Single Pixel Imaging Based on Generative Adversarial Network Optimized With Multiple Prior Information

Shida Sun, Qiurong Yan, *Member, IEEE*, Yongjian Zheng, Zhen Wei, Jian Lin, and Yilin Cai

***Abstract*—Reconstructing high-quality images at low measurement rate is one of the research objectives for single-pixel imaging (SPI). Deep learning based compressed reconstruction methods have been shown to avoid the huge iterative computation of traditional methods, while achieving better reconstruction results. Benefiting from improved modeling capabilities under the constant game of generation and identification, Generative Adversarial Networks (GANs) has achieved great success in image generation and reconstruction. In this paper, we proposed a GAN-based compression reconstruction network, MPIGAN. In order to obtain multiple prior information from the dataset and thus improving the accuracy of the model, multiple Autoencoders are trained as regularization terms to be added to the loss function of the generative network, and then adversarial training is performed with a multi-label classification network. Experimental results show that our scheme can significantly improve reconstruction quality at a very low measurement rate, and reconstruction results are better than the existing network.**

***Index Terms*—Compressed sensing (CS), single pixel imaging (SPI) system, photon counting, generative adversarial networks (GAN), deep learning, multiple prior information.**

I. Introduction

COMPRESSED sensing theory proposes that if the data is sparse in a certain transform domain, the original signal can always be reconstructed [1]–[3]. This theory has important applications in many fields, and single-pixel imaging (SPI) is one of them [4]–[9]. In the SPI system, the target is first imaged on the DMD, then a series of masks is loaded onto the digital micro-mirror device (DMD) to spatially modulate the image of object, and finally a single-pixel detector with no positional resolution receives the modulated light intensity. The image of the target can be recovered using an algorithm based on compressed sensing theory. Single-pixel imaging is promising to be widely used in medical diagnosis, astronomical observation and spectral measurement.

Manuscript received 16 May 2022; revised 10 June 2022; accepted 16 June 2022. Date of publication 22 June 2022; date of current version 4 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62165009, and in part by the National Natural Science Foundation of China under Grant 61865010. *(Corresponding author: Qiurong Yan.)*

The authors are with the School of Information Engineering, Nanchang University, Nanchang 330031, China (e-mail: xueluo365@gmail.com; yanqiurong@ncu.edu.cn; brephoszyj@163.com; 6105119141@email.ncu.edu.cn; 6105119144@email.ncu.edu.cn; 768276454@qq.com).

Digital Object Identifier 10.1109/JPHOT.2022.3184947

Nevertheless, SPI requires plenty of time in sampling which limits its application in real-time scenarios, thence reconstructing high-quality images at a low measurement rate is one of the research objectives. According to different principles, reconstruction algorithms can be divided into traditional mathematical algorithms and deep-learning-based schemes. Traditional mathematical algorithms have high reconstruction accuracy, and the typical representatives are OMP [10], GPSR [11], BCS [12], TVAL3 [13] and so on. However, this type of methods has the problem of slow reconstruction speed, and it is difficult to adapt to more complex features. The machine-learning-based schemes have faster reconstruction speed and higher imaging accuracy, which can dynamically learn features of different sampled image. With the rapid development of machine learning, many excellent solutions have been derived [14]–[17]. In fact, other networks for compressive reconstruction can also be applied to single-pixel imaging tasks, which essentially solve the inverse problem from low-dimensional signals to high-dimensional signals. In 2016, Kulkarni *et al.* proposed a CNN-based algorithm named ReconNet [18], which trained an end-to-end model by stacking convolutional blocks. In the same year, Yao *et al.* proposed a DR2Net model with a residual structure [19], which could effectively reduce the number of parameters and improved the training speed while further improving the reconstructed image details. However, the above-mentioned networks have problems such as simple structure and poor anti-disturbance ability of the network.

Generative Adversarial Network (GAN) is a deep learning model proposed by Goodfellow *et al.* [20]. The model includes a generation network and a discrimination network, which produce fairly good outputs through mutual game learning between the two. GAN, as a new framework for generative model through adversarial optimization, has been studied for reconstruction of compressed sampled images. Wu *et al.* proposed DCS [21], which combined compressed sensing and meta-learning to improve the reconstruction effect, so that it can optimized the operation efficiency and iteration number of GANs. Kabkab *et al.* [22] proposed CSGAN, which used GAN as structural constraints instead of sparsity to reduce the measurement rate requirements. However, there are several problems in the above studies: lack of image quality evaluation indicators, single prior information, and differences between generated images and actual images. Overall, both the lack of prior information and

the uncertainty of the output limit the development of GANs for single-pixel imaging tasks.

The essence of compressive reconstruction is to use prior information to solve underdetermined equations. Prior information includes sparse prior [23], [24], edge information prior, Fourier prior [25], and Autoencoder prior [26], etc. Introducing strong prior information is equivalent to adding constraints to the feature space, which can enhance the quality of reconstructed images effectively. Recent studies reported that adding prior information to GAN could improve reconstruction accuracy. Yang *et al.* proposed DAGAN [27], which combined the prior information of multiple domains such as time and frequency to train MRI images with GAN and achieved higher MRI image reconstruction accuracy, showing us the potential for prior information. Autoencoder can learn the representation of the input information by taking the input information as the learning target. Zhang *et al.* [28] proposed REDAEP, which used an Autoencoder to refine image priors on high-dimensional data and introduces a mathematical iterative method to compute imaging, indicating that Autoencoder priors have a strong driving effect on image feature learning. The GAN is prone to generate images that are inclined to a specific category when the prior information is inclined in a certain direction. That is, the problem of mode collapse may occur when the label training is introduced. We believe that one of the main solutions to the above problems is to increase the diversity of prior information, as sufficiently diverse information can assist GAN to get out of the local optimum trap.

We regard including multi-scale and multi-level prior information to the GAN to pay attention to a more complex feature environment, thereby improving the image reconstruction accuracy. Therefore, in this study, multiple Autoencoders focusing on different types of information are trained as prior information. On this basis, multi-label training and several other optimizations methods are introduced to further improving the imaging accuracy. In summary, the contributions of this study are mainly in three aspects:

1) We proposed a GAN-based compression reconstruction networks (MPIGAN and MPIGAN+). Different from existing studies, we add multiple Autoencoder priors and auxiliary classifier losses to the loss function, resulting in higher imaging accuracy and faster convergence.
2) Through theoretical analysis, we illustrate the working principle of the Autoencoder regular term in GAN, and prove that the Autoencoder regular term can accelerate the gradient descent of GAN.
3) We test MPIGAN and MPIGAN+ on multiple datasets and the single-pixel imaging system, respectively, and obtain better imaging results than other networks. The experiments demonstrate that our algorithm can reconstruct the main features of images even at a very low measurement rate.

