 2017))**.** *Let $S \subseteq \mathbb{R}^n$. For some parameters $\gamma, \delta > 0$, a matrix $A \in \mathbb{R}^{m \times n}$ is said to satisfy S-REC(S, γ, δ) if $\forall x_1, x_2 \in S$, we have that:*

$$||A(x_1 - x_2)||_2 \geq \gamma ||x_1 - x_2||_2 - \delta. \tag{2}$$

The S-REC condition, introduced in CSGM (Bora et al.,

Figure 1. Results on the inpainting task. Rows 1, 2, 3 and 5 are real images (outside of the test set, collected from the web) while rows 4, 6 are StyleGAN-2 generated images. Column 2: the first five images have masks that were chosen to remove important facial features. The last row is an example of randomized inpainting, i.e. a random 1% of the total pixels is observed. Columns 3-5: reconstructions using the CSGM (Bora et al., 2017) algorithm with the StyleGAN-2 generator and the optimization setting described in PULSE (Menon et al., 2020). While PULSE only applies to super-resolution, we extend it using MSE, LPIPS and jointly MSE+LPIPS loss. The experiments of Columns 3-5 form an ablation study of the benefits of each loss function. Column 6: reconstructions with ILO (ours). As shown, ILO consistently gives better reconstructions of the original image. Also, many biased reconstructions can be corrected by our method. In the last two rows, recovery of the image is still possible from very few pixel observations using our method.

2017), guarantees that if two vectors, $x_1, x_2 \in \mathbb{R}^n$, are very different (right side of the equation), then their measurements will be significantly different as well (left side of the equation). In CSGM, the set S of interest is the range of the generator. Therefore, S-REC is a key property for proving small reconstruction error when observing Ax. Bora et al. (2017) show that if 1) A is a matrix with i.i.d. Gaussian entries drawn from $\mathcal{N}(0, \frac{1}{m})$ and 2) $m = \frac{1}{a^2}\Omega\left(k \log\left(\frac{L_1 \cdot L_2 r_1}{\delta}\right)\right)$, then with probability $1 - e^{-a^2\Omega(m)}$, S-REC$(G_2(G_1(B_1^k(r_1))), 1 - a, \delta))$ is satisfied.

3.2. Intermediate Layer Optimization

Our theoretical result is a sample complexity bound for the reconstruction algorithm that optimizes in the full extended range of the generative model, similar in style to the CSGM (Bora et al., 2017) result.

Let $B_q^k(r_1)$ denote a ball of radius r_1 measured in l_q norm and $\oplus$ denote the Minkowski sum operation, i.e. given sets S_1, S_2, the set

$$S_1 \oplus S_2 = \{x + y | x \in S_1, y \in S_2\}.$$

If the initial vector z^k lies in a ball of radius r_1, denoted as $B_2^k(r_1)$, the range of the first generator is $G_1(B_2^k(r_1))$. We

Algorithm 1 ILO for one layer of the generator

```
// CSGM solution
ẑ^k ← argmin_{z^k ∈ B_2^k(r_1)} ||AG(z^k) − Ax||_2
ẑ^p ← G_1(ẑ^k)
for t ← 0 to r do
    // Best solution within an l_1 ball
    centered around the prev. solution
    z̃^p ← argmin_{z^p ∈ ẑ^p ⊕ B_1^p(r_2)} ||AG_2(z^p) − Ax||
    // Projection back to the range
    ẑ^k ← argmin_{z^k ∈ B_2^k(r_1)} ||G_1(z^k) − z̃^p||
    ẑ^p ← G_1(ẑ^k)
end
// Return the best solution within an l_1
ball of some point in the range
return G_2(z̃^p)
```

are expanding this set to create the *extended range*:

$$G_1(B_2^k(r_1)) \oplus B_1^p(r_2).$$

Our result is showing that minimizing the measurements in this extended range gives a reconstruction that is close to the best reconstruction that the extended generator G_2 can produce. This result is obtained with high probability over the random measurement matrix A, if the number of measurements is sufficiently large:

Theorem 1. *Let $G = G_2 \circ G_1$ with $G_1 : \mathbb{R}^k \to \mathbb{R}^p$ be an L_1-Lipschitz function and $G_2 : \mathbb{R}^p \to \mathbb{R}^n$ be an L_2-Lipschitz function. Let $A \in R^{m \times n}$ be the measurements matrix with $A_{ij} \sim \mathcal{N}(0, 1/m)$ i.i.d. entries.*

Let K be a parameter of our choice where $K \leq \sqrt{p}$, and $r_2 = \frac{K\delta}{L_2}$. Consider the true optimum in the extended range

$$\bar{z}^p = \text{argmin}_{z^p \in G_1(B_2^k(r_1)) \oplus B_1^p(r_2)} ||x - G_2(z^p)||, \quad (3)$$

and the measurements optimum in the extended range

$$\tilde{z}^p = \text{argmin}_{z^p \in G_1(B_2^k(r_1)) \oplus B_1^p(r_2)} ||Ax - AG_2(z^p)||. \quad (4)$$

Then, if the number of measurements is sufficiently large:

$$m = \frac{1}{(1-\gamma)^2} \Omega\left(k \log \frac{L_1 L_2 r_1}{\delta} + K^2 \log p\right), \quad (5)$$

then with probability at least $1 - e^{-\Omega((1-\gamma)^2 \cdot m)}$, we have the following error bound:

$$||x - G_2(\tilde{z}^p)|| \leq \left(1 + \frac{4}{\gamma}\right) ||x - G_2(\bar{z}^p)|| + \delta \cdot \frac{\log(4K)}{\gamma} \cdot \frac{\sqrt{p}}{K} \log \frac{\sqrt{p}}{K}. \quad (6)$$

We will now try to develop intuition about the theorem. We begin by explaining the sets involved in Equations (3), (4). We consider $B_2^k(r_1)$ to be a set containing all the latent codes of the first layer of the generator that could be potentially pre-images of any sensed signal x. We refer to $B_2^k(r_1)$ as the domain of G and to $G_1(B_2^k(r_1))$ as the range of G_1. The *extended range* contains all vectors that lie within an l_1 ball of radius r_2 from some point in the range of G_1. This is the set $G_1(B_2^k(r_1)) \oplus B_1^p(r_2)$ that both minimizations are performed in.

Let's now consider the error bound of (6). First, $\bar{z}^p$ is the latent code in the extended range that best explains the image x. We refer to this as the true optimum latent code. Next, $\tilde{z}^p$, is the measurements optimum, i.e. the latent code in the extended range that best explains the measurements Ax. It is important to realize that a reconstruction algorithm only has access to this measurement error and can never compute $\bar{z}^p$. Our goal is to show that $\tilde{z}^p$ produces an image close to the one produced by $\bar{z}^p$.

Our theorem states that given enough measurements m, the measurements optimum is nearly as good as the true optimum (see (6) and Remark 3).

Remark 1 (Choice of K)**.** *The size of the extended range affects the required number of measurements ((4)) and our error bound (see (6)). Observe that the size of the extended range is directly controlled by K, since, for any fixed δ, we set $r_2 = \frac{K\delta}{L_2}$. As K increases, we explore a bigger set and both terms on the right side of (6) become smaller. However, measurements scale quadratically with K. We can set K to scale approximately as $\sqrt{k}$ (see Remark 3 for details on how all the quantities can scale). For that choice of K, observe that our result requires measurements that scale linearly on k (and only logarithmic in p) while the CSGM result requires measurements that scale linearly on p. The costs for the small increase in the measurements, are 1) the additive error scales with $\sqrt{p}$, 2) we are restricted to exploring a small radius.*

In practice, these can be tuned as hyperparameters and our experiments show that even small expansions significantly outperform CSGM in numerous inverse problems.

Remark 2 (CSGM sample bound applied directly on the intermediate layer)**.** *We compare to the result we obtain by applying CSGM to the intermediate layer generator. That would yield measurements that scale as:*

$$m = \Omega\left(k \log\left(\frac{L_1 L_2 r_1}{\delta}\right) + p \log\left(\frac{L_2 r_2}{\delta}\right)\right).$$

These many measurements result in an additive error term of $O(\delta)$. Our new bound requires fewer measurements when the free parameter K is smaller than $\sqrt{p}$.

Remark 3 (Parameter Scaling)**.** *There are various ways to set the parameters in our bounds, depending on the scaling of sizes of the intermediate layers and the Lipschitz constants. For typical piecewise linear networks with d layers and maximum n neurons in each layer, we know that the end-to-end Lipschitz constant $L \leq L_1 \cdot L_2$ might scale as n^d for bounded maximum weights. Hence, as in CSGM, we may set r_1 to scale as n^d. The error term $||x - G_2(\bar{z}^p)||$ scales linearly with n. Hence, we need to choose δ, K such that the additive term in inequality (6) scales sublinearly. We may set δ to scale as $\frac{1}{\sqrt{p}}$. To get the same order of measurements as CSGM, we may set K to scale as $\sqrt{k}$. For that choice of parameters, the radius for the intermediate search, i.e. r_2 scales as $\sqrt{\frac{k}{p}} n^{-d_2}$, where d_2 is the depth of G_2.*

3.3. Sketch of the proof

The central novelty of our proof is how we upper bound the metric entropy of the epsilon nets used to cover the extended range of the generator, i.e. the set $G_1(B_2^k(r_1)) \oplus B_1^p(r_2)$. First, we observe that if S_1 is a epsilon net for $G_1(B_2^k(r_1))$ and S_2 is an epsilon net for $B_1^p(r_2)$, then a simple bound for the size of an epsilon net on the extended range will have at most $|S_1| \cdot |S_2|$ elements.

