REPRESENTATION ERROR REDUCTION IN COMPRESSIVE SENSING USING GENERATIVE MODELS WITH PIVOTAL TUNING

João Victor D. Sobrinho[1] and Miguel Elias M. Campista[1]
[1]Grupo de Teleinformática e Automação (GTA), PEE-COPPE, DEL-Poli, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

NOTE: Corresponding author: João Victor D. Sobrinho, joaosobrinho@gta.ufrj.br

Abstract *– Reducing the transmitted data volume is essential for implementing networks with resource-limited and energy-constrained devices. In this sense, compressive sensing becomes a powerful alternative as it moves the most computationally complex task to the central server node, in contrast to the traditional compression scheme. Recently a combination of compressive sensing and generative models has appeared, giving rise to Compressive Sensing using Generative Models (CSGM). Although CSGM reduce reconstruction errors, they introduce the so-called representation error. This paper proposes CSGM-Pivotal Tuning Inversion (CSGM-PTI), a technique based on model retraining in decompression time to reduce the representation error in CSGM. The idea of CSGM-PTI is to expand the scope of the generative model to include the desired signal. The results show robustness to noise and performance gains in signal reconstruction of up to 20% compared with Deep Image Prior (DIP), standalone CSGM, and Wavelet Thresholding (WT) techniques, all traditionally used in the related literature.*

Keywords – Compressed sensing, data compression, Internet of Things

1. INTRODUCTION

The increasing deployment of energy-limited devices introduces challenges regarding the implementation feasibility of Internet of Things (IoT)-based projects. During their lifecycle, these devices must take on sensing, processing, and transmitting data, with each one of these tasks introducing new challenges. In particular, transmitting ever-growing volumes of sensed data is troublesome since this is one of the most energy-consuming tasks for IoT devices [1]. For this reason, different approaches for dealing with highly frequent transmissions have been proposed in the literature, such as data aggregation [2], fusion [3], and compression [4].

The Compressed Sensing (CS) approach has turned out to be an advantageous alternative for data compression in scenarios where sensing devices have constraints on energy or computational resources. Such advantages arise from the ability of CS to invert the relational costs between compression and decompression. Unlike traditional data compression approaches, which are costly for sensing devices, with CS this task consists simply of undersampling a target signal. Because of this, sensing devices with low resource availability can reduce both compression and transmission costs, delegating the signal reconstruction to a more powerful participant in the cloud, for instance. Fig. 1 illustrates CS in an IoT scenario where a group of low-power sensing devices measures a target object. By utilizing CS, these devices can undersample their measurements and send them to a more energetically-capable cloud environment, which can perform a costly reconstruction step and generate the required signal.

Besides reducing the costs related to transmission and compression, CS also reduces the number of measurements the sensing device needs to perform, which can be another positive aspect in challenging scenarios such as seismic data acquisition. In such a context, seismic surveyors must plan the deployment of both seismic wave sources and receivers in specific locations. Using a CS-based approach, one can reduce the number of needed sources, mitigating the impacts caused by generating seismic waves in the surrounding environment. Alternatively, CS also enables surveyors to deploy fewer receivers, which can reduce costs and complexity.

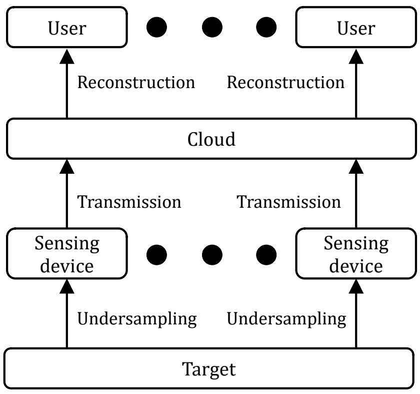


Figure 1 – Example IoT architecture using CS.

Despite the favorable cost-benefit relationship between compression and decompression, classic CS presents limitations that can become an obstacle for large-scale implementations. One such limitation is the non-trivial requirement of compressibility, i.e., the existence of some transformation that results in a sparse collection of measurements when applied to the original signal. For sparsity, we mean a signal with many near-zero coefficients [5]. In contrast, Compressed Sensing using Generative Models (CSGM) changes the previous requirement by the existence of a Generative Model (GM) that has the target signal within its reach [6]. Thus, CSGM can only recover an undersampled signal if a generative model, such as a Generative Adversarial Network (GAN), can generate the signal of interest. Results from the literature show that CSGM can achieve a similar reconstruction error for sampling rates 10 times lower than classic CS, even dismissing the sparsity requirement [6].

Despite the advantages gained by CSGM, its use introduces new challenges. Using GMs during the decompression step leads to the risk of choosing a GM that cannot generate a signal similar to the original one. The difference between the original signal and the best approximation obtained with a GM is known in the literature as a representation error [7].