II. The Proposed Network

In the single-pixel imaging, we denote the compression measurement process as (1).

$$y = Sx + q \tag{1}$$

where $x \in \mathbb{R}^{N\times 1}$ is source signal, $S \in \mathbb{R}^{M\times N}$ is the sampling matrix, M is the number of measurements, N is the source image dimension, q is the noise generated during the sampling, and $y \in \mathbb{R}^{M\times 1}$ is the measured value matrix. In general, the total number of measurements (M) is much smaller than the total number of pixels (N). This means that to restore x from y, one needs to solve an inverse problem with far fewer equations than the number of variables. With symbol G as the generator of a GAN, we can express this inverse problem as (2).

$$\hat{x} = G(y) \tag{2}$$

where $\hat{x} \in \mathbb{R}^{N\times 1}$ is the reconstructed images. The output of the generator is sent to the discriminator, which continuously compares the difference between $\hat{x}$ and x. The generator's job is to generate images that can fool the discriminator, and the discriminator's job is to differentiate between the real images and the generator's output. The better the discriminator is trained, the less likely it is to be fooled by the generator's output.

A. Network Structure

In this section, we introduce the network structure of MPIGAN (Fig. 1). MPIGAN consists of a sampling network, a generator, a discriminator, and multiple Autoencoders. We set a fully-connected layer as sampling network, which will be jointly trained with generator and discriminator. Previous networks usually use random matrices for image sampling, but it is easy to lose key feature information and affect the accuracy of reconstructed images. Therefore, we use a fully-connected layer instead of a random matrix, which is jointly optimized with the image reconstruction network. Related work shows [29] that fully-connected layers for sampling not only ensure sufficient weights to reconstruct images, but also reduce training time. The generator is responsible for restoring the images from the measurements to its original dimensions, upsampling the images multiple times in a row. The measured value is first reconstructed by a fully-connected layer (the first layer of generator), and then enters the generator to obtain the reconstructed images $\hat{x}$. The original images x and the reconstructed images $\hat{x}$ are fed into the discriminator, respectively, and the binary classification output and auxiliary classification output is obtained. As shown in Fig. 1, the discriminator produces two losses, with $l^{Binary}(x)$ corresponding to the binary classification loss and $l^{Multi}(x)$ corresponding to the auxiliary classification loss. The denoising Autoencoder (DAE) is a simple mapping network to represent latent image features, consists of two fully-connected layers. Descriptions of each part of the MPIGAN are detailed in the Section II.B.

B. Loss Function Design

With letter D as the discriminator, we denote optimization process of the GAN as (3).

$$\min_G \max_D \mathbb{E}_x [\log D(x)] + \mathbb{E}_{\hat{x}} [\log (1 - D(\hat{x}))] \tag{3}$$

Here, $D(x)$ and $D(\hat{x})$ is the binary classification probability. However, the loss function with min-max form is not conducive

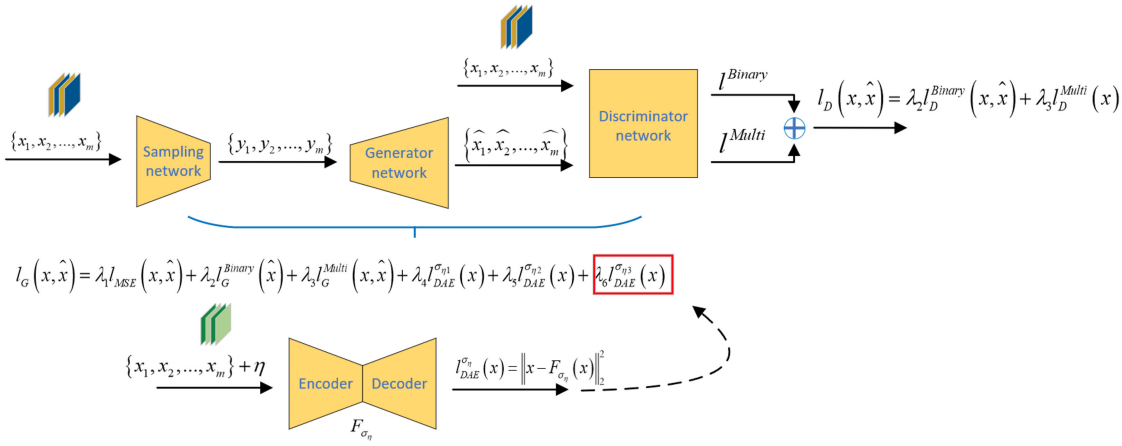

Fig. 1. Algorithm for our proposed GAN-based network with multiple priors (MPIGAN). Specifically, MPIGAN consists of an analog sampling network, a generator, a discriminator, and multiple Autoencoders. Notedly, we set a total of three pretrained Autoencoders with variances of 0.05, 0.07 and 0.1, respectively, and each of them has the same structure.

to using code implementation, so we rewrite the function using cross entropy:

$$l_G^{Binary}(\hat{x}) = \mathbb{E}_{\hat{x}}\left[H\left([1,0],\left[D\left(\hat{x}\right),1-D\left(\hat{x}\right)\right]\right)\right] \tag{4}$$

$$l_D^{Binary}(x,\hat{x}) = \mathbb{E}_x\left[H\left([1,0],\left[D\left(x\right),1-D\left(x\right)\right]\right)\right] + \mathbb{E}_{\hat{x}}\left[H\left([0,1],\left[D\left(\hat{x}\right),1-D\left(\hat{x}\right)\right]\right)\right] \tag{5}$$

where $l_G^{Binary}(x)$ and $l_D^{Binary}(x,\hat{x})$ represent the loss functions of the generator and discriminator, respectively, and $H(p,q) = -\sum_i p_i \log q_i$ represents the cross entropy. In this way, (3) can be transformed into two minimization problems. Because (4) and (5) are implemented based on real/fake classification, we call it binary classification loss. However, such a simple classification has not paid attention to the deep image features, so it is difficult for discriminator to distinguish real and fake images with complex features. One of the problems with this is that the loss function of GANs is difficult to train. Inspired by ACGAN [30], other than retaining the original real/fake classifier, we add auxiliary classifier under the discriminator.

Studies have shown that adding multi-label classifier could help improving the network learning effect [30]–[33], which could alleviate the problem of single type and poor diversity of images generated by GAN, too. [34] further discussed how the multi-label information works. Discriminator with multi-label classifier is regarded as an implicit target class model, refining each generated image towards one of the specific classes, which will help improve the image's quality. As such, the use of multi-label information can enhance the network ability to gain complex features and enhance stability of loss function. We incorporate auxiliary classifiers to help MPIGAN to extract deep features, enriching the details of reconstructed images. We assumed images to possess a total of *K* categories, so the auxiliary classification loss can be expressed as:

$$l_G^{Multi}(x,\hat{x}) = \mathbb{E}_{\hat{x}}\left[H\left(T\left(x\right),D_C\left(\hat{x}\right)\right)\right] \tag{6}$$

$$l_D^{Multi}(x) = \mathbb{E}_x\left[H\left(T\left(x\right),D_C\left(x\right)\right)\right] \tag{7}$$

where $D_C(x) = \{D_{C1}(x), D_{C2}(x), \ldots, D_{Ci}(x), \ldots, D_{CK}(x)\}$, $T(x) = \{T_1(x), T_2(x), \ldots, T_i(x), \ldots, T_K(x)\}$ with $T_i(x) = 1$ if $x \in class\ i$ and $T_i(x) = 0$ if $x \notin class\ i$. $l_G^{Multi}(x,\hat{x})$ and $l_D^{Multi}(x)$ are the auxiliary classifier losses for the generator and discriminator, respectively. $D_C(\hat{x})$ and $D_C(x)$ are the class label prediction outputs of the discriminator for the reconstructed images and the original images, assigning a probability to each class. Since (4) to (7) are designed based on the adversarial structure of GAN, we call them adversarial loss.