CSGM uses a volumetric argument to upper bound the size of the epsilon net for S_1. Our key idea is that using the same method to bound the size of the cover for the l_1 ball is sub-optimal for small radii. Instead, we use Maurey's empirical method or the related Sudakov's minoration inequality (Pisier, 1986; Wainwright, 2019) yielding logarithmic (instead of linear) dependence on the dimension p. Maurey's bound poses technical challenges that we need to address when extending the chaining argument of the CSGM proof. With Maurey's method, successive nets in the chaining can have significantly higher metric entropy for large radii. To minimize the additive error in our bound during chaining, we switch from volumetric epsilon-nets to Maurey's method at the right selected scale. The full proof of our Theorem can be found in the Appendix.

3.4. S-REC for partial circulant matrices

We extend the theory of matrices that satisfy the S-REC condition beyond i.i.d. Gaussian measurements. To establish that a family of random matrices satisfies this condition (and hence obtains sample complexity bounds), three conditions must be proved with high probability (Bora et al., 2017; Baraniuk et al., 2008): (1) The random matrix A should satisfy the Johnson-Lindenstrauss (JL) lemma on a suitable ϵ-net, (2) The matrix operator norm should be bounded: $\|A\|_{op} \leq \sqrt{n}$, and (3) for a fixed vector x, $\|Ax\| \leq 2\|x\|$.

Here we establish that randomly signed partial circulant matrices satisfy the S-REC condition for a number of measurements scaling similarly to Gaussian i.i.d. measurements.

Lemma 1. *Consider the setting of Theorem 1. Let $g = [g_1, \cdots, g_n]$ be a vector with i.i.d. Gaussian entries of variance $1/m$, let $F \in \mathbb{R}^{m \times n}$ be a partical circulant matrix that has g in its first row, and let $D \in \mathbb{R}^{n \times n}$ be a diagonal matrix with uniform ± 1 entries along its diagonal. Then for $m = \Omega\left(\frac{1}{(1-\gamma)^2}(k \log \frac{L_1 L_2 r_1}{\delta} + K^2 \log p) \log^4(n)\right)$, FD satisfies S-REC$(G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2)), 1-\gamma, \delta \cdot \frac{\log(4K)}{\gamma} \cdot \frac{\sqrt{p}}{K} \log \frac{\sqrt{p}}{K})$ with probability $1 - e^{-\Omega(m)}$.*

Our proof of this lemma can be found in the Appendix and relies on previous results establishing JL properties for partial circulant matrices post-multiplied by random diagonal matrices (Krahmer & Ward, 2011; Hinrichs & Vybíral, 2011).

There is an important computational benefit in such structured measurement matrices. We are sensing high resolution images that are 1024×1024 for 3 color channels resulting in signal dimension n being 3 million. If measurements are at ten percent (a typically challenging compressed sensing regime), that results to $m \times n$ matrices that are $300k \times 3m$ which require gigabytes to store and hit GPU memory limitations. Therefore random Gaussian measurement matrices cannot be implemented for high resolution imaging. Partial circulant matrices require orders of magnitude less memory due to their structure and matrix-vector products can be computed much faster using FFT. We expect that these benefits will have a key role for future high-resolution imaging systems.

4. Experiments

4.1. Algorithmic adaptations to StyleGAN

Up to this point, we have presented and theoretically analyzed the ILO algorithm. Our method is not tied to any specific architecture and it only assumes access to a generative model and the underlying domain of the latent space of the initial layer. In this section, we present empirical innovations on how to use our framework with the state-of-the-art generative model StyleGAN-2 (Karras et al., 2020).

StyleGAN-2 has several peculiarities that need to be taken into account for the design of a compressed sensing algorithm. First, in StyleGAN-2 the initial latent code $z^k \in \mathbb{R}^k$ is not fed directly to the model. Instead, it is first mapped through a multilayer linear network, *the mapping network*, to an intermediate representation $w^k \in \mathbb{R}^k$. We refer to the domains of z^k, w^k as $\mathcal{Z}, \mathcal{W}$ respectively. During training, a z^k is sampled according to a distribution on $\mathcal{Z}$, it gets transformed through the mapping network to a $w^k \in \mathcal{W}$ and one copy of w^k is fed to each one of the 18 layers of StyleGAN-2. Additionally, each one of the layers receives a noise vector u^k (unique for each layer).

4.1.1. Optimization setting

The first thing to decide is which intermediate layer will be used to split the StyleGAN-2 generator. We observe that we obtain better results with multiple splits. We consider the generator of StyleGAN-2 as a composition of layers $G_1 \circ G_2 \circ ... \circ G_{18}$ and we run Algorithm (1) in rounds, where in each round the initial layer is discarded.

To ensure that we stay in an l_1 ball around the manifold at each layer, we use Projected Gradient Descent (PGD) (Nesterov, 2003). To implement the projection to an l_1 ball around the current best solution (see line 4 of Algorithm (1)), we use the method of Duchi et al. (2008). Guided by our theory, we increase the maximum allowed deviation as we move to higher dimensional latent spaces. The radii of

the balls are tuned separately as hyperparameters, for a full description see the Appendix.

For all inverse problems, it is helpful to allow the w^k vectors to deviate (Menon et al., 2020), i.e. we can optimize over a sequence $\{w_i^k\}_{i=1}^{18}$. The deviations are typically regularized with an additional term in the loss function, which captures the geodesic distance of the vectors. PULSE reports that optimizing only over the first five noise vectors, i.e. $\{u_i^k\}_{i=1}^{5}$, yields better reconstructions for super-resolution comparing to optimizing over the whole sequence. We show that this is not necessarily true if this optimization is performed sequentially. Our method starts by optimizing only the first five noise vectors (as in PULSE), but we gradually allow optimization of the rest of the latent vectors as we move to higher dimensional latent spaces.

4.2. Loss functions and adaptation to general inverse problems

Here we consider the effect of different loss functions in solving general inverse problems. It has been observed that LPIPS yields optimal performance with image size 256×256 (Karras et al., 2020). Therefore, we downsample images from 1024×1024 to 256×256 pixels. If the given image is inpainted, missing pixels are mixed with observed pixels during this downsampling. We observe that this blending leads to distorted reconstructions when using the LPIPS loss. Hence, for inpainting under scarce measurements we use only the MSE loss. We note that unlike the previously proposed methods, ILO can work for inpainting with extremely few observed pixels – even with less than 1% of the whole image. If we observe a significant portion of the image, then we use both LPIPS and MSE. To address these distortion issues, we minimize the perceptual distance between the generated image and a superimposed reconstruction, i.e. we replace the missing pixels of the observed image with the ones generated by StyleGAN prior to downsampling.

For super-resolution, we use a weighted average of LPIPS and MSE (as in inpainting with sufficient measurements). To compare the high-resolution and low-resolution images, we first downsample with cubic interpolation (Keys, 1981) as in PULSE. We also consider the problem of denoising, where Gaussian noise is added to the image. As usual, we assume knowledge to the forward operator $\mathcal{A}(x)$. Simply inverting a noisy high-resolution image creates grainy reconstructions due to the expressive power of StyleGAN-2. We address this in the optimization process by adding gaussian noise to the generated images before using them in the loss function. We call this new technique **Stochastic Noise Addition** (SNA).

4.3. Results

We show that ILO obtains state-of-the-art unsupervised performance for solving inverse problems with deep generative models in four different settings: inpainting, super-resolution, denoising and compressed sensing with circulant matrices. We compare with different variants of the CSGM algorithm using optimization and loss function innovations introduced in PULSE and StyleGAN. Unless stated otherwise, we will denote with CSGM + MSE the optimization procedure described in PULSE for the StyleGAN generator. Through a wide variety of experiments, we observe that ILO largely outperforms alternative techniques, both in terms of visual quality and in terms of true MSE error. We measure the latter on images sampled randomly from Celeba-HQ (Liu et al., 2018; Lee et al., 2020). Finally, to show the benefits of extending the range of the generator, we illustrate how one can use an adversarially robust classifier to guide the generation of human faces that look like objects from ImageNet (Deng et al., 2009).

Inpainting: For inpainting, the algorithm tries to complete missing pixels to a given image. The measurement process corresponds to a linear matrix that has rows that are a subset of the identity. Results for inpainting are shown in Figure 1. We perform two types of experiments. First, we mask important facial features from real images (collected from the web) and generated images from StyleGAN-2. Next, we do randomized inpainting, i.e. we inpaint pixels of a given image independently with a pre-defined probability. We experiment with observation probabilities up to 1%. This is a very challenging scenario: a human observer cannot distinguish face characteristics from such few pixels, e.g. see Figure 1 last row, second column. As shown in the Figure, ILO gives reconstructions that look much closer to the hidden image than the other methods. Our method is able to give surprisingly accurate reconstructions even under extreme scarce measurements (see last column, last row of Figure 1). To quantify the performance of the different methods we randomly select a few images from Celeba-HQ (Liu et al., 2018; Lee et al., 2020) and reconstruct at different levels of sparsity. Figure 2 column 1 shows that ILO is $2\times$ better in terms of reconstruction error anywhere between $5\% - 100\%$ observed pixels.