This paper proposes CSGM-PTI[1], a technique to reduce the representation error in CSGM using Pivotal Tuning Inversion (PTI) [8]. Since pivotal tuning was originally developed for image editing, we adapt PTI to the CSGM context. Consequently, PTI becomes an additional step to the traditional CSGM reconstruction algorithm. Firstly, one must obtain the pivot value or the GM input that generates the most similar signal to the target. Then, using the pivot point as a fixed input, the GM's network is adjusted through retraining. As a result, the model tuning extends the GM's generative capabilities, potentially including in it the target signal.

Experimental results show that CSGM-PTI is successful at reducing the representation error. Improvements of up to 5% and 20% Peak Signal-to-Noise ratio (PSNR) can be observed when comparing the proposed technique with Deep Image Prior (DIP) [9] and traditional CSGM, respectively. CSGM-PTI also performs better regarding perceptual similarity losses and is shown to be robust to different levels of additive Gaussian noise. These results advocate for implementing the proposed technique in scenarios like IoT since the reduction in transmitted data promotes energetic gains to the sensor devices. Furthermore, a comparison between CSGM-PTI and Wavelet Thresholding (WT) indicates that, although it has been outperformed considering the PSNR and Structural Similarity Index Measure (SSIM) metrics, our proposal achieves better Learned Perceptual Image Patch Similarity (LPIPS) for high compression ratios. This result suggests that the proposed technique is capable of preserving key image features and structures.

This paper is organized as follows. Section 2 discusses related work. Section 3 provides background on compressive sensing and generative modeling theory. Then, Section 4 introduces our proposal, CSGM-PTI, which uses pivotal tuning as a solution to representation error mitigation. Section 5 presents the methodology employed and the experimental results observed. We conclude this paper and draw future plans in Section 6.

2. RELATED WORK

The limitations of the GM inherited by CSGM are the focus of different proposals in the literature aiming to reduce the representation error. A technique from the literature which is particularly relevant to our proposal is called Deep Image Prior

[1]A preliminary version of our paper was published in SBRC 2023, a Brazilian national conference. This version can be accessed at `http://www.gta.ufrj.br/ftp/gta/TechReports/SAC23.pdf`.

(DIP) [9]. DIP is a technique for solving inverse problems in imaging processing, such as denoising, super-resolution, inpainting, and CSGM. It can successfully solve such issues by manipulating an Artificial Neural Network (ANN) with randomized weights, i.e., with an untrained network. More specifically, in the context of CSGM, the network input is fixed at a random value, while its weights are adjusted during decompression time. These adjustments are such that the measurements from the network output must match the measured samples from the original signal.

This process can be understood as a generalization of the CSGM-PTI approach that, instead of relying on pretrained generative models as a prior requirement, takes advantage of the network structure for signal reconstruction. Since DIP eliminates the training requirement, the bias associated with the training data and process is also eliminated. Consequently, by definition, DIP does not have a representation error. Nevertheless, by refraining from using highly expressive generative models openly available in the literature, such as StyleGAN2 [10], one expects the signal reconstruction problem to become more difficult.

Daras et al. propose Intermediate Layer Optimization (ILO) [11] as an alternative technique for reducing the representation error in generative models. Similar to DIP and PTI, ILO is characterized by the inversion-time tuning of a generator network to expand its reach, consequently including the target signal. Nevertheless, ILO differs from the other techniques by the network retraining implementation. After an initial inversion of the generative network G, resulting in the input vector z, G's input is fixated at z, and its layers are sliced, dividing the generator into intermediate layers. The network tuning is then executed on the separated intermediate layers, such that one obtains intermediate representations, $z_i = G_i(z_{i-1})$, more adequate to the target signal. Each intermediate layer uses these representations in successive inversion and tuning steps. This process results in a generative network divided into subnetworks with inputs adjusted to the specific purpose of expanding the entire network reach, such that it becomes able to generate the target signal.

In contrast with CSGM, Asim et al. propose the use of invertible generative models in place of GANs [7]. The authors argue that this architecture would eliminate the representation error since constructing invertible generative models is based on mathematically invertible networks. However, training these models can be challenging and costly. Additionally, since invertible generative models are not as popular in the literature as GANs, the availability of high-quality pretrained models is not as common.

In this paper, we propose an adaptation to the pivotal tuning technique for the context of compressive sensing. Unlike other techniques, CSGM-PTI adapts the idea of retraining the prior generator network to the CSGM context in addition to executing the traditional CSGM approach. Besides this, CSGM-PTI prevents the retraining from inflicting distortions on the generative model, which could degrade the reconstruction performance for different signals by employing local regularization. A summary of the comparison between our proposal with the related techniques is presented in Table 1.