However, optimizing a GAN using adversarial loss alone would require a large number of iterative steps and much time to finally produce high-quality images. With the growth of training data size and network size, it is unrealistic to obtain the best results in a limited time. Usually, using MSE as the loss can get images of better quality in a short time. Therefore, we consider adding the MSE loss to speed up the training of the GAN:

$$l_{MSE}(x,\hat{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{8}$$

We noticed that MSE tends to produce blurry output and is hard to recover high frequency contents in the image, which is bad for reconstructing images with complex features. We thus consider ways to introduce prior information so as to supply high frequency contents into the reconstructed images.

Bigdeli *et al.* [26] further studied the properties of DAE and proposed that DAE $F_{\sigma_\eta}(x)$ have the following mathematical relationship with the image data density *p*:

$$F_{\sigma_\eta}(x) = \frac{\mathbb{E}_\eta[p(x-\eta)(x-\eta)]}{\mathbb{E}_\eta[p(x-\eta)]} \tag{9}$$

where η is Gaussian noise with variance σ_η. As observed by (9), output of the self-encoder is equivalent to the local average of the image x. It is further proved that when the noise has a Gaussian distribution, the Autoencoder error is proportional to the log-likelihood gradient of the smoothing density:

$$F_{\sigma_\eta}(x) - x = \sigma_\eta^2 \nabla \log \mathbb{E}_\eta[p(x-\eta)] \tag{10}$$

The Autoencoder error guides the feature space to the confidence interval that maximizes the probability density distribution of the image, so that the Autoencoder finally learns the probability distribution of the training images. Inspired by (10) that introducing the Autoencoder as prior information into the network can supplement the missing details in the reconstructed image, we thus use two-norm error as the regularizer term:

$$l_{DAE}^{\sigma_\eta}(x) = \left\| x - F_{\sigma_\eta}(x) \right\|_2^2 \tag{11}$$

Each of Autoencoder error denoted by (11) is a vector pointing in the direction of real image's maximum probability density, supplementing missing details in reconstructed images. Different Autoencoders are trained by different noise variance, forcing the MPIGAN to focus on the distribution of image data at different scales and targeting at introducing multi prior information. Combining the above loss functions, we propose the loss function of the MPIGAN:

$$\begin{aligned} l_G(x, \hat{x}) = \lambda_1 l_{MSE}(x, \hat{x}) + \lambda_2 l_G^{Binary}(\hat{x}) + \lambda_3 l_G^{Multi}(x, \hat{x}) \\ + \lambda_4 l_{DAE}^{\sigma_{\eta 1}}(x) + \lambda_5 l_{DAE}^{\sigma_{\eta 2}}(x) + \lambda_6 l_{DAE}^{\sigma_{\eta 3}}(x) \end{aligned} \tag{12}$$

$$l_D(x, \hat{x}) = \lambda_2 l_D^{Binary}(x, \hat{x}) + \lambda_3 l_D^{Multi}(x) \tag{13}$$

Where λ_1 to λ_6 are the weight coefficients to balance each loss, respectively.

C. Network Training

The training process can be divided into two parts: Autoencoder training and MPIGAN training.

Autoencoder consists of encoder and decoder. The training of the autoencoder is a process to optimize the encoder weights θ_e and decoder weights θ_d. To ensure that the Autoencoder error is kept to a minimum stably during training, we pretrained each autoencoder with different noise variances, and froze its weights when MPIGAN is training. We denote the images input with noise as (14).

$$x^{noise} = x + \eta \tag{14}$$

Each autoencoder is trained to minimize the (15).

$$\left\| x - F_{\sigma_\eta}\left(x^{noise}\right) \right\|_2^2 \tag{15}$$

In our experiment, we trained each Autoencoder with 100 epochs. The details of autoencoder training are listed in Algorithm 1.

The training of MPIGAN includes generator training and discriminator training. We then freeze all pretrained autoencoder's weights and reload them when MPIGAN is training. The discriminator weights and three DAE errors are frozen when we train the generator using (12) as loss function. At the same time, we will freeze the generator weights when discriminator is training using (13) as loss function. The generator and discriminator are trained alternately until the minimum weight of generator θ_G is obtained. The details of MPIGAN training are listed in Algorithm 2.

Importantly, since the loss function consists of multiple parts, it is necessary to balance the contribution of each part during training to ensure the stability of the training. According to the observation of the experimental results, we have adjusted the coefficients of the regular terms in time. We found that the stability of the network is the best when the values of each part of the loss function are of the same order of magnitude.

Algorithm 1: Autoencoder training.

Initialize encoder weights θ_e and decoder weight θ_d.
for number of epoch **do**:
  for number of batches **do**:
    -Sample minibatch of m samples $\{x_1, x_2, \ldots, x_m\}$.
    -Compute $x_i^{noise} = x_i + \eta$.
    -Compute coded value $F_{\sigma_\eta}(x_i^{noise})$.
    -Update the θ_e and θ_d to minimize the loss (15):
      $\underset{\theta_e, \theta_d}{\operatorname{argmin}} \|x_i - F_{\sigma_\eta}(x_i^{noise})\|_2^2$
  end for
end for
Get θ_e and θ_d.

Algorithm 2: MPIGAN algorithm.

MPIGAN training process:
Restore the Autoencoder weights: θ_e and θ_d.
Initial the weights of sampling network, generator and discriminator: S, θ_G and θ_D.
for number of epoch **do:**
  for number of batches **do:**
    - Sample minibatch of m samples $\{x_1, x_2, \ldots, x_m\}$.
    - Compute $y_i = Sx_i$, $\hat{x}_i = G(y_i)$, $F_{\sigma 1}$, $F_{\sigma 2}$ and $F_{\sigma 3}$.
    - Freeze the discriminator weights.
    - Update the generator and sampling network to minimize $l_G(x_i, \widehat{x_i})$ loss.
    $\underset{\theta_G, \theta_S}{\operatorname{argmin}}\ \lambda_1 l_{MSE}(x_i, \hat{x}_i) + \lambda_2 l_G^{Binary}(\hat{x}_i) +$
    $\lambda_3 l_G^{Multi}(x_i, \hat{x}_i)$
    $+\lambda_4 l_{DAE}^{\sigma_{\eta 1}}(x_i) + \lambda_5 l_{DAE}^{\sigma_{\eta 2}}(x_i) + \lambda_6 l_{DAE}^{\sigma_{\eta 3}}(x_i)$
    - Freeze the generator weights.
    - Update the discriminator to minimize $l_D(x_i, \hat{x}_i)$ loss.
      $\underset{\theta_D}{\operatorname{argmin}}\ \lambda_2 l_D^{Binary}(x_i, \hat{x}_i) + \lambda_3 l_D^{Multi}(x_i)$
  end for
end for
MPIGAN testing process:
Restore the weights of sampling network and generator: S, θ_G.
- Compute the reconstructed images:
  $\hat{x}_i = G(Sx_i)$
Get $\hat{x}_i$.