Denoising: Our next experiment is on denoising. To ablate the SNA framework we introduced, we show results with and without our technique on an image with additive noise of standard deviation $\sigma = 30$. Results are summarized in Table 1. Since it is clear that SNA consistently improves reconstruction, we use it in all subsequent denoising experiments.

We compare with the CSGM framework using MSE, only LPIPS or a combination of both loss functions. For ILO, we only use a weighted combination of MSE and LPIPS. We

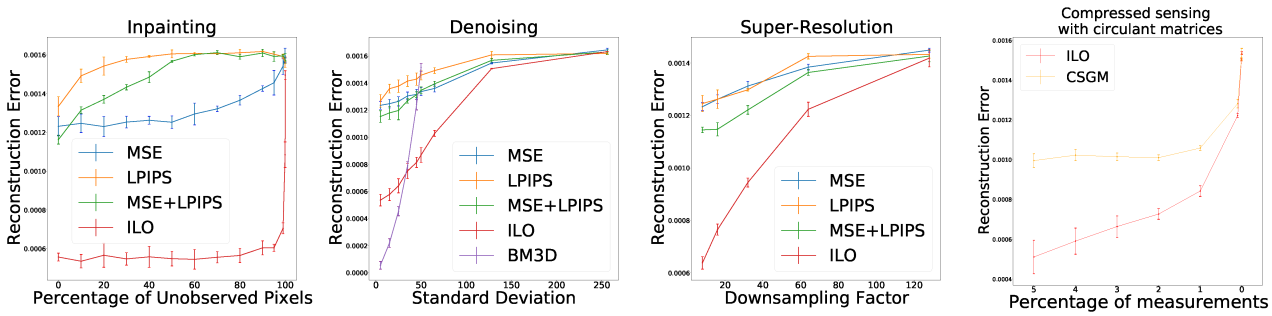

Figure 2. Plots showing the true MSE error on Celeba-HQ images, i.e. the MSE between the real image (that we never observe) and the reconstructed image from the measurements. From left to right: Inpainting, Denoising, Super-resolution and Compressed sensing with partial circulant matrices. As shown, ILO significantly outperforms all previous methods except in the very noisy regime.

Algorithm	SNA	PSRN (dB)
CSGM	✗	19.89
	✓	21.38
ILO	✗	28.34
	✓	**32.92**

Table 1. Results with and without SNA for a noisy image ($\sigma = 30$).

Figure 3. Results on the super-resolution task. ILO (ours) gives more accurate reconstructions comparing to PULSE (third column) and other baselines. Many biased reconstructions can be corrected by applying ILO on the weighted combination of MSE and LPIPS.

also compare with a standard denoising method, the BM3D algorithm (Dabov et al., 2006). We vary the noise standard deviation from 5 to 256 and clip the perturbed values to the range $[0, 255]$ (RGB). Results are shown in Figure 2, second column. We observe that ILO outperforms all the previously proposed CSGM based methods by a large margin. For the typical setting of $\sigma = 25$, ILO is $1.8\times$ better than the best performing CSGM baseline. BM3D shows excellent performance, outperforming all other methods in the very low noise regime but rapidly deteriorates for harder settings. We refer the reader to the for visual results and additional denoising experiments.

Super-resolution: We report results on super-resolution, the only task PULSE was actually designed for. We sample images from Celeba-HQ, downsample using Bicubic Downsampling (as done in PULSE) and measure the reconstruction error. Three example reconstructions are shown in Figure 3. We also report reconstruction error on Celeba-HQ. Results are reported in Figure 2, third column. As shown, ILO outperforms significantly all the other methods, including PULSE (CSGM + MSE). To give some examples, when the image is downscaled from 1024×1024 to 64×64 (scaling factor 16), ILO is $1.65\times$ better than PULSE in terms of reconstruction error. For 32×32 images, ILO is $1.4\times$ better than PULSE.

As shown, our method not only generates reconstructions that look much closer to the true image, but also appears to generate more racially diverse samples (Jain et al., 2020; Tan et al., 2020; Menon et al., 2020), e.g. see third row.

Compressed sensing with partial circulant matrices: For an experiment with observations of random projections we used partial circulant measurement matrices with random signs. Lemma 1 establishes that such matrices satisfy the conditions for Theorem 1. Figure 2, column 4, shows the reconstruction error when varying the number of measurement rows. When the number of measurements is 5% of the dimension n, ILO performs $2\times$ better than CSGM in terms of reconstruction error.

Out of Distribution generation: Our method can generate images that lie outside of the range of the pre-trained generator. By choosing the radius of the l_1 ball for each layer, we control the trade-off between how natural (comparing to the dataset the model was trained on) these images look, and the out-of-distribution generation capability of our model.

To demonstrate this, we run the following experiment; we remove entirely the loss functions that relate the generated

Figure 4. Illustration of using a classifier as a differentiable forward operator. Here we assume that the only observation is $y = \mathcal{A}(x|\text{class})$ where $\mathcal{A}$ is an ImageNet classifier. The classes used in this Figure are (from top-left): Frog, Coral, Irish Wolf Dog, Goldfish, Boston Terrier Dog and Apple. We use a robust classifier as proposed by Santurkar et al. (2019) and solve the inverse problem to generate images that look like these classes. The difference with Santurkar et al. (2019) is that we perform the search using ILO in the StyleGAN-2 generator latent spaces as opposed to pixels and that keeps images closer to human faces.

images with a reference image (i.e. MSE and LPIPS) and we add a new classification loss term using an external classifier trained on a different domain. Essentially, we search for latent codes that lie in an l_1 ball around the range of intermediate layers and maximize the probability that the generated image belongs to a certain category. We consider a classifier trained on ImageNet (Deng et al., 2009). This optimization problem is one of the simplest methods to create adversarial examples (Xiao et al., 2018) and hence the generated images will not be visually interesting. However, if our classifier is adversarially robust, then even optimizing directly over the pixel space leads to an interesting generative process (Santurkar et al., 2019). We use the latent space of StyleGAN-2 to generate images of faces with fruit or animal characteristics. The radius of the l_1 projection at different layers controls the distance of the generated images to human faces. The results are shown in Figure 4.

Running time: Our algorithm runs CSGM as the first step and therefore initially seems to be strictly slower. Surprisingly, ILO can find better solutions than CSGM in *fewer* total steps. StyleGAN-2 typically requires $300 - 1000$ optimization steps (on the first layer) for a good reconstruction (Karras et al., 2019; 2020). However, we observe that running 50 steps in each one of the first four layers outperforms CSGM. That said, ILO continues to improve with more iterations, also depending on task, number of measurements and hyperparameters. In practice, the obtained inverse problems required approximately $30 - 60$ seconds per image on a single 1080Ti GPU. For a discussion on hyperparameters, their effects on running times and comparisons to other baselines see the Appendix.

Related Work: There has been significant recent work in unsupervised methods for inverse problems using pre-trained generative models. Recently, Liu & Scarlett (2020) showed that the sample scaling of CSGM is near-optimal in the absence of further assumptions. Hand & Voroninski (2018) proved algorithmic convergence guarantees for solving non-convex linear inverse problems with deep generative priors under random weight assumptions. Faster recovery algorithms were proposed by Raj et al. (2019); Shah & Hegde (2018) while Pandit et al. (2019) analyzed approximate message passing (AMP) for inverse problems in the high-dimensional random limit. Beyond AMP, Regularization-by-Denoising (RED) methods have shown excellent recent performance in imaging, see e.g. Sun et al. (2019). Deep generative models have been developed for MRI (Mardani et al., 2018) and benefited from task-awareness (Kabkab et al., 2018), meta-learning (Wu et al., 2019) and specifically designed autoencoders (Mousavi et al., 2019).

The theoretical framework we introduce is related to the ideas proposed by Dhar et al. (2018) on allowing additive sparse deviations in the generated images. In that case, the recovered signals have the form $G(z) + v$, where $G : \mathbb{R}^k \to \mathbb{R}^n$ is a deep generative model, $z \in \mathbb{R}^k$ is a latent variable and $v \in \mathbb{R}^n$ is an l-sparse vector. The additive term allows the recovery of signals that lie outside of the range. Our approach is very close to this framework but generalizes it since it allows sparse deviations anywhere in the latent space.

Another related recent work is that on GAN surgery (Park et al., 2020). In that paper, the range of the generator is expanded by optimizing intermediate layers directly. The main difference is in the optimization procedure; it is not performed sequentially nor regularized by a previous search in the lower dimensional space as we propose in this paper. Our paper also benefits from the StyleGAN-2 architecture (Karras et al., 2019; 2020) and builds on several key ideas from PULSE (Menon et al., 2020).

Finally, there is significant prior work on deep learning methods that do not rely on pre-trained generators, see e.g. (Lucas et al., 2018; Yu et al., 2019; Liu et al., 2019; Sun & Chen, 2020; Sun et al., 2020; Yang et al., 2019; Tian et al., 2020; Tripathi et al., 2018). Such methods can show excellent performance but require training a network specifically for each reconstruction task. This is in contrast with our framework that can solve all inverse problems universally, leveraging the same pre-trained network.