Table 1 – Comparison between our proposal (CSGM-PTI) with related work.

Paper	**Prior**	**Inversion**	**Tuning**
[9]	Untrained Net	No	Prior
[11]	GM	Yes	Prior Slices
[7]	Invertible GM	Yes	No
Ours	GM	Yes	Prior

3. BACKGROUND

This section summarizes the theoretical background regarding compressive sensing and generative models. In the following section, we propose the CSGM-PTI technique.

3.1 Compressive sensing

The compressive sensing paradigm combines efficient data acquisition and transmission in a single step, modeled by a measurement matrix $A \in \mathbb{R}^{M \times N}$, such that a signal $x \in \mathbb{R}^N$ is related to a collection of measurements $y \in \mathbb{R}^M$ through Equation *(1)*,

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

where $\epsilon \in \mathbb{R}^M$ represents some additive Gaussian noise, a product of the measuring process. Additionally, it must be true that $M \ll N$, since for data compression, one must achieve a relevant dimensionality reduction from x to y. This relation results in an underdetermined system of equations, i.e., infinitely many signals x can satisfy Equation *(1)*, for fixed A, y, and ϵ. Although this problem seems unsolvable, considering the requirement that x be k-sparse in some basis ψ, the problem can be reformulated as shown in Equation *(2)*,

$$y = A\psi^{-1}s + \epsilon, \tag{2}$$

where s is the k-sparse representation of x in ψ. With the imposed sparsity requirement for s, the infinitely many solutions x can be restricted to the signal that simultaneously satisfies the equation and maximizes the vector sparsity in ψ. This leads to a configuration where a unique solution to the previously underdetermined system becomes viable upon solving the optimization problem shown in Equation *(3)*.

$$\min_s ||s||_0 : ||y - A\psi^{-1}s||_2 \leq \epsilon, \tag{3}$$

where $||\cdot||_2$ is the l_2 norm and $||\cdot||_0$ is a notation that represents the cardinality of the subset of all non-zero entries in s, known as the l_0 norm.

Finding a vector with minimal l_0 norm is an NP-hard problem [12] though. Therefore, a direct l_0 minimization approach is excessively costly, making it unviable. As a solution to this issue, Candes et al. show that the l_1 minimization is a suitable approximation when the measurement matrix A satisfies a set of properties, such as the Restricted Isometry Property (RIP) [13, 14]. This property defines the construction rule the matrix A, which models the measurement process, must follow. Despite these restrictions, the l_1 minimization works fine for common matrices such as Gaussian random matrices.

The strategy of exchanging the NP-hard problem of l_0 minimization with l_1 minimization is known in the literature as basis pursuit [15], presented in Equation *(4)*.

$$\min_s ||s||_1 : ||y - A\psi^{-1}s||_2 \leq \epsilon. \tag{4}$$

By using the l_1 norm, which is convex, the basis pursuit procedure makes it possible to recover the compressed signal through convex optimization, which can be solved through highly efficient linear programming algorithms.

3.2 Generative models

Generative Adversarial Networks (GANs) [16] can generate data points without obtaining an explicit definition of the probability distribution. To that end, a GAN learns how to generate data by simultaneously training two Artificial Neural Networks (ANNs). Note that the ability to explicitly obtain the data distribution is avoided, as this involves costly and complex procedures.

The central characteristic that defines a GAN is an adversarial training strategy. More specifically, GANs are built from two competing neural networks, G and D, that are trained using an adversarial method. While G, the generator network, aims to generate samples that convincingly represent the training data probability distribution, the discriminator network, D, is trained to classify G's output as either generated or real samples. In doing so, the discriminator's cost function is defined to maximize the probability that D's output provides the correct label, i.e., real or generated sample. In contrast, G's training aims to minimize the probability that D is correct. Through this adversarial training scheme, one expects that, as D becomes more accurate at detecting the correct label, G is incentivized to generate more convincing samples. As a result, G becomes capable of mapping a random noise vector as input z into a sample from the probability distribution of the training data p_x.

3.3 Compressive sensing using generative models (CSGM)

The traditional compressive sensing approach assumes a sparse representation of the target signal for original signal recovery from an undersampled set of measurements. In contrast, the model proposed by Bora et al. [6] does not depend on signal compressibility for recovery guarantees. The Compressive Sensing Using Generative Models (CSGM) makes a different assumption, i.e., it considers that the target signal x lies within the reach of some generative model $G : \mathrm{R}^k \to \mathrm{R}^N$, with input $z \in \mathrm{R}^k$ and output $G(z) \in \mathrm{R}^N$.