D. Working Principle of the Autoencoder Regular Term

In subsequent experiments, we found that the images generated by MPIGAN had higher accuracy than that of other networks, and obtained higher PSNR and SSIM scores. In this

section, we try to analyze the effectiveness of the regularization term of the Autoencoder from a theoretical point of view.

The essence of the compression imaging task is to solve the maximum a posteriori problem $P(\hat{x}|y)$, and solve the images $\hat{x}$ that satisfies the maximum probability distribution by observing the sample value y. Regardless of the network structure, even if we adopt GAN as the basic structure of the network, the above basic theory will not be changed. Since the Autoencoder (11) we used as the regular term is pre-trained, we regard that optimization for the MPIGAN loss function can be understood from a probabilistic perspective as solving the maximum a posteriori problem:

$$P\left(\hat{x}|y, \underset{F_{\sigma_\eta}}{\operatorname{argmin}} \left\|x - F_{\sigma_\eta}\left(x+\eta\right)\right\|_2^2\right) \tag{16}$$

That is, the generator solves the maximum probability distribution of *x* under the constraints of the Autoencoder.

Notedly, our improved maximum a posteriori problem is fundamentally different from the original problem: the independent variable space of the original problem is taken from the complete set of images, and the feature space of the improved problem are from images and Autoencoders. This means that the generator not only learns features from the images, but also from the Autoencoders. In order to illustrate the role of the regular term of the Autoencoder in GAN vividly, we take out $l_G^{Binary}(\hat{x}) + L_{DAE}^{\sigma_\eta}(x)$ from MPIGAN loss function, and then solve the gradient of it about *x*:

$$\begin{aligned} &-\frac{\partial\left(l_G^{Binary}\left(\hat{x}\right)+l_{DAE}^{\sigma_\eta}\left(x\right)\right)}{\partial x} \\ &= -\frac{\partial H\left(\left[1,0\right],\left[D\left(\hat{x}\right),1-D\left(\hat{x}\right)\right]\right)}{\partial x} - \frac{\partial\left\|x-F_{\sigma_\eta}\left(x\right)\right\|_2^2}{\partial x} \end{aligned} \tag{17}$$

If we use *L* to represent the logit vector and *q* to be the sigmoid output, then:

$$\begin{aligned} &-\frac{\partial H\left(p,q\left(L\right)\right)}{\partial L} = \frac{\partial\sum_i p_i \log q_i\left(L_i\right)}{\partial L} \\ &= \frac{\partial\sum_i p_i \log\frac{1}{1+e^{-L}}}{\partial L} = \frac{-\sum_i p_i \partial\log\left(1+e^{-L_i}\right)}{\partial L} \\ &= -\sum_i p_i\left(1-\frac{1}{1+e^{-L_i}}\right) = -p\left(1-q\left(L\right)\right) \end{aligned} \tag{18}$$

We substitute the conclusion of the (18) into (17):

$$\begin{aligned} &-\frac{\partial\left(l_G^{Binary}(\hat{x})+l_{DAE}^{\sigma_\eta}(x)\right)}{\partial x} = -\left(1-D\left(\hat{x}\right)\right)\frac{\partial\hat{x}}{\partial L(x)}\frac{\partial L\left(x\right)}{\partial x} \\ &+\left(2\sigma_\eta^2\nabla log\mathbb{E}_\eta\left[p\left(x-\eta\right)\right]\right)\left(\frac{\partial F_{\sigma_\eta}\left(x\right)}{\partial x}-1\right) \end{aligned} \tag{19}$$

where $(1-D(\hat{x}))$ is the GAN's generator gradient and $(2\sigma_\eta^2\nabla log\mathbb{E}_\eta[p(x-\eta)])$ is the Autoencoder gradient. As observed from (19), we find the gradients of the log likelihood were added after the adversarial loss. This means that adding the Autoencoder error is equivalent to adding a vector pointing in the direction of maximum probability density, helping the network to converge faster and improving the accuracy of the imaging.

As such, the existence of the Autoencoder regularizer gradient succinctly connects the prior information of the maximum probability distribution of the image with the category gradient. The discriminator plays a role in directly guiding the gradient descent of the generator, both intuitively and theoretically.

E. MPIGAN+

We think that the regular term can be set as an Autoencoder other than DAE. Replacing the regular term of the Autoencoder established for other tasks will help the network learn the feature space of other tasks, which can also reduce the impact of data disturbance. For example, we cover the input samples with evenly spaced 3×3 mask blocks, and train the Autoencoder to fill the mask blocks with higher degrees of freedom to improve the robustness of the MPIGAN model. The main modifications are as follows.

1) Masked Autoencoder F_M

We overlay the images by multiple 3×3 black blocks with equally space, thereby removing part of the feature information of the images, forcing the Autoencoder to invert samples that approximate *x* with less information.

$$l_{Mask}\ \left(x\right) = \left\|x - F_M\left(x\right)\right\|_2^2 \tag{20}$$

$F_M(x)$ is trained to minimize (21):

$$\left\|x - F_M\left(M\left(x\right)\right)\right\|_2^2 \tag{21}$$

where $M(\sim)$ means mask operation.

1) Revolved Autoencoder F_R

We flip the images and feed it to the Autoencoder. The aim of designing this Autoencoder is to avoid images sampled from different directions being incorrectly identified in real situations and thus improving the robustness of MPIGAN.

$$l_{Revolve}\ \left(x\right) = \left\|x - F_R\left(x\right)\right\|_2^2 \tag{22}$$

$F_R(x)$ is trained to minimize (23):

$$\left\|x - F_R\left(R\left(x\right)\right)\right\|_2^2 \tag{23}$$

where $R(\sim)$ means revolve operation.

Combined with the above discussion, we modify the generator loss of MPIGAN:

$$\begin{aligned} l_G\ \left(x,\hat{x}\right) = {} & \lambda_1\, l_{MSE}\left(x,\hat{x}\right) + \lambda_2 l_G^{Binary}\left(\hat{x}\right) + \lambda_3 l_G^{Multi}\left(x,\hat{x}\right) \\ & + \lambda_4 l_{DAE}^{\sigma_{\eta 1}}\left(x\right) + \lambda_5 l_{Revolve}\left(x\right) + \lambda_6 l_{Mask}\left(x\right) \end{aligned} \tag{24}$$

The above model is called **MPIGAN+**.