5. Conclusions and Future Work

We proposed a novel framework for solving inverse problems leveraging pre-trained generative models. Our method expands the range of the generator by optimizing different intermediate layers and achieves excellent performance for several tasks. On the theory side, a central open problem would be to establish global convergence of ILO, possibly following the ideas of (Hand & Voroninski, 2018; Hand et al., 2018) or surfing (Song et al., 2019).

On the empirical side, a central open problem would be the application of our framework in other domains like medical imaging, but that would require pre-trained generative models e.g. for high-resolution MRI images. Another open direction that is particularly exciting is the use of classifiers to generate out-of-distribution samples. Our generated samples show the powerful modularity of combining pre-trained generators with differentiable forward operators that can guide image reconstruction in a data-driven way.

6. Intended Use

ILO is intended as a proof of concept for solving inverse problems by leveraging pre-trained Generative models. The intended use of our implementation using StyleGAN2 and also the classifier is purely as an art project. Our primary goal is to demonstrate that a classifier can produce images outside the range of a pre-trained generator (i.e. human faces) by leveraging intermediate layer optimization. This model is not suitable, for face recognition or any real subject identification or any real subject image manipulation. We are not releasing the classifier transformation code in public because of the potential for abuse. Interested artists can contact us for code.

The training dataset of the used generator (StyleGAN) has been noted to have imbalance of white faces compared to faces of people of color. Furthermore, different reconstruction optimization methods may be biasing reconstructions, an issue we are investigating in on-going work. Our method can be used with any generative model and perhaps a model trained e.g. with FairFace would be better but this is part of our on-going research.

7. Appendix

Symbols

$N(\delta, \Theta, ||\cdot||_q)$ δ-cover of Θ w.r.t. $||\cdot||_q$ norm

$M(\delta, \Theta, ||\cdot||_q)$ δ-packing of Θ w.r.t. $||\cdot||_q$ norm

$B_q^k(r)$ k-dimensional ball of radius r w.r.t. $||\cdot||_q$ norm

$S_1 \oplus S_2$ Minkowski sum of the sets S_1, S_2, i.e. the set $\{x+y | x \in S_1, y \in S_2\}$

$\mathcal{G}(T)$ Gaussian complexity of set T

$[M]$ Set $\{1, ..., M\}$

7.1. Proofs

7.1.1. Metric entropy for the l_1 ball

Definition 2 (Covering number (Wainwright, 2019)). *A δ-covering of a set $\mathbb{T}$ with respect to a metric ρ is a set $\{\theta^1, ..., \theta^M\} \subset \mathbb{T}$ such that for each $\theta \in \mathbb{T}$, there exists some $i \in [N]$ such that $\rho(\theta, \theta^i) \leq \delta$. The δ-covering number $N(\delta, \mathbb{T}, \rho)$ is the cardinality of the smallest δ-cover.*

Definition 3 (Packing number (Wainwright, 2019)). *A δ-packing of a set $\mathbb{T}$ with respect to a metric ρ is a set $\{\theta^1, ..., \theta^M\} \subset \mathbb{T}$ such that $\rho(\theta^i, \theta^j) > \delta$ for all distinct $i, j \in [M]$. The δ-packing number $M(\delta, \mathbb{T}, \rho)$ is the cardinality of the largest δ-packing.*

Lemma 2 (Wainwright (2019)). *For all $\delta > 0$, the packing and covering numbers are related as follows:*

$$M(2\delta, \mathbb{T}, \rho) \leq N(\delta, T, \rho) \leq M(\delta, T, \rho). \tag{7}$$

Theorem 2 (Maurey's Empirical Method (Pisier, 1986)). *Let $B_1^d(r) = \{x \in \mathbb{R}^d \mid ||x||_1 \leq r\}$. Then,*

$$\log N(\delta, B_1^d(r), ||\cdot||_2) \leq \frac{r^2}{\delta^2} \log(2d+1). \tag{8}$$

A short proof of this result follows.

Proof. Fix $x \in \mathbb{R}^d$. Let Z be the following RV:

$$Z = \begin{cases} \text{sgn}(x_i) r e_i, & \text{w.p. } \frac{|x_i|}{r} \\ 0, & \text{w.p.} 1 - \frac{||x||_1}{r} \end{cases} \tag{9}$$

Observe that: $E[Z_i] = \text{sgn}(x_i) r \cdot \frac{|x_i|}{r} = x_i$ and $V[Z_i] = r^2 \cdot \frac{|x_i|}{r} = r|x_i|$.

Let

$$\bar{Z} = \frac{1}{t} \sum_{i=1}^{t} Z_i \tag{10}$$

where Z_i are independent copies of Z.

We have that:

$$E[||\bar{Z} - x||^2] = E\left[\sum_{j=1}^{d} (\bar{Z}_j - x_j)^2\right] \tag{11}$$

$$= \sum_{j=1}^{d} E\left[(\bar{Z}_j - x_j)^2\right] = \sum_{j=1}^{d} V(\bar{Z}_j) \tag{12}$$

$$= \sum_{j=1}^{d} V\left(\frac{1}{t} \sum_{i=1}^{t} (Z_i)_j\right) = \tag{13}$$

$$\frac{1}{t^2} t \sum_{j=1}^{d} V(Z_j) = \frac{1}{t} \sum_{j=1}^{d} r|x_i| = \frac{r||x||_1}{t} \leq \frac{r^2}{t}. \tag{14}$$

If we choose t such that: $\frac{r^2}{t} \leq \delta^2$, then, we have that $E[||\bar{Z} - x||^2] \leq \delta^2$. Hence, for $t \geq \frac{r^2}{\delta^2}$, by the Pigeonhole Principle, we have that there is a $\bar{Z}$ such that: $||\bar{Z} - x|| \leq \delta$. In other words, the set of all possible $\bar{Z}$ form an $\delta-$net for $B_1^d(r)$ for $t \geq \frac{r^2}{\delta^2}$. Set $t = \frac{r^2}{\delta^2}$. We will now count how many $\bar{Z}$ there are. For each $\bar{Z}$, we have t choices, each one of which can take one value among $2d+1$ values. Hence, there are $(2d+1)^t$ different $\bar{Z}$. Therefore, we can create an δ-net of $B_1^d(r)$ that has $(2d+1)^{\frac{r^2}{\delta^2}}$ elements, i.e.

$$\log N(\delta, B_1^d(r), ||\cdot||_2) \leq \frac{r^2}{\delta^2} \log(2d+1).$$

□

The same result (up to constants) for the size of the ϵ-net for an l_1 ball follows from Sudakov's minoration inequality.

Theorem 3 (Sudakov minoration (Sudakov, 1969; Wainwright, 2019)). *Let $\{X_\theta, \theta \in T\}$ be a zero-mean Gaussian process on $T \subset R^d$. Then,*

$$\mathcal{G}(T) \geq \frac{\delta}{2} \sqrt{\log M(\delta/2, T, \rho_X)}, \tag{15}$$

with $\rho_X(\theta_1, \theta_2) = \sqrt{E[(X_{\theta_1} - X_{\theta_2})^2]}$.

Corollary 1.

$$\log N(\delta, B_1^d(r), ||\cdot||_2) \leq \frac{16 r^2}{\delta^2} \log d. \tag{16}$$

Proof. Observe that:

$$\mathcal{G}(B_1^d(r)) = E_w\left[\sup_{||u||_1 \leq r} u^T w\right] \tag{17}$$

$$\leq r E_w\left[||w||_\infty\right] \tag{18}$$

$$\leq 2r\sqrt{\log d}. \tag{19}$$

By Inequality (15),

$$\mathcal{G}(T) \geq \frac{\delta}{2}\sqrt{\log M(\delta/2, T, \rho_X)} \Rightarrow \tag{20}$$

$$\log M(\delta/2, B_1^d(r), ||\cdot||_2) \leq 16\frac{r_1^2}{\delta^2}\log d. \tag{21}$$

It follows from the definition of the covering number that:

$$M(\delta, T, ||\cdot||_2) \leq M(\delta/2, T, ||\cdot||_2).$$

By Inequality (7), we also have:

$$N(\delta, T, ||\cdot||_2) \leq M(\delta, T, ||\cdot||_2). \tag{22}$$

Hence,

$$\log N(\delta, B_1^d(r), ||\cdot||_2) \leq \frac{16r^2}{\delta^2}\log d. \tag{23}$$

□

Theorem 4 (Volume rations and metric entropy (Wainwright, 2019))**.** *Let Θ be an arbitrary set. Then,*

$$\frac{\text{vol}(\Theta)}{\text{vol}(\delta B_q^d(1))} \leq N(\delta, \Theta, ||\cdot||_q) \leq \frac{\text{vol}\left(\frac{2}{\delta}\Theta \oplus B_q^d(1)\right)}{\text{vol}(B_q^d(1))} \tag{24}$$

Corollary 2.