By exchanging the use of signal sparsity with a generative model, the target signal can be recovered through the minimization problem shown in Equation *(5)*:

$$\min_z ||y - AG(z)||_2, \quad (5)$$

where the target signal is recovered by obtaining the model input z such that its compressed model output $AG(z)$ matches the compressed signal y. In the case of GANs, which are the focus of this paper, the solution to this optimization problem is called GAN inversion [17].

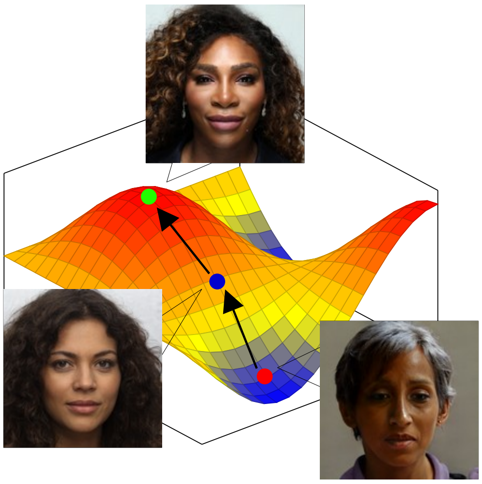


Figure 2 – GAN inversion steps.

Fig. 2 illustrates the steps involved in the inversion of a GAN. The surface S presented in the figure represents the collection of all possible outputs of a GAN with a three-dimensional output, i.e., for any input vector z, the GAN will generate $G(z) \in S$. Since the input vector z that minimizes Equation *(5)* is yet unknown, one could initialize it at a random value, as represented by the red dot. Note that at this point, the GAN output completely differs from the target signal, shown in Fig. 2 as the green dot. However, through consecutive iterations of an optimization algorithm, such as the gradient descent [18], it is expected that the output of the generator network can gradually become more similar to the target signal by minimizing the difference between the compressed signal and the compressed output. This approximation is illustrated in Fig. 2 by the blue dot. If the assumption that the chosen generative model can generate the target signal holds, then the optimization algorithm converges to the optimal solution denoted by the green dot.

4. OUR PROPOSAL: CSGM-PTI

The pivotal tuning technique, or PTI, was initially developed for applications in GAN-based image editing. PTI extends the inversion procedure, taking advantage of the input $z = z^{'}$ attained during the GAN inversion step as a pivot point, around which the generator network goes through precise adjustments. The resulting generator has extended capabilities, potentially including the previously unreachable target signal, by tuning the generator around the pivot point. Since the GAN generator is a neural network, the tuning process refers to a network retraining under a modified loss function. While fixing the network input at the previously obtained pivot point, the retraining aims to generate a new set of weights $\theta^{'}$ that minimizes a combination of both per-pixel and perceptual losses. Equation *(6)* presents the main portion of the loss function used in PTI:

$$L_{pivot}(\hat{x}, x) = \lambda_{LPIPS} \cdot LPIPS(\hat{x}, x) + \lambda_{l_2} \cdot ||\hat{x} - x||_{l_2}, \quad (6)$$

where $\hat{x} = G(z^{'}; \theta^{'})$ is the estimated signal generated by the network with weights $\theta^{'}$. The presented portion of the total loss, L_{pivot} is built as the combination of two metrics, the Learned Perceptual Image Patch Similarity (LPIPS) [19] and the l_2 norm. By defining two hyperparameters λ_{LPIPS} and λ_{l_2}, it is possible to balance the influence of each corresponding metric on the loss function. Both metrics compare the signal estimate $\hat{x}$ and the target signal but with different approaches. While the l_2 norm is a traditional per-pixel metric used in signal reconstruction techniques, it has limitations regarding image comparison. In contrast, perceptual metrics such as LPIPS aim to represent a human notion of similarity between images.

In addition to the loss component described in Equation *(6)*, PTI proposes including a local regularizer term. Such incorporation makes it possible to maintain the generative model's global behavior. Thus, while tuning the generative model around its pivot point, the PTI technique uses the regularizer term to prevent the degradation of the generative model's capabilities of generating other signals. Equation *(7)* describes the complete loss function minimized during PTI:

$$L_{total}(\hat{x}, x) = L_{pivot}(\hat{x}, x) + L_{local}(\hat{x}, x). \quad (7)$$

Thus, the pivotal tuning step consists of retraining G to generate a signal that, when compressed by the measurement matrix A, corresponds to the measurements y. The optimization problem in Equation *(8)* formulates the procedure:

$$\min_{\theta'} L_{pivot}(\hat{x}, x) + L_{local}(\hat{x}, x) : ||y - \hat{x}||_{l_2} \leq \epsilon, \quad (8)$$

where the solution is obtained by varying the network weights θ' so that the total loss function is minimized and the signal estimate corresponds to the observations y. However, in compressive sensing, one does not have direct access to the original signal x since x is compressed during acquisition, resulting in a reduced set of measurements y. Thus, the original PTI approach is incompatible with compressed sensing implementations since it directly compares $\hat{x}$ with x.