III. Experiments and Results

In this chapter, we use several comparative experiments to illustrate the advantages of MPIGAN in single-pixel imaging tasks from multiple aspects. The batch-size is set as 64. The epoch is set as 100 for Mnist dataset, 20 for CelebA dataset.

TABLE I
NETWORK STRUCTURE OF THE MPIGAN

Net	Operation	Kernel	Padding	Filters/Units	Activation
Sampling	Dense	N/A	N/A	$M\times1$	LeakyRelu
Generator	Dense	N/A	N/A	$N\times1$	LeakyRelu
	Conv	[7,7]	Same	64	LeakyRelu
	Conv	[1,1]	Same	1	LeakyRelu
	Conv	[7,7]	Same	64	LeakyRelu
	Conv	[3,3]	Same	1	LeakyRelu
	Conv	[7,7]	Same	64	LeakyRelu
	Conv	[1,1]	Same	1	LeakyRelu
	Conv	[7,7]	Same	64	LeakyRelu
	Conv	[3,3]	Same	1	LeakyRelu
Discriminator	Conv	[1,3]	Same	32	Relu
	Conv	[3,1]	Same	64	Relu
	Conv	[3,3]	Same	128	Relu
	Conv	[3,3]	Same	256	Relu
	Dense	N/A	N/A	1	Sigmoid
	Dense	N/A	N/A	K	Softmax
Autoencoder	Dense	N/A	N/A	$M\times1$	LeakyRelu
	Dense	N/A	N/A	$N\times1$	LeakyRelu

The image size of Mnist is set as 32×32, and the image size of CelebA is set as 64×64. The optimizer is set as Adam with the learning rate of 1e-3. The measurement rate can be denoted as:

$$MR\ (Measurement\ Rate) = \frac{M}{N} \tag{25}$$

The network settings we used in our experiments are shown as Table I. Notedly, increasing the number of layers or setting other parameters may achieve better experimental results.

A. The Ablation Studies in MPIGAN

The experiments in this part focus on the contribution of each part in MPIGAN loss function. We denote MPIGAN networks with and without adversarial part losses by "MAD" (MSE, Adversarial loss and DAEs, representing $l_{MSE}(x,\hat{x}) + l_G^{Binary}(\hat{x}) + l_G^{Multi}(x,\hat{x}) + l_{DAE}^{\sigma_\eta}(x)$) and "MD" (MSE and DAEs, representing $l_{MSE}(x,\hat{x}) + l_{DAE}^{\sigma_\eta}(x)$), respectively. The parameters used in the training process at the same measurement rate are consistent, and the noise standard deviations used in different DAE training are 0.05, 0.07 and 0.1, respectively. We set the six coefficients of λ_1 to λ_6 from (12) and (13) to be 1, 0.001, 0.001, 0.05, 0.05, 0.05 in our experiments, and fine-tune them according to the actual situation. The dataset adopts Mnist, one part used as training set and the other part used to test network performance. Experiment results are shown as Table II. Bold numbers indicate the best value in that measurement rate. Under the conditions of the same number of regularizer terms (same row) and the same measurement rate, the network with adversarial structure (MAD) performs better. With the same network (same column) and the same measurement rate, the network with more Autoencoder regularizers performs better. Overall, the PSNR score is proportional to the number of regularization terms added, indicating that the network can absorb multi-scale prior information from autoencoders with different standard deviations.

To further illustrate the contribution of each part of the loss function clearly, we list the PSNR scores for different combinations of loss functions in Table III (MR = 0.01). Compared with combinations without adversarial loss, combinations with adversarial loss terms achieved higher PSNR scores (at most 0.42 dB higher). The PSNR with the Autoencoder regularizer added is higher than that without the Autoencoder regularizer (at most 0.481 dB higher). Adding multiple Autoencoder regularizers achieved a higher PSNR than adding a few autoencoder regularizers (at most 0.172 dB higher). As the result of experiments, autoencoder regularizers can effectively improve the quality of reconstructed images and enhance the features learning ability of GAN. Fig. 2 shows the situation of Table II in a line graph (MR = 0.01). As observed by Fig. 2, we note that regularization terms of the Autoencoder works better in GAN framework than it does in the standard network. There may be two main reasons, one is that the increase in the number of parameters improves the accuracy of network fitting, and the other is that the working principle of the Autoencoder in GAN is different from that in standard networks. In the framework of GAN, the loss function gradient of the generator is mainly contributed by the probability distribution output from discriminator. The main role of the Autoencoder can be considered as accelerating the GAN to assign the gradient to a certain category and indirectly accelerating the convergence of the loss function. In the standard network, the loss function gradient of the generator is mainly contributed by losses such as MSE. At this point, the Autoencoder does not make an obvious connection to the category of the image. Therefore, we believe that the connection between Autoencoders and image categories in GAN is the key to improving quality of reconstructed images.

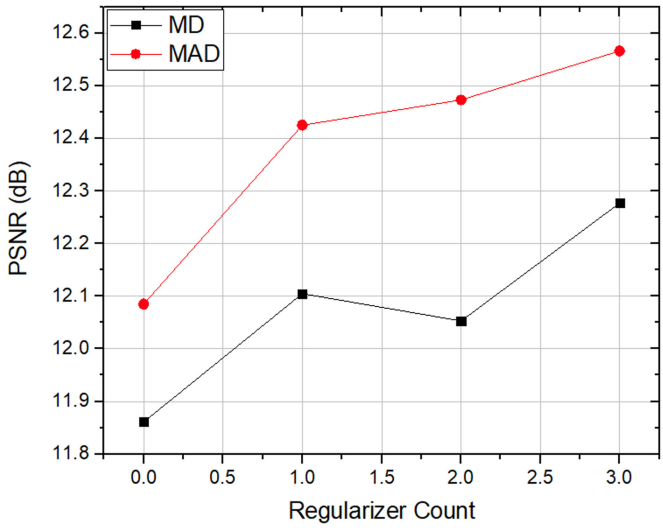

Fig. 2. PSNR values for testing images by different algorithms at different number of Autoencoder regularization terms (MR = 0.01). The abscissa is the number of regular items, and the ordinate is the PSNR score. The red and black lines represent adversarial structures and no adversarial structures, respectively, and the results show that networks with adversarial structures perform better in imaging. Points on the same polyline represent PSNR scores after adding different amounts of Autoencoder regularization terms. In general, the polyline shows a monotonically increasing trend, indicating that the superposition of the autoencoder effectively improves the amount of prior information.