$$\log N(\delta, B_1^d(r), ||\cdot||_2) \leq d\log\frac{4r}{\delta} \tag{25}$$

Proof. By Theorem 4, we have that:

$$N(\delta, B_1^d(r), ||\cdot||_2) \leq \frac{\text{vol}\left(\frac{2}{\delta}B_1^d(r) \oplus B_2^d(1)\right)}{\text{vol}(B_2^d(1))} \tag{26}$$

$$\frac{\text{vol}\left(\frac{2}{\delta}B_2^d(r) \oplus B_2^d(1)\right)}{\text{vol}(B_2^d(1))} \leq \left(\frac{2r}{\delta}+1\right)^d \tag{27}$$

$$\leq \left(\frac{4r}{\delta}\right)^d. \tag{28}$$

□

Remark 4. *Observe that by Theorem 2 and Corollary 2, we get two different upper bounds regarding the covering of the l_1-ball. With Maurey's method, the covering number depends logarithmically in the dimension but polynomially on $\frac{1}{\epsilon}$. On the other hand, the volumetric argument gives polynomial dependence on the dimension and logarithmic dependence on $\frac{1}{\epsilon}$. The Maurey's bound is tighter when $\epsilon = \Omega\left(\frac{r}{\sqrt{d}}\right)$.*

7.1.2. S-REC

Lemma 3 (S-REC for nested l_1-ball)**.** *Let $G = G_2 \circ G_1$ with $G_1 : \mathbb{R}^k \to \mathbb{R}^p$ be an L_1-Lipschitz function and $G_2 : \mathbb{R}^p \to \mathbb{R}^n$ be an L_2-Lipschitz function. Let $A \in R^{m\times n}$ be a random matrix with $A_{ij} \sim \mathcal{N}(0, 1/m)$ i.i.d. entries.*

Then, if

$$m = \frac{1}{(1-\gamma)^2}\Omega\left(k\log\frac{L_1L_2r_1}{\delta} + K^2\log p\right) \tag{29}$$

$$r_2 = \frac{K\cdot\delta}{L_2}, \quad 1 < K < \sqrt{p} \tag{30}$$

w.p. $1 - e^{-\Omega((1-\gamma)^2 m)}$, we have that A satisfies S-REC$(G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2)), \gamma, \log(4K)\cdot\frac{\sqrt{p}}{K}\cdot\log\frac{\sqrt{p}}{K})$.

Proof. Using Theorem 4, we get that:

$$N\left(\frac{\delta}{L_1\cdot L_2}, B_2^k(r_1), ||\cdot||_2\right) \leq \left(\frac{2L_1L_2r_1}{\delta}+1\right)^k \leq \left(\frac{4L_1L_2r_1}{\delta}\right)^k \tag{31}$$

Using the fact that $G_2 \circ G_1$ is L_1L_2 Lipschitz, we get that:

$$N(\delta, G(B_2^k(r_1)), ||\cdot||_2) \leq \left(\frac{4L_1L_2r_1}{\delta}\right)^k \tag{32}$$

Using Maurey's Empirical Method (see Theorem 2), we get that:

$$\log N\left(\frac{\delta}{L_2}, B_1^p(r_2), ||\cdot||_2\right) \leq \frac{r_2^2L_2^2}{\delta^2}\log(2p+1). \tag{33}$$

Setting $r_2 = \frac{K\cdot\delta}{L_2}$ and using the fact that G_2 is L_2-Lipschitz, we get:

$$\log N\left(\delta, G_2(B_1^p(r_2)), ||\cdot||_2\right) \leq K^2\log 3p \tag{34}$$

By (32), (34), we get that:

$$\log N\left(\delta, G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2)), ||\cdot||_2\right) \leq k\log\frac{4L_1L_2r_1}{\delta} + K^2\log 3p. \tag{35}$$

By JL lemma, if $m = \frac{1}{a^2}\Omega\left(k\log\frac{4L_1L_2r_1}{\delta} + K^2\log 3p\right)$, then w.p. $1 - e^{-\Omega(a^2m)}$, we have that:

$$||AG_2(\hat{z}_2^p) - AG_2(\hat{z}_1^p)|| \geq (1-a)||G_2(\hat{z}_2^p) - G_2(\hat{z}_1^p)||, \ \forall \hat{z}_1^p, \hat{z}_2^p \in S \tag{36}$$

where S is a minimal δ-net of $G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2))$.

Let $z_1^p, z_2^p \in G_1(B_2^k(r_1))) \oplus B_1^p(r_2)$ and $\hat{z}_1^p = \text{argmin}_{\tilde{z}_1^p \in S}||z_1^p - \tilde{z}_1^p||$, $\hat{z}_2^p = \text{argmin}_{\tilde{z}_2^p \in S}||z_2^p - \tilde{z}_2^p||$. Then,

$$||AG_2(z_2^p) - AG_2(z_1^p)|| \geq ||AG_2(\hat{z}_2^p) - AG_2(\hat{z}_1^p)|| \\ -||AG_2(z_2^p) - AG_2(\hat{z}_2^p)|| - ||AG_2(z_1^p) - AG_2(\hat{z}_1^p)|| \quad (37)$$

$$\geq (1-a)||G_2(\hat{z}_2^p) - G_2(\hat{z}_1^p)|| - ||AG_2(z_2^p) - AG_2(\hat{z}_2^p)|| \\ -||AG_2(z_1^p) - AG_2(\hat{z}_1^p)|| \quad (38)$$

$$\geq (1-a)||G_2(z_2^p) - G_2(z_1^p)|| - \\ (1-a)\left(||G_2(z_2^p) - G_2(\hat{z}_2^p)|| + ||G_2(z_1^p) - G_2(\hat{z}_1^p)||\right) \\ -||AG_2(z_2^p) - AG_2(\hat{z}_2^p)|| - ||AG_2(z_1^p) - AG_2(\hat{z}_1^p)|| \quad (39)$$

$$\geq (1-a)||G_2(z_2^p) - G_2(z_1^p)|| - 2\delta \\ -||AG_2(z_2^p) - AG_2(\hat{z}_2^p)|| - ||AG_2(z_1^p) - AG_2(\hat{z}_1^p)|| \quad (40)$$

By Lemma 4, we have that w.p. $1 - e^{-\Omega(m)}$, $||AG_2(z_2^p) - AG_2(\hat{z}_2^p)|| + ||AG_2(z_1^p) - AG_2(\hat{z}_1^p)|| = O\left(\log(4K) \cdot \frac{\sqrt{p}}{K} \cdot \log\frac{\sqrt{p}}{K}\right) \cdot \delta$. Let $a = 1 - \gamma$. Hence,

$$||AG_2(z_2^p) - AG_2(z_1^p)|| \geq \gamma||G_2(z_2^p) - G_2(z_1^p)|| \\ - \log(4K) \cdot \frac{\sqrt{p}}{K} \cdot \log\frac{\sqrt{p}}{K} \cdot \delta. \quad (41)$$

□

Lemma 4. *Let $G = G_2 \circ G_1$ with $G_1 : \mathbb{R}^k \to \mathbb{R}^p$ be an L_1-Lipschitz function and $G_2 : \mathbb{R}^p \to \mathbb{R}^n$ be an L_2-Lipschitz function. Let $A \in R^{m \times n}$ be a random matrix with $A_{ij} \sim \mathcal{N}(0, 1/m)$ i.i.d. entries. Let M_0 be a $\frac{\delta}{L_2}$ net of $G_1(B_2^k(r_1)) \oplus B_1^p(r_2)$ such that $\log|M_0| \leq k\log\left(\frac{4L_1L_2r_1}{\delta}\right) + K^2\log 3p$.*

Then, if

$$m = \Omega\left(k\log\left(\frac{4L_1L_2r_1}{\delta}\right) + K^2\log p\right), \\ r_2 = \frac{K\cdot\delta}{L_2}, \quad 1 < K < \sqrt{p}. \quad (42)$$

then for any $x \in G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2))$, if $x' = \text{argmin}_{\hat{x} \in G_2(M_0)}||x - \hat{x}||$, w.p. $1 - e^{\Omega(m)}$, we have that:

$$||A(x - x')|| = O\left(\log(4K) \cdot \frac{\sqrt{p}}{K} \cdot \log\frac{\sqrt{p}}{K}\right) \cdot \delta. \quad (43)$$

Proof. From Lemma 8.2 of (Bora et al., 2017), we have that if $\epsilon \geq 2 + \frac{4}{m}\log\frac{2}{f}$, then

$$P(||Ax|| \geq (1+\epsilon)||x||) \leq f. \quad (44)$$

Let $N_0 \subseteq N_1 \subseteq ... \subseteq N_l$ be a chain of minimal δ_i-nets of $G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2))$.

Let also:

$$T_i = \{x_{i+1} - x_i | x_{i+1} \in N_{i+1}, x_i \in N_i\}. \quad (45)$$

By union bound,

$$P(||At|| \leq (1+\epsilon_i)||t||, \quad \forall i \in [0, ..., l-1], \quad \forall t \in T_i) \geq \\ 1 - \sum_{i=0}^{l-1} |T_i| f_i, \quad (46)$$

where $\epsilon_i = 2 + \frac{4}{m}\log\frac{2}{f_i}$. We want to choose f_i such that $\sum_{i=0}^{l-1}|T_i|f_i$ decays exponentially with m.

First notice that:

$$\log|T_i| \leq \log|N_{i+1}| + \log|N_i| \quad (47)$$

To develop bounds for $\log|N_i|, \log|N_{i+1}|$ we first need to decide how δ_i decays and then whether we are going to use Maurey's method or the volumetric argument.

We choose $\delta_i = \frac{\delta}{2^i}$. Now assume $m = K^2\log(3p) + k\log\left(\frac{L_1L_2r_1}{\delta}\right)$.