In this paper, we propose a solution to the incompatibility problem by adapting PTI to use it in the context of CSGM. This is achieved by altering the original loss function of PTI, consequently originating the CSGM-PTI technique. More specifically, CSGM-PTI proposes that the loss function compares the compressed versions of both target and estimate signals, making direct access to x unnecessary. Additionally, since LPIPS was developed for measuring the perceptual similarity between uncompressed signals, its use can create distortions during training. Therefore, the adapted loss function depends only on the l_2 norm as a metric. Equation *(9)* presents the adapted loss function:

$$L_{cs}(\hat{y}, y) = \lambda_{l_2} \cdot ||\hat{y} - y||_{l_{l2}} + L_{local}(\hat{y}, y), \quad (9)$$

where L_{cs} is the adapted loss, y is the compressed original signal, and $\hat{y}$ is the compressed estimate signal, being y and $\hat{y}$ obtained by the same sampling procedure. The signal can then be reconstructed by solving the optimization problem shown in Equation *(10)*:

$$\min_{\theta'} \lambda_{l_2} ||\hat{y} - y||_{l_{l2}} + L_{local}(\hat{y}, y) : ||\hat{y} - y||_{l_{l2}} \leq \epsilon. \quad (10)$$

The proposed CSGM-PTI aims to improve CSGM by reducing the representation error associated with the generator. This type of error can be observed when the target signal is outside the generative model's reach, i.e., the generator cannot generate the signal of interest. Whenever this occurs, the traditional GAN inversion achieves the best approximation within the generator's reach, illustrated as the green dot in Fig. 3.

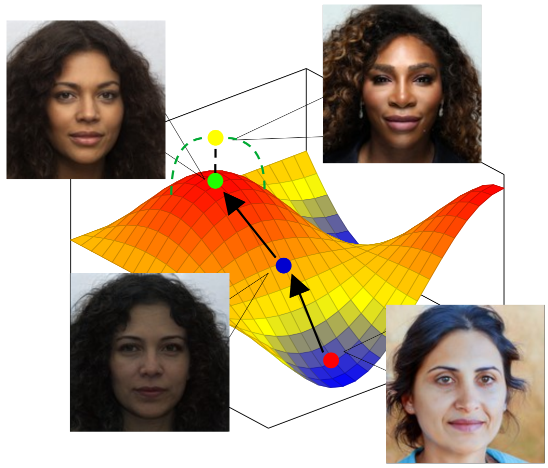


Figure 3 – Representation error in GAN inversion using pivotal tuning as a solution.

After obtaining the best approximation with the generator, the proposed technique is applied, retraining the generator network such that it becomes able to generate the target signal. In Fig. 3, this process is illustrated by a precise deformation of the output surface around the pivot point to include the yellow dot. Note that the CSGM-PTI approach is a two-step process: the GAN inversion traditionally used in CSGM in addition to the pivotal tuning. This composition is promising if we consider the used approaches for representation error mitigation. CSGM-PTI derivation takes advantage of advances in two popular research areas: GAN inversion and neural network training.

Table 2 – Variable glossary.

Variable	Definition
y	Measured signal
A	Measurement matrix
x	Target signal
G(z;θ)	Generative model with input z and model parameters θ
λ	Scalar hyperparameter
$\hat{y}$	Measured signal estimated by G
$\hat{x}$	Target signal estimated by G

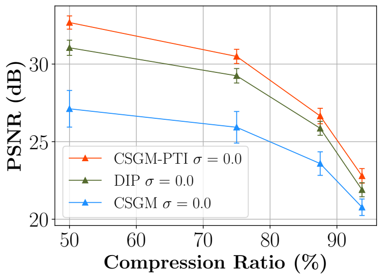
5. EXPERIMENTS

Experiments comparing the reconstruction quality of CSGM-PTI with other traditional techniques were executed to evaluate the performance gains

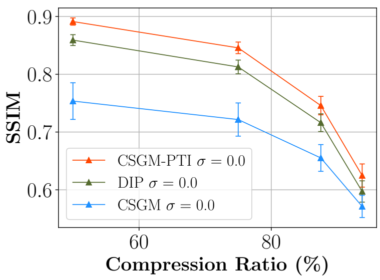
Table 3 – PSNR, SSIM and LPIPS values for 93.75% compression using different techniques under different levels of additive Gaussian noise.