B. Comparison With Other Networks

In this section, we compare MPIGAN with other algorithms, proving that our network has advantages in image reconstruction task. We set the standard deviations of the three DAEs in the

TABLE II
PSNR OF MPIGAN WITH DIFFERENT LOSS FUNCTION COMBINATIONS UPON THE MNIST AT DIFFERENT MEASUREMENT RATES

Number of DAEs	MR=0.1		MR=0.05		MR=0.01		MR=0.005	
	MD	MAD	MD	MAD	MD	MAD	MD	MAD
0	24.102	**25.695**	**19.719**	19.241	11.861	**12.085**	9.653	**10.082**
1	24.356	**26.107**	**20.350**	20.185	12.105	**12.425**	9.725	**10.06**
2	24.534	**26.008**	20.390	**20.395**	12.053	**12.473**	9.873	**9.929**
3	24.163	**26.061**	20.481	**20.823**	12.277	**12.566**	9.844	**10.124**

TABLE III
THE ABLATION STUDY OF MPIGAN IN MR 0.01

MSE	√	√	√	√	√	√	√	√
Adversarial loss					√	√	√	√
DAE (variance=0.05)		√	√	√		√	√	√
DAE (variance=0.07)			√	√			√	√
DAE (variance=0.1)				√				√
PSNR	11.861	12.105	12.053	12.277	12.085	12.425	12.473	12.566

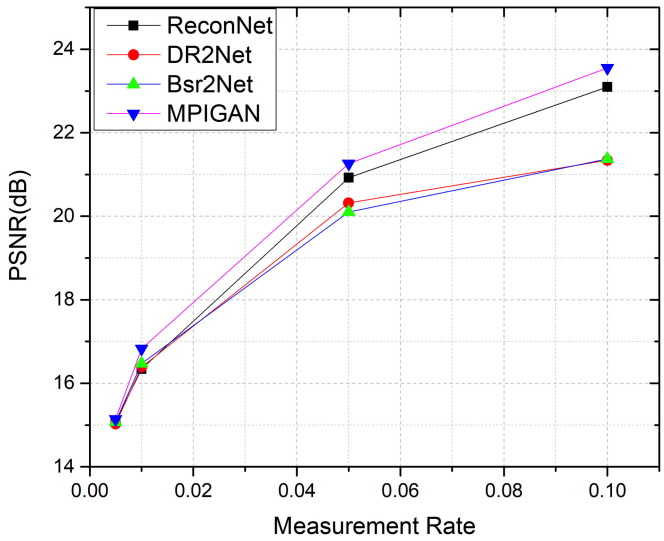

Fig. 3. PSNR values for testing images by different algorithms at different measurement rates.

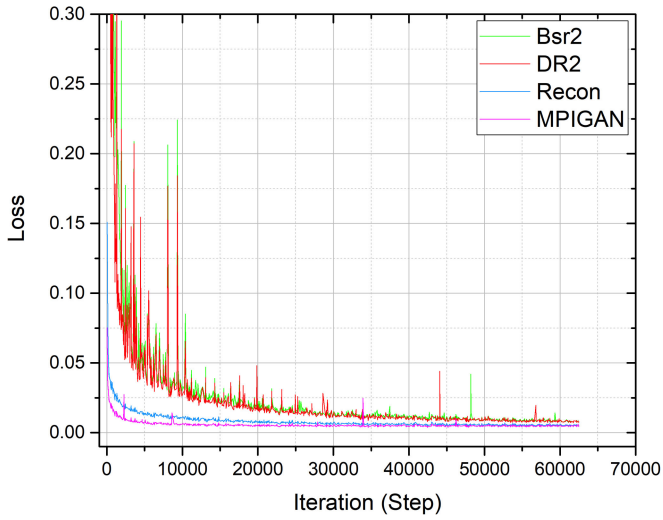

Fig. 4. Loss values for testing images by different algorithms.

Fig. 5. Reconstructed images from different networks (MR = 0.1). (a) Original images, (b) Images reconstructed by ReconNet, (c) Images reconstructed by DR2Net, (d) Images reconstructed by Bsr2Net [35], (e) Images reconstructed by MPIGAN.

MPIGAN network to be 0.1, 0.07, and 0.05, and performed dozens of tunings for each measurement rate to obtain reliable experimental results. The experimental dataset uses celebA with scaled and is cropped to a standard size of 64 × 64. We use 200,010 images for training and the rest for testing. In each round of testing, 64 images are taken to calculate the average value of the indicators. The experimental results are shown as Table IV. From Table IV, we could see that MPIGAN achieves the highest PSNR and SSIM scores in all comparative experiments. For example at MR 0.1, the highest PSNR is 23.548 from MPIGAN, which is obviously higher than 21.34 dB from DR2Net. Fig. 3 shows the PSNR values for testing images by different algorithms at different measurement rates. Fig. 4 shows loss values for testing images by different algorithms. It can be observed from Fig. 4 that MPIGAN obtains the lowest loss value at the same iteration, which demonstrates the point that Autoencoder regularizers can accelerate the gradient descent of GAN. Besides, the loss value of our algorithm is less volatile, which proves the stability of the MPIGAN.

Fig. 5 shows the generated samples of several models, indicating that MPIGAN generates more detailed and textured images than other algorithms. To ensure the fairness of the experiments,

TABLE IV
PSNR AND SSIM OF DIFFERENT ALGORITHMS UPON THE CELEBA DATASET AT DIFFERENT MEASUREMENT RATES

Net	MR=0.1		MR=0.05		MR=0.01		MR=0.005	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Recon	23.097	0.856	20.927	0.772	16.345	0.524	15.080	0.458
DR2	21.340	0.781	20.320	0.738	16.388	0.532	15.026	0.461
Bsr2	21.370	0.783	20.105	0.726	16.476	0.539	15.071	0.473
MPIGAN	**23.548**	**0.866**	**21.261**	**0.787**	**16.829**	**0.571**	**15.142**	**0.474**

TABLE V
PSNR AND SSIM OF DIFFERENT ALGORITHMS UPON THE REVOLVED CELEBA DATASET AT DIFFERENT MEASUREMENT RATES

Net	MR=0.1		MR=0.05		MR=0.01		MR=0.005	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Recon	23.056	0.857	20.879	0.770	16.385	0.526	15.078	0.458
DR2	21.318	0.779	20.302	0.736	16.414*	0.533*	15.065*	0.462*
Bsr2	21.374*	0.784*	20.035	0.724	16.468	0.539	15.106*	0.476*
MPIGAN	23.515	0.866	21.220	0.784	16.800	0.567	15.157*	0.472
MPIGAN+	**23.596**	**0.868**	**21.220**	**0.784**	**16.831**	**0.570**	**15.179**	**0.476**

we add a simulate sampling network to all the networks used for comparison. Relevant work has proved that this approach will instead improve the image reconstruction accuracy of the network [29], so our modification will not affect the performance of the original network.