For $0 \leq i < \log\frac{\sqrt{p}}{K}$ we will use Maurey's method.

$$\log|T_i| \leq 2\log|N_{i+1}| \quad (48)$$

$$\leq 2\left(\left(\frac{K\delta}{\delta_{i+1}}\right)^2\log(3p) + k\log\left(\frac{L_1L_2r_1}{\delta_i}\right)\right) \quad (49)$$

$$\leq 2\cdot\left(4^{i+1}K^2\log(3p) + k\log\left(\frac{L_1L_2r_1}{\delta}\right) + k(i+1)\right) \quad (50)$$

$$\leq 2\cdot\left(4^{i+1}K^2\log(3p) + 2k\log\left(\frac{L_1L_2r_1}{\delta}\right)(i+1)\right) \quad (51)$$

$$\leq 2\cdot 4^{i+1}m. \quad (52)$$

To get probability that decays exponentially with m, we choose:

$$\log f_i = -3\cdot 4^{i+1}m \quad (53)$$

$$\epsilon_i = O(1) + 3\cdot 4^{i+1}. \quad (54)$$

For $\log\frac{\sqrt{p}}{K} \leq i \leq l-1$, we will use the volumetric argument.

$$\log|T_i| \leq p\log\left(4K\frac{\delta}{\delta_i}\right) + p\log\left(4K\frac{\delta}{\delta_{i+1}}\right) + \\ k\log\left(\frac{L_1L_2r_1}{\delta_i}\right) + k\log\left(\frac{L_1L_2r_1}{\delta_{i+1}}\right) \quad (55)$$

$$\leq 2p\log(4K) + p(2i+1) + \\ 2k\log\left(\frac{L_1L_2r_1}{\delta}\right) + k(2i+1) \quad (56)$$

$$\leq 2p\log(4K) + 3pi + 2k\log\left(\frac{L_1L_2r_1}{\delta}\right) + 3ki \quad (57)$$

$$\leq 5ip\log(4K) + 5ik\log\left(\frac{L_1L_2r_1}{\delta}\right) \quad (58)$$

We choose:

$$\log f_i = -6ip\log(4K) - 6ik\log\left(\frac{L_1L_2r_1}{\delta}\right) \quad (59)$$

$$\epsilon_i \leq O(1) + \log(4K)\frac{ip}{m} + i. \quad (60)$$

Notice that:

$$\log|T_i|f_i \leq -ip\log(4K) - ik\log\left(\frac{L_1L_2r_1}{\delta}\right) \leq -im. \quad (61)$$

For that choice of parameters, observe that:

$$P(||At|| \leq (1+\epsilon_i)||t||, \quad \forall i \in [0,...,l-1], \quad \forall t \in T_i) \quad (62)$$

$$= 1 - e^{-\Omega(m)}. \quad (63)$$

Let x be the image we want to recostruct and x_i be the closest point of that image to the δ_i net. Then,

$$x - x_0 = \sum_{i=0}^{l-1}(x_{i+1} - x_i) + x - x_l \Rightarrow \quad (64)$$

$$||Ax - Ax_0|| \leq \sum_{i=0}^{l-1}||Ax_{i+1} - Ax_i|| + ||Ax - Ax_l||. \quad (65)$$

Now w.h.p. $||Ax_{i+1} - Ax_i|| \leq (1+\epsilon_i)||x_{i+1} - x_i||$. Therefore, w.h.p.:

$$||Ax - Ax_0|| \leq \sum_{i=0}^{l-1}(1+\epsilon_i)||x_{i+1} - x_i|| + ||Ax - Ax_l|| \quad (66)$$

$$\leq \sum_{i=0}^{l-1}(1+\epsilon_i)\delta_i + ||Ax - Ax_l|| \quad (67)$$

$$\leq \sum_{i=0}^{\log\frac{\sqrt{p}}{K}-1}\left(O(1) + 3\cdot 4^{i+1}\right)\frac{\delta}{2^i} +$$

$$+ \sum_{i=\log\frac{\sqrt{p}}{K}}^{l-1}\left(O(1) + \log(4K)\frac{ip}{K^2\log 3p} + i\right)\frac{\delta}{2^i} +$$

$$+||Ax - Ax_l|| \quad (68)$$

$$\leq O\left(\log(4K)\cdot\frac{\sqrt{p}}{K}\cdot\log\frac{\sqrt{p}}{K}\right)\cdot\delta + ||Ax - x_l||. \quad (69)$$

Observe that:

$$||Ax - Ax_l|| \leq ||A||\cdot||x - x_l|| \quad (70)$$

$$\leq 2\sqrt{n}||x - x_l|| \quad (71)$$

$$\leq 2\frac{\sqrt{n}}{2^l}\delta. \quad (72)$$

For $l = \log n$, we have that $||Ax - Ax_l|| \leq \delta$. Hence,

$$||Ax - Ax_0|| \leq O\left(\log(4K)\cdot\frac{\sqrt{p}}{K}\cdot\log\frac{\sqrt{p}}{K}\right)\cdot\delta. \quad (73)$$

□

7.1.3. Proof of main theorem

Proof of Theorem 1. Let $\delta_{l_1} = \left(\log(4K)\cdot\frac{\sqrt{p}}{K}\log\frac{\sqrt{p}}{K}\right)\delta$. Then,

$$||G_2(\tilde{z}^p) - G_2(\tilde{z}^p)|| \leq \quad (74)$$

$$\frac{||AG_2(\tilde{z}^p) - AG_2(\bar{z}^p)|| + \delta_{l_1}}{\gamma} \quad (75)$$

$$\leq \frac{||Ax - AG_2(\bar{z}^p)|| + ||Ax - AG_2(\hat{z}^p)|| + \delta_{l_1}}{\gamma} \quad (76)$$

$$\leq \frac{2||Ax - AG_2(\bar{z}^p)|| + \delta_{l_1}}{\gamma} \quad (77)$$

$$\leq \frac{4||G_2(\bar{z}^p) - x|| + \delta_{l_1}}{\gamma}. \quad (78)$$

Finally, observe that:

$$||G_2(\tilde{z}^p) - x|| \leq ||G_2(\bar{z}^p) - x|| + ||G_2(\bar{z}^p) - G_2(\tilde{z}^p)|| \quad (79)$$

$$\leq \left(1 + \frac{4}{\gamma}\right)||x - G_2(\bar{z}^p)|| + \frac{\delta_{l_1}}{\gamma}. \quad (80)$$

□

Remark 5. *Similar to the analysis of the CSGM paper (see Lemma 4.3), γ is a constant that we control and we may set it to $\gamma = \frac{4}{5}$ to get the same scaling term with CSGM.*

7.1.4. Proof of Lemma 1

Lemma 5. *Consider the setting of Theorem 1. Let $g = [g_1, \cdots, g_n]$ be a vector with i.i.d. Gaussian entries of variance $1/m$, let $F \in \mathbb{R}^{m\times n}$ be a partical circulant matrix that has g in its first row, and let $D \in \mathbb{R}^{n\times n}$ be a diagonal matrix with uniform ± 1 entries along its diagonal. Then for $m = \Omega\left(\frac{1}{(1-\gamma)^2}(k\log\frac{L_1L_2r_1}{\delta} + K^2\log p)\log^4(n)\right)$, FD satisfies S-REC$(G_2(G_1(B_2^k(r_1)) \oplus B_1^p(r_2)), 1-\gamma, \delta\cdot\frac{\log(4K)}{\gamma}\cdot\frac{\sqrt{p}}{K}\log\frac{\sqrt{p}}{K})$ with probability $1 - e^{-\Omega(m)}$.*

Proof. The proof follows from the proof of Lemma 3 above and Theorem 3.1 in (Krahmer & Ward, 2011). The proof of Lemma 3 requires the Johnson-Lindenstrauss guarantee for a set of size $2^{O(m)}$, and invoking Theorem 3.1 in (Krahmer & Ward, 2011), this is guaranteed to hold for the matrix FD.

The proof of Lemma 3 also requires $\|FD\|_{op} \leq \sqrt{n}$. This is also guaranteed by noting that

$$\|FD\|_{op} \leq \|FD\|_F \leq \sqrt{n},\ w.p. 1 - e^{-\Omega(m)}.$$

□

7.2. Code

Our source-code is available under the following url: https://github.com/giannisdaras/ilo.

Our code is implemented in PyTorch (Paszke et al., 2019). Our code is based on the following open-source implementations of StyleGAN-2: https://github.com/rosinality/stylegan2-pytorch, https://github.com/NVlabs/stylegan2. We also draw inspiration from the open-source implementation of PULSE: https://github.com/tg-bomze/Face-Depixelizer. A Tensorflow (Abadi et al., 2016) implementation is in the works.

Our current repository includes:

- Detailed instructions on how to setup the environment and download the dependencies.
- Code for image pre-processing, such as random inpainting, interactive masks, noise addition, automatic face alignment, etc.
- Examples on how to run inpainting, denoising, super-resolution and compressed-sensing with circulant matrices for custom images.
- Code for out-of-distribution generation on 1000 ImageNet (Deng et al., 2009) classes using a robust classifier. We use a robust classifier from the `robustness` (Engstrom et al., 2019) library.
- Code for evaluating the performance of ILO and previous methods on Celeba-HQ (Liu et al., 2018; Lee et al., 2020).
- Tools to visualize performance and track experiments.
- Code for generating GIF files by collecting frames during the optimization.