σ	CSGM-PTI			CSGM			DIP		
	PSNR	**LPIPS**	**SSIM**	**PSNR**	**LPIPS**	**SSIM**	**PSNR**	**LPIPS**	**SSIM**
0	22.8 ± 0.5	0.24 ± 0.02	0.62 ± 0.02	20.77 ± 0.5	0.33 ± 0.02	0.57 ± 0.02	21.9 ± 0.5	0.33 ± 0.02	0.6 ± 0.02
1	21.7 ± 0.4	0.3 ± 0.01	0.53 ± 0.02	20.08 ± 0.5	0.4 ± 0.03	0.49 ± 0.02	20.93 ± 0.4	0.43 ± 0.03	0.48 ± 0.02
2	19.3 ± 0.3	0.44 ± 0.01	0.34 ± 0.01	18.63 ± 0.3	0.5 ± 0.04	0.37 ± 0.02	18.69 ± 0.3	0.58 ± 0.03	0.3 ± 0.01

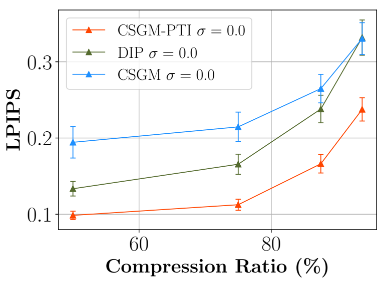
of our proposal. We considered the Deep Image Prior (DIP), presented in Section 2, given its analogous construction to CSGM-PTI. Our proposed technique can be understood as a specific implementation of the more generalized approach in DIP, with the difference being the use by CSGM-PTI of pretrained models for generating an initial approximation. Given its wide employment in the literature and image generation capacity, the generative model chosen for the experiments was NVIDIA's StyleGAN2 [10] trained on a 256x256 resolution version of the human face image dataset Flickr-Faces-HQ (FFHQ) [20]. FFHQ is a dataset with 70 000 high-quality images extensively used in the literature, particularly for evaluating GANs. To avoid reusing images from those used for GAN training, we use the celebA-HQ [21] dataset for experiments. The celebA-HQ dataset was chosen as a set of out-of-distribution human face images for a fair evaluation of the proposed technique. Following this, results from comparative experiments between CSGM-PTI and other traditional techniques, DIP, CSGM, and WT, are presented.



(a) PSNR.



(b) SSIM.



(c) LPIPS.

Figure 4 – Reconstruction error comparison between CSGM, CSGM-PTI and DIP with no noise addition for different compression ratios and different metrics.

Figures 4a, 4b, and 4c present the results from an experiment comparing CSGM, CSGM-PTI, and DIP without noise under three different metrics: PSNR, SSIM, and LPIPS. While higher PSNR and SSIM mean lower reconstruction error, lower LPIPS indicates more similar images before and after compression. All three metrics are shown for images compressed under compression ratios of 50%, 75%, 87.75%, and 93.75%. Additionally, each data point corresponds to the average reconstruction error from the compression of one hundred images from the celebA-HQ dataset. For all combinations of metric and compression ratio, it is possible to observe performance gains for CSGM-PTI when compared with both techniques, reaching 5% and 20% higher PSNR compared with DIP and CSGM, respectively. On top of this, it is possible to observe that the proposed technique presents smaller error bars for all metrics. This indicates that CSGM-PTI produces a more accurate output when compared with the other techniques. It is also noticeable that, as compression ratios increase, all techniques reach their recovery limits, and the performance gains achieved with CSGM-PTI become more subtle.

A similar experiment was carried out by comparing the reconstruction error of the techniques using two different levels of additive Gaussian noise. Table 3 presents the reconstruction error values ob-

(a) Original **(b)** CSGM **(c)** DIP **(d)** CSGM-PTI

Figure 5 – Images resulting from the process of compression and decompression using CSGM, DIP and CSGM-PTI. Compression ratio set at 75% and $\sigma = 0$.

(a) Original **(b)** CSGM **(c)** DIP **(d)** CSGM-PTI

Figure 6 – Images resulting from the process of compression and decompression using CSGM, DIP and CSGM-PTI. Compression ratio set at 75% and $\sigma = 2$.

(a) Original **(b)** CSGM **(c)** DIP **(d)** CSGM-PTI

Figure 7 – Images resulting from the process of compression and decompression using CSGM, DIP and CSGM-PTI. Compression ratio set at 93% and $\sigma = 0$.

(a) Original **(b)** CSGM **(c)** DIP **(d)** CSGM-PTI

Figure 8 – Images resulting from the process of compression and decompression using CSGM, DIP and CSGM-PTI. Compression ratio set at 93% and $\sigma = 2$.

tained by performing a 93.75% compression with additive noise with standard deviation $\sigma = 0$, $\sigma = 1$ and $\sigma = 2$. It is possible to observe that for high compression ratios, the proposed technique outperforms CSGM and DIP for almost all noise and error metric combinations, falling short only in the case of $\sigma = 2$ for the SSIM metric. These results show that CSGM-PTI is at least as robust to additive Gaussian noise as the other techniques.