C. MPIGAN+ Related Exploration and Verification Experiments

In this section, we further explore the improvement of the imaging effect and the robustness of the model by the autoencoder regularizer terms. We replace the three DAEs in MPIGAN with DAE, RAE, and MAE, and train each Autoencoder in advance using differently preprocessed images as described in Section II.E. Most importantly, during the MPIGAN+ alternating training process, we still use images that have not been revolved as input, and we use revolved images for simulation experiments during testing. This means that the network does not actually learn the mappings required for revolved image reconstruction in advance except for the regularization part, yet MPIGAN+ still succeeds in reconstructing the sampled images with high quality in the end. Table V shows the reconstructed sample performance of MPIGAN+ and other networks when dealing with revolved images input. Any number that higher than the same position in Table IV will be marked '$*$', while lower won't be marked. It can be observed that MPIGAN+ performs well at the four measurement rates, while the scores of other networks including MPIGAN is lower than that of Table IV.

As shown in Fig. 6, MPIGAN+ obtains the highest PSNR value. Fig. 7 shows the result of reconstructed images reconstructed by MPIGAN+ and other algorithms, which can be observed that the reconstructed images from MPIGAN+ are clearer than others. The experimental results illustrate that we can improve the robustness of the network by modifying the regular term, so that the network can be adapted to other types of tasks.

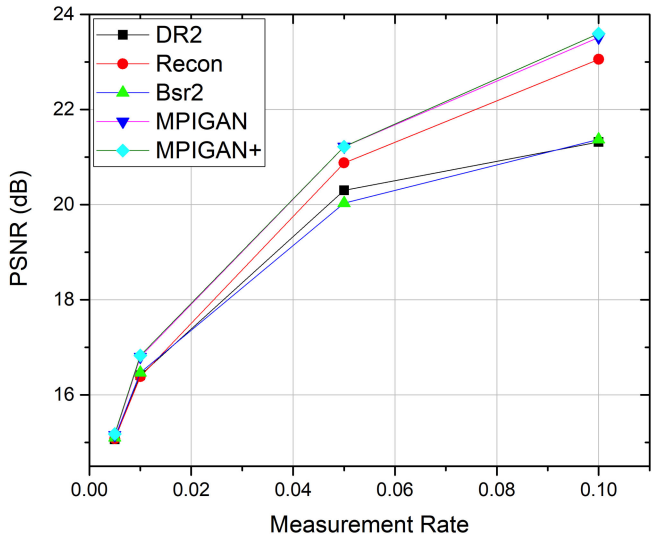

Fig. 6. PSNR values for testing images by different algorithms at different measurement rate.

D. Test on Single-Pixel Imaging System

Fig. 8 is the single-pixel imaging system we used, which mainly consists of a light source, a DMD (0.7XGA 12° DDR), and a photon counter (Hamamatsu H10682). The light source emits very weak light to illuminate the imaging target, and the target is imaged on the DMD through the optical system. The DMD can be viewed as a combination of micro-mirrors with a size of 1024 $\times$ 768, and each micro-mirror can be individually controlled by programming.

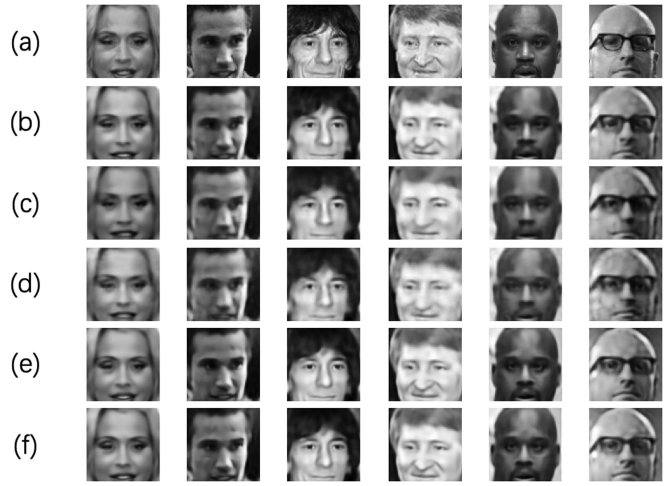

Fig. 7. Revolved images reconstructed by different networks (MR = 0.1). (a) Original revolved images, (b) Images reconstructed by ReconNet, (c) Images reconstructed by DR2Net, (d) Images reconstructed by Bsr2Net, (e) Images reconstructed by MPIGAN, (f) Images reconstructed by MPIGAN+.

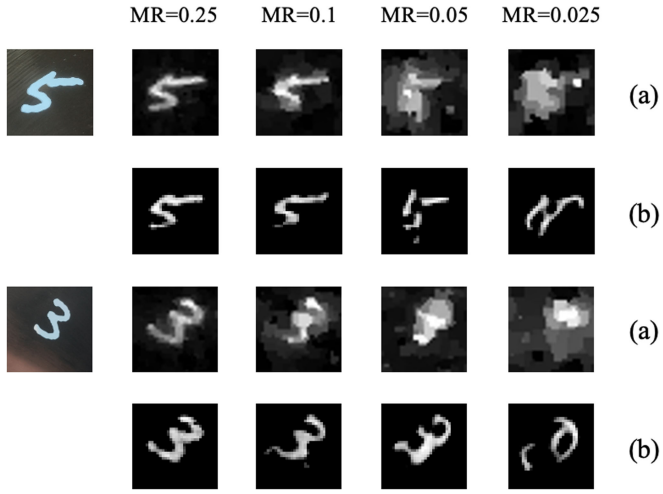

Fig. 9. Reconstruction results of sampled images from a single-pixel imaging system. (a) TVAL3 reconstruction results; (b) MPIGAN reconstruction results.

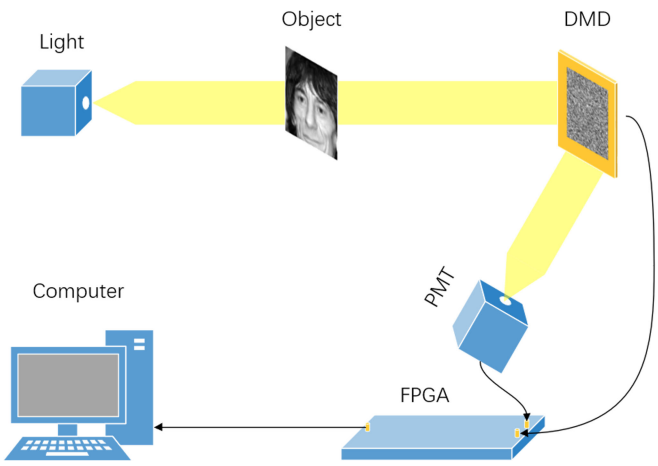

Fig. 8. Image sampling and reconstruction using single-pixel imaging system. The photon counter is set up to capture pulse information instead of million pixel sensors in conventional camera.

Specifically, for each measurement, we program a binarized sampling matrix (i.e., a row of the sampling matrix S) onto the DMD. "1" in the binarization matrix will control the corresponding micromirror in the DMD to rotate +12 degrees, and "−1" will rotate −12 degrees. The rotated micromirror will reflect light from the object in both directions, but only the +12-degree reflected light will enter the photon counter. Under the control of FPGA (Altera DE2-115), the light intensity modulated by DMD is received by the single-point single-photon detector PMT without spatial resolution. We input the single-photon pulse signal from PMT into FPGA for counting, and output the count value to the host computer for image reconstruction.