Our code is GPU/CPU compatible.

7.3. Experimental details

We performed all our experiments on a single GPU. As mentioned in the paper, obtaining a solution for a single inverse problem requires less than a minute on a single 1080Ti. All the experiments can be reproduced in less than a day on a single GPU.

Unless mentioned otherwise, we use Adam (Kingma & Ba, 2014) optimizer with an initial learning rate of 0.1 for each layer. During a single layer optimization, learning rate ramps up linearly and is ramped down using a cosine scheduler, as proposed by (Karras et al., 2020).

Loss functions are changed for each task as explained in the paper. For all tasks, we use a geodesic loss with coefficient 0.01. For random inpainting, we use both MSE and LPIPS when we have more than 20% observed pixels, otherwise we only use MSE. When both MSE and LPIPS are used, we search co-efficients in the set $\{0.5, 1, 2, 5\}$ for each of the terms. For inpainting with continuous black boxes, we used both MSE and LPIPS. For the experiments of Figure 1 of the main paper, we used the same co-efficient for both MSE and LPIPS.

Our optimization algorithm is Projected Gradient Descent (Nocedal & Wright, 2006). First, we project each latent code to the unit sphere. Next, when optimizing over deeper layers, we use l_1 projection to stay close to the manifold induced by the previous layers. The projection in that case includes the solution of the previous layers, the latent codes (i.e. w_i) and the noises, (i.e. u_i). We tune seperately the l_1 radii for each one of the optimization variables and for each one of the layers. Empirically, we find that the following radii for the first four layers works decently for most of the tasks/images:

- Radius of noises: $300, 2000, 2000, 4000$.
- Radius of latent codes: $300, 500, 1000, 2000$.
- Radius of previous solutions: $500, 1000, 2000$.

Projection to the l_1 ball allows for optimization on deep layers of the generator (that is not possible without projections). By doing that we get better reconstruction that comes with the cost of increased number of optimization steps. Generally, tuning the radii for each layer is an especially difficult procedure. Even worse, these hyperparameters do not transfer across tasks. For the first four layers, we encourage the reader to use the parameters mentioned above.

To obtain the plots of Figure 2, we sampled (randomly) 5 images from Celeba-HQ and we reported the best score for each point on the horizontal axis over 5 different runs (25 runs in total for each method for each point in the plot) with different hyperparameters. The error bars are computed across the experiments for different images. For the plots of Figure 2, we searched over the following combinations of number of steps for each layer (starting from the first): $\{300, 200, 200, 100\}$, $\{300, 200, 100\}$, $\{300, 200, 200, 100, 50\}$, $\{50, 50, 50, 50, 500\}$, $\{100, 100, 100, 100, 100\}$. Each reported point is the average (across images) of the minimums of those runs.

7.4. Additional Experiments

In this section, we list additional Figures and Experiments that could not fit in the paper due to space limit.

Figure 5 shows that by combining MSE loss to a reference image and the classification probability, we can morph a given person to an ImageNet (Deng et al., 2009) class. We observe experimentally that better results are obtained by only using MSE and LPIPS loss during the first ILO rounds and only using the additional classification term in the deeper layers of the generator. An extra benefit of this method is that we can interpolate intermediate frames to see how actually a human face can be transformed to an imagenet class since the generator first matches the phase and then uses the classifier to alter it.

Figure 6 shows visual results for the task of denoising. As shown, ILO gives superior visual reconstructions and better actual performance comparing to the (adapted for denoising) PULSE and the classical BM3D method.

Finally, in order to show that our method can be successfully in other datasets as well, we perform inpainting experiments using a pre-trained StyleGAN-2 generators on cats. Results are shown in Figure 7.

Figure 5. Morphing using a classifier for Bull Frog class, keeping also a loss term for distance to a well-known machine learning researcher, with his permission.

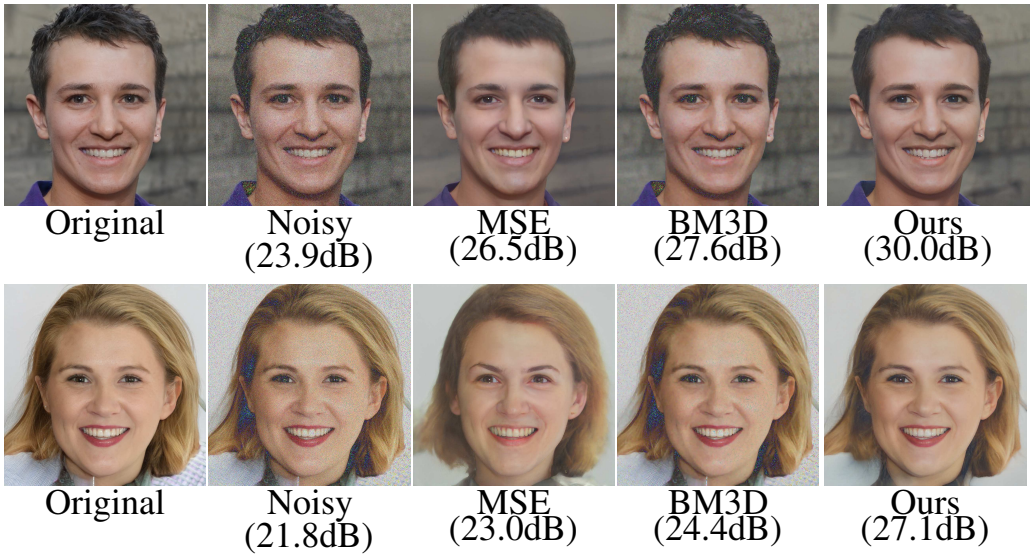

Figure 6. Results on the task of denoising. Gaussian noise ($\sigma = 25$, known) is added to the original image and recovered with various methods. The MSE images indicate the reconstructed images obtained by inverting the noisy image.

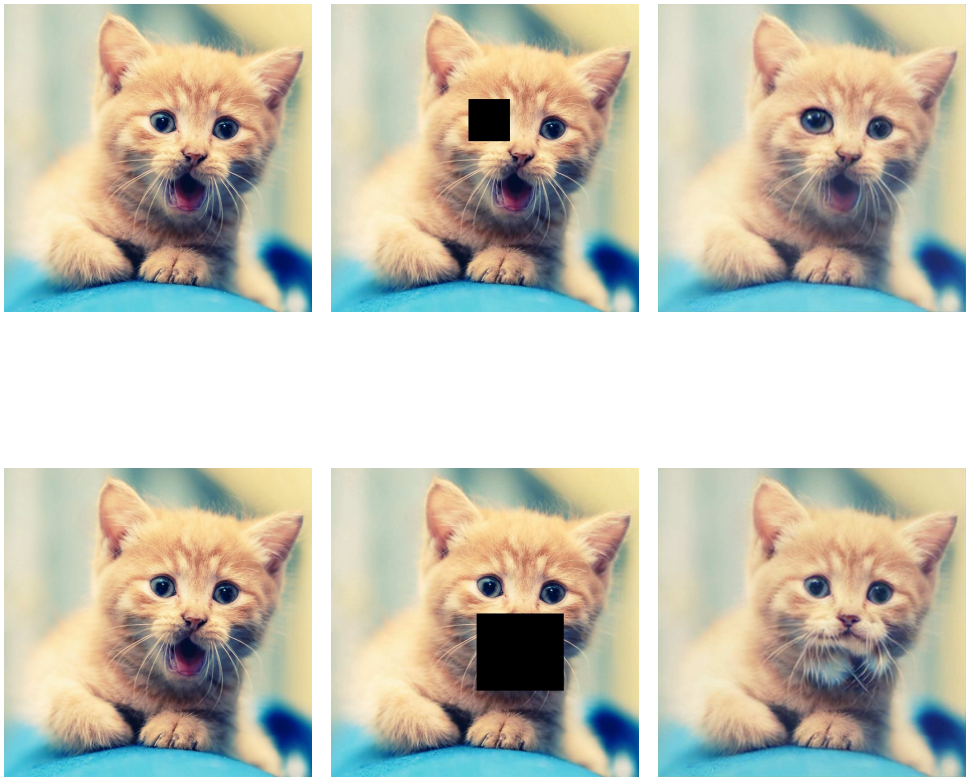

Figure 7. Inpainting using a StyleGAN trained to generate cat images. First column: Original image (never observed). Second column: Observed image. Third column: ILO reconstruction.

7.5. Ethical Considerations

Previous research has reported that StyleGAN leads to biased generations (Menon et al., 2020; Jain et al., 2020; Tan et al., 2020; Salminen et al., 2020). In practice, we observe that by extending the range of the generator we obtain more diverse generations. A similar finding has been reported by Abdal et al. (2019). Even though we observe less biased reconstructions, we encourage a lot more research on this topic.

Admittedly, our method makes the creation of DeepFakes (Korshunov & Marcel, 2018) easier, since it expands the range of the generator. Arguably, the technology behind DeepFakes is already very powerful so the negative effect of this work will be diminishing. An interesting topic of research is whether existing defences against DeepFakes (Matern et al., 2019; Güera & Delp, 2018; Nguyen et al., 2019; Yang et al., 2020) are robust to images that lie outside of the range of the GAN.