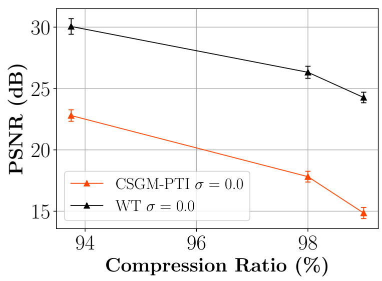
Figures 5, 6, 7, and 8 show a set of example outputs from CSGM, DIP, and CSGM-PTI under different compression ratios and additive noise. Note that all CSGM images look similar but represent someone other than the original person. This is a consequence of the generator's limitation in producing the original image, which introduces representation errors. One can also observe that, for higher compression ratios, both CSGM-PTI and DIP become heavily distorted, which corroborates the PSNR and SSIM approximation of the two techniques seen in figures 4a and 4b. Although both get distorted, CSGM-PTI's distortion better preserves shapes and overall facial structure, leading to the lower LPIPS growth observed in Figure 4c. The preservation of broad structures, even for high compression ratios, can be valuable for downstream tasks such as image classification.

Figures 9a, 9b, and 9b present a comparison between CSGM-PTI and WT, which is a classic transform-based approach for data compression. In contrast to the compressed sensing strategy, WT does not compress by undersampling the target signal. Instead, WT filters the signal's Discrete Wavelet Transform (DWT) coefficients such that only the most energetic are kept. Hence, compression is achieved not by taking fewer measurements but by reducing the number of coefficients that need to be transmitted. This compression technique presents similarities with parts of the ubiquitous image compression standard JPEG 2000.

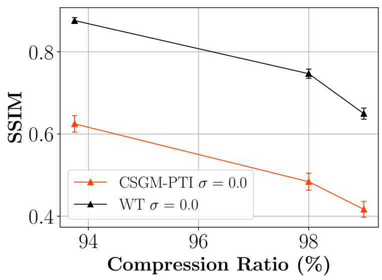
The results in figures 9a and 9b indicate that WT outperforms CSGM-PTI for PSNR and SSIM metrics. However, Figure 9c shows that, for high compression ratios, the proposed technique achieves better LPIPS performance. Considering that LPIPS is a perceptual similarity metric based on the activations of a deep neural network, this result suggests that CSGM-PTI better maintains key features for high compression ratios when compared with WT, which tends to uniformly distort the signal.

6. CONCLUSION

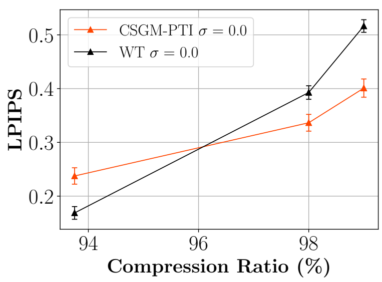
This paper proposed a technique for reducing representation error in compressive sensing using generative models. Compressive Sensing Using Generative Models - Pivotal Tuning Inversion (CSGM-PTI) is the adaptation of a latent-based image editing technique, allowing the use of CSGM in the compressive sensing context without direct access to the target signal. Consequently, CSGM-PTI reduces the volume of data transmitted by devices with low energy and computation availability, such as devices in IoT settings. Since CSGM-PTI is based on the compressive sensing paradigm, the technique promotes, besides transmission cost reductions, a simplified compression procedure at the encoder. This comes at a higher reconstruction cost for the decoder. However, these costly steps can be transferred to devices with higher resource availability. In IoT, decoding is typically conducted at resource-rich devices in the cloud. CSGM-PTI combines two distinct procedures, GAN inversion



(a) PSNR



(b) SSIM



(c) LPIPS

Figure 9 – Reconstruction error comparison between CSGM-PTI and WT with no noise addition for different compression ratios and different metrics.

and neural network training, placing it in a privileged position regarding scientific advances. These components are highly relevant in the literature, and improvements in their respective areas can promote better performance for CSGM-PTI.

Comparative experiments were conducted between CSGM-PTI and traditional techniques from the literature. The results indicated that, for different compression ratios and additive noise, the proposed technique outperformed Deep Image Prior (DIP) and traditional CSGM on both objective and perceptual similarity metrics for reconstruction errors. A different comparison was carried out between CSGM-PTI and Wavelet Thresholding (WT). Although CSGM-PTI has fallen short in both PSNR and SSIM, promising results for the LPIPS metric suggested that the proposed technique promotes better maintenance of the structure, as well as key features from the original image.

The carried-out analysis suggested a few directions for future research, which can promote new developments for the proposed technique. For instance, the study of how CSGM-PTI's induced distortion influences tasks such as classification and how this influence compares with the effects caused by traditional compression techniques. Additionally, evaluating the impact of the technique's reconstruction time on communications is essential for gauging the effect of implementing CSGM-PTI in real-world applications.

ACKNOWLEDGEMENT

This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It was also supported by CNPq, FAPERJ Grants E-26/211.144/2019 and E-26/202.689/2018, and FAPESP Grant 15/24494-8.

REFERENCES

[1] Soroush Abbasian Dehkordi, Kamran Farajzadeh, Javad Rezazadeh, Reza Farahbakhsh, Kumbesan Sandrasegaran, and Masih Abbasian Dehkordi. "A survey on data aggregation techniques in IoT sensor networks". In: *Wireless Networks* 26 (2020), pp. 1243–1263. DOI: 10.1007/s11276-019-02142-z.

[2] Mandeep Kaur and Amit Munjal. "Data aggregation algorithms for wireless sensor network: A review". In: *Ad Hoc Networks* 100 (2020), p. 102083. DOI: https://doi.org/10.1016/j.adhoc.2020.102083.

[3] Behrouz Pourghebleh, Negin Hekmati, Zahra Davoudnia, and Mehrdad Sadeghi. "A roadmap towards energy-efficient data fusion methods in the Internet of Things". In: *Concurrency and Computation: Practice and Experience* 34 (2022), e6959. DOI: https://doi.org/10.1002/cpe.6959.

[4] Keleadile Lucia Ketshabetswe, Adamu Murtala Zungeru, Bokani Mtengi, Caspar K. Lebekwe, and S. R. S. Prabaharan. "Data Compression Algorithms for Wireless Sensor Networks: A Review and Comparison". In: *IEEE Access* 9 (2021), pp. 136872–136891. DOI: 10.1109/ACCESS.2021.3116311.

[5] Yan Wu, Mihaela Rosca, and Timothy Lillicrap. "Deep Compressed Sensing". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. June 2019, pp. 6850–6860.

[6] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. "Compressed Sensing using Generative Models". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Aug. 2017, pp. 537–546.

[7] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. "Invertible generative models for inverse problems: mitigating representation error and dataset bias". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. July 2020, pp. 399–409.

[8] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. "Pivotal Tuning for Latent-Based Editing of Real Images". In: *ACM Trans. Graph.* 42 (2022), pp. 1–13. DOI: 10.1145/3544777.

[9] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *International Journal of Computer Vision* 128 (2020), pp. 1867–1888. DOI: 10.1007/s11263-020-01303-4.

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. *Analyzing and Improving the Image Quality of StyleGAN*. 2020. arXiv: 1912.04958 [cs.CV].

[11] Giannis Daras, Joseph Dean, Ajil Jalal, and Alex Dimakis. "Intermediate Layer Optimization for Inverse Problems using Deep Generative Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. July 2021, pp. 2421–2432.

[12] B. K. Natarajan. "Sparse Approximate Solutions to Linear Systems". In: *SIAM Journal on Computing* 24 (1995), pp. 227–234. DOI: 10.1137/S0097539792240406.

[13] Emmanuel J. Candès. "The restricted isometry property and its implications for compressed sensing". In: *Comptes Rendus. Mathématique* 346 (2008), pp. 589–592. DOI: 10.1016/j.crma.2008.03.014.

[14] E.J. Candes, J. Romberg, and T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on Information Theory* 52 (2006), pp. 489–509. DOI: 10.1109/TIT.2005.862083.

[15] Shaobing Chen and D. Donoho. "Basis pursuit". In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Vol. 1. Oct. 1994, pp. 41–44.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[17] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. "GAN Inversion: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2023), pp. 3121–3138. DOI: 10.1109/TPAMI.2022.3181070.

[18] Haskell B. Curry. "The method of steepest descent for non-linear minimization problems". In: *Quarterly of Applied Mathematics* 2 (1944), pp. 258–261. DOI: 10.1090/qam/10667.

[19] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 586–595.

[20] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2021), pp. 4217–4228. DOI: 10.1109/TPAMI.2020.2970919.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2018. arXiv: 1710.10196 [cs.NE].

AUTHORS

João Victor D. Sobrinho is an undergraduate student of electronics and computer engineering at UFRJ. His interests include compressive sensing, signal processing, inverse problems in imaging, Internet of things, and machine learning.

Miguel Elias M. Campista has been an associate professor at the Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, since 2010. Campista received his D.Sc. degree in electrical engineering from UFRJ in 2008 and spent 2012 in the LIP6 lab at Sorbonne Université, Paris, France, as an invited professor. He is a former affiliate member of the Brazilian Academy of Sciences and is currently an associate editor of Annals of Telecommunications, Springer. His major research interests include network and data science, machine learning, data security, and cloud and fog computing.