Experiments with a robust classifier combined with similarity to a reference image can be abused to generate images that are offensive in various ways. In this paper we are only exploring with what is possible, but future work should consider detecting and preventing such abuse.

7.6. Things that did not work

We share some negative results we encountered during the process of writing this paper. Our goal is to inform the research community about some methods that failed so that future research can avoid them, reformulate them or even contradict our findings. We also suggest ways to mitigate some of the issues we experienced.

First, we observed that joint optimization of all noise vectors leads to poor visual reconstructions. Even in cases where the MSE loss to the unobserved image was going down, joint-optimization of all noise vectors was giving blurry and/or unrealistic reconstructions. Thus we believe that expanding the generator space without sequential optimization and constraints fails since it makes it too powerful.

We also tried to establish a criterion on how many steps to run per layer. In practice, we observed that a working heuristic is to move to the next layer when the observed MSE error flattens. Even though this idea works well for the first layers, it can lead to unrealistic reconstructions when applied to deeper layers of the generator. To mitigate this issue, we choose very small radii when optimizing in deep layers. Tuning the hyperparameters (learning rates, number of steps and optimization radii) for each of the layers can be a particularly toilsome procedure. Sadly, we observed that these parameters do not generalize across different tasks (even though they mostly generalize across different images).

On the theoretical side, we tried (unsuccessfully) to obtain similar results for an l_2 dilation of the range of the first generator. The main bottleneck is the measurements bound; for the l_2 ball, we cannot avoid linear dependence on the intermediate dimension. It is not clear yet whether a similar result for the l_2 case could be proved. In practice, we did not observe significant difference between projecting in l_1 or l_2 balls. Moreover, it is known that l_1 projection encourages sparsity in some cases, e.g. see LASSO (Tibshirani, 1996). Establishing a connection between the l_0 and the l_1 solutions is left as future work.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

Abdal, R., Qin, Y., and Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441, 2019.

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3): 253–263, 2008.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In *International Conference on Machine Learning*, pp. 537–546. PMLR, 2017.

Chen, G.-H., Tang, J., and Leng, S. Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical physics*, 35(2):660–663, 2008.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising with block-matching and 3d filtering. In *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, volume 6064, pp. 606414. International Society for Optics and Photonics, 2006.

Daras, G., Odena, A., Zhang, H., and Dimakis, A. G. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dhar, M., Grover, A., and Ermon, S. Modeling sparse deviations for compressed sensing using generative models. In *International Conference on Machine Learning*, pp. 1214–1223. PMLR, 2018.

Duarte, M. F., Davenport, M. A., Takhar, D., Laska, J. N., Sun, T., Kelly, K. F., and Baraniuk, R. G. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, 2008.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Güera, D. and Delp, E. J. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6. IEEE, 2018.

Hand, P. and Voroninski, V. Global guarantees for enforcing deep generative priors by empirical risk. In *Conference On Learning Theory*, pp. 970–978. PMLR, 2018.

Hand, P., Leong, O., and Voroninski, V. Phase retrieval under a generative prior. *arXiv preprint arXiv:1807.04261*, 2018.

Hegde, C., Duarte, M. F., and Cevher, V. Compressive sensing recovery of spike trains using a structured sparsity model. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Hinrichs, A. and Vybíral, J. Johnson-lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011.

Jain, N., Olmo, A., Sengupta, S., Manikonda, L., and Kambhampati, S. Imperfect imaganation: Implications of gans exacerbating biases on facial data augmentation and snapchat selfie lenses. *arXiv preprint arXiv:2001.09528*, 2020.

Kabkab, M., Samangouei, P., and Chellappa, R. Task-aware compressed sensing with generative adversarial networks. In *AAAI*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00453. URL http://dx.doi.org/10.1109/CVPR.2019.00453.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.00813. URL http://dx.doi.org/10.1109/cvpr42600.2020.00813.

Keys, R. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Korshunov, P. and Marcel, S. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

Krahmer, F. and Ward, R. New and improved johnson–lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3): 1269–1281, 2011.

Lee, C.-H., Liu, Z., Wu, L., and Luo, P. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5549–5558, 2020.

Lei, Q., Jalal, A., Dhillon, I. S., and Dimakis, A. G. Inverting deep generative models, one layer at a time. In *NeurIPS*, 2019.

Liu, H., Jiang, B., Xiao, Y., and Yang, C. Coherent semantic attention for image inpainting. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. doi: 10.1109/iccv.2019.00427. URL http://dx.doi.org/10.1109/ICCV.2019.00427.

Liu, Z. and Scarlett, J. Information-theoretic lower bounds for compressive sensing with generative models. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 292–303, May 2020. ISSN 2641-8770. doi: 10.1109/jsait.2020.2980676. URL http://dx.doi.org/10.1109/JSAIT.2020.2980676.

Liu, Z., Luo, P., Wang, X., and Tang, X. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15 (2018):11, 2018.

Lucas, A., Iliadis, M., Molina, R., and Katsaggelos, A. K. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.

Lustig, M., Donoho, D., and Pauly, J. M. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Lustig, M., Donoho, D. L., Santos, J. M., and Pauly, J. M. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82, 2008.

Mardani, M., Gong, E., Cheng, J. Y., Vasanawala, S. S., Zaharchuk, G., Xing, L., and Pauly, J. M. Deep generative adversarial neural networks for compressive sensing mri. *IEEE transactions on medical imaging*, 38(1):167–179, 2018.

Matern, F., Riess, C., and Stamminger, M. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92. IEEE, 2019.

Menon, S., Damian, A., Hu, S., Ravi, N., and Rudin, C. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. doi: 10.1109/cvpr42600.2020.00251. URL http://dx.doi.org/10.1109/cvpr42600.2020.00251.

Mousavi, A., Dasarathy, G., and Baraniuk, R. G. A data-driven and distributed approach to sparse signal representation and recovery. In *International Conference on Learning Representations*, 2019.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., and Nahavandi, S. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Pajot, A., de Bezenac, E., and Gallinari, P. Unsupervised adversarial image reconstruction. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJg4Z3RqF7.

Pandit, P., Sahraee, M., Rangan, S., and Fletcher, A. K. Asymptotics of map inference in deep networks. *2019 IEEE International Symposium on Information Theory (ISIT)*, Jul 2019. doi: 10.1109/isit.2019.8849316. URL http://dx.doi.org/10.1109/ISIT.2019.8849316.

Park, J. Y., Smedemark-Margulies, N., Daniels, M., Yu, R., van de Meent, J.-W., and HAnd, P. Generator surgery for compressed sensing. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020. URL https://openreview.net/forum?id=s2EucjZ6d2s.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Pisier, G. Probabilistic methods in the geometry of banach spaces. In *Probability and analysis*, pp. 167–241. Springer, 1986.

Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., and Lee, S. Compressive sensing: From theory to applications, a survey. *Journal of Communications and networks*, 15(5): 443–456, 2013.

Raj, A., Li, Y., and Bresler, Y. Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. 2019.

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., and Cohen-Or, D. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020.

Salminen, J., Jung, S.-g., Chowdhury, S., and Jansen, B. J. Analyzing demographic bias in artificially generated facial pictures. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2020.

Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., and Madry, A. Image synthesis with a single (robust) classifier. *arXiv preprint arXiv:1906.09453*, 2019.

Shah, V. and Hegde, C. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018. doi: 10.1109/icassp.2018.8462233. URL http://dx.doi.org/10.1109/ICASSP.2018.8462233.

Song, G., Fan, Z., and Lafferty, J. Surfing: Iterative optimization over incrementally trained deep networks. *NeurIPS*, 2019.

Sudakov, V. N. Gaussian measures, cauchy measures and ε-entropy. In *Soviet Math. Dokl*, volume 10, pp. 310–313, 1969.

Sun, W. and Chen, Z. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020. ISSN 1941-0042. doi: 10.1109/tip.2020.2970248. URL http://dx.doi.org/10.1109/TIP.2020.2970248.

Sun, Y., Liu, J., and Kamilov, U. S. Block coordinate regularization by denoising. *NeurIPS*, 2019.

Sun, Y., Liu, J., and Kamilov, U. S. Block coordinate regularization by denoising. *IEEE Transactions on Computational Imaging*, 6:908–921, 2020. ISSN 2573-0436. doi: 10.1109/tci.2020.2996385. URL http://dx.doi.org/10.1109/TCI.2020.2996385.

Tan, S., Shen, Y., and Zhou, B. Improving the fairness of deep generative models without retraining, 2020.

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., and Lin, C.-W. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, Nov 2020. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.07.025. URL http://dx.doi.org/10.1016/j.neunet.2020.07.025.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tripathi, S., Lipton, Z. C., and Nguyen, T. Q. Correction by projection: Denoising images with generative adversarial networks. *arXiv preprint arXiv:1803.04477*, 2018.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wu, Y., Rosca, M., and Lillicrap, T. Deep compressed sensing, 2019.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Yang, C., Ding, L., Chen, Y., and Li, H. Defending against gan-based deepfake attacks via transformation-aware adversarial faces. *arXiv preprint arXiv:2006.07421*, 2020.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21 (12):3106–3121, 2019.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. Free-form image inpainting with gated convolution. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. doi: 10.1109/iccv.2019.00457. URL http://dx.doi.org/10.1109/ICCV.2019.00457.