To load the sampling network into the DMD, we first create a binarized matrix using the sign function:

$$W^b = sign\ (W) = \begin{cases} +1,\ W \geq 0 \\ -1,\ \text{Otherwise} \end{cases} \tag{26}$$

where $W^b \in \mathbb{R}^{M \times N}$ is binary weight of the sampling network. However, the derivative of the sign function is almost 0 everywhere, which makes the backpropagation process unable to proceed smoothly. Inspired by Binarized neural network [36], we still use the sign function to binarize the sampling layer during the forward propagation of the network. Meanwhile, we use the $Htanh$ function for gradient calculation during backpropagation.

$$Htanh\ (x) = Clip\ (x, -1, 1) = \max\left[x - 1,\ \min\left(1, x\right)\right] \tag{27}$$

where $Clip(x, -1, 1) = -1$ if $x < -1$; $Clip(x, -1, 1) = 1$ if $x > 1$; $Clip(x, -1, 1) = x$ if $-1 \leq x \leq 1$. Notably, after binarizing the sampling network, the entire model needs to be retrained to decrease the impact of reduced network weights accuracy. For each sample, we load a row of W^b into the DMD, and finally get a measured value from PMT. After M times of sapling, we finally get a measured value matrix.

We acquired two different sets of image measurements using the experimental setup described above, and loaded the measurements directly into networks to perform reconstruction (Fig. 9).

The contours of the digits can still be reconstructed by MPIGAN at a low measurement rate (MR = 0.05). However, the images reconstructed by TVAL3 are hard to recognize.

At higher measurement rate (MR = 0.25), the images from MPIGAN are clearer and less noisy than images from TVAL3. This means that MPIGAN can filter the noise light existing in the actual imaging environment effectively. To sum up, MPIGAN has excellent performance in real systems, and the anti-disturbance ability is stronger than other networks. Especially, the images reconstructed by our algorithm still retains the main features at extremely low measurement rate. Therefore, MPIGAN can be widely used in astronomy, medicine, biology and other fields. In addition, we also find that the reconstructed images from MPIGAN at a very low measurement rate (MR = 0.025) showed sharp streaks, while the reconstructed images

from TVAL3 were blurry. We speculate that the main reason for this phenomenon is that the handwritten digit images can be attributed to different combinations of simple lines. When the measured value is not enough to map to a complete number, MPIGAN tends to classify existing information into a certain line combination, thereby reconstructing unrecognizable lines.

IV. Conclusion

In this study, we propose the GAN-based compressed reconstruction networks (MPIGAN and MPIGAN+) optimized by multiple Autoencoder priors and auxiliary classifier losses. The experimental results show that the reconstruction effect and accuracy of our network are better than the existing compression imaging network, especially at low measurement rates. Under the condition of the same number of iterations, our algorithm has a faster convergence rate than other networks. After theoretical and experimental demonstration, the pre-trained Autoencoders we added into MPIGAN can extract multi-scale image features and improve the robustness of the network, effectively. In addition, our network can be directly applied to single-pixel imaging system, and the binarization matrix after joint optimization of multiple priors can be directly loaded on DMD for computational imaging. Experiments show that MPIGAN captures the semantic information of samples effectively, restores specific images with higher accuracy and completeness than other networks.

References

[1] B. Kasin, "The widths of certain finite-dimensional sets and classes of smooth functions," *Izvestiya Acad. Nauk SSSR Seriya Matematicheskaya*, vol. 41, no. 2, pp. 334–351, 1977.

[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[3] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.: J. Issued Courant Inst. Math. Sci.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[4] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008.

[5] F. Ferri, D. Magatti, L. Lugiato, and A. Gatti, "Differential ghost imaging," *Phys. Rev. Lett.*, vol. 104, no. 25, 2010, Art. no. 253603.

[6] A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, "Ghost imaging with thermal light: Comparing entanglement and classicalcorrelation," *Phys. Rev. Lett.*, vol. 93, no. 9, 2004, Art. no. 093602.

[7] M. F. Duarte *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[8] J. H. Shapiro, "Computational ghost imaging," *Phys. Rev. A*, vol. 78, no. 6, 2008, Art. no. 061802.

[9] W.-K. Yu *et al.*, "Single photon counting imaging system via compressive sensing," 2012, *arXiv:1202.5866*.

[10] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[11] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[12] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[13] C. Li, *An Efficient Algorithm for Total Variation Regularization With Applications to the Single Pixel Camera and Compressive Sensing*. Houston, Texas, USA: Rice Univ., 2010.

[14] F. Wang, H. Wang, H. Wang, G. Li, and G. Situ, "Learning from simulation: An end-to-end deep-learning approach for computational ghost imaging," *Opt. Exp.*, vol. 27, no. 18, pp. 25560–25572, 2019.

[15] R. Zhu *et al.*, "Ghost imaging based on Y-net: A dynamic coding and decoding approach," *Opt. Exp.*, vol. 28, no. 12, pp. 17556–17569, 2020.

[16] Y. Rivenson, T. Liu, Z. Wei, Y. Zhang, K. de Haan, and A. Ozcan, "PhaseStain: The digital staining of label-free quantitative phase microscopy images using deep learning," *Light: Sci. Appl.*, vol. 8, no. 1, pp. 1–11, 2019.

[17] X. Xie, Y. Wang, G. Shi, C. Wang, J. Du, and X. Han, "Adaptive measurement network for CS image reconstruction," in *Proc. Chin. Conf. Comput. Vis.*, 2017, pp. 407–417.

[18] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 449–458.

[19] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "Dr^2-Net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, pp. 483–493, 2019.

[20] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2672–2680, 2014.

[21] Y. Wu, M. Rosca, and T. Lillicrap, "Deep compressed sensing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6850–6860.

[22] M. Kabkab, P. Samangouei, and R. Chellappa, "Task-aware compressed sensing with generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2297–2304.

[23] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.,* vol. 25, no. 2, pp. 21–30, Mar. 2008.

[24] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[25] C. Qiao *et al.*, "Evaluation and development of deep neural networks for image super-resolution in optical microscopy," *Nature Methods*, vol. 18, no. 2, pp. 194–202, 2021.

[26] S. A. Bigdeli and M. Zwicker, "Image restoration using autoencoding priors," 2017, *arXiv:1703.09964*.

[27] G. Yang *et al.*, "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018.

[28] F. Zhang *et al.*, "REDAEP: Robust and enhanced denoising autoencoding prior for sparse-view CT reconstruction," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 108–119, Jan. 2021.

[29] Y. Guan, Q. Yan, S. Yang, B. Li, Q. Cao, and Z. Fang, "Single photon counting compressive imaging based on a sampling and reconstruction integrated deep network," *Opt. Commun.,* vol. 459, 2020, Art. no. 124923.

[30] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[31] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2172–2180, 2016.

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2226–2234, 2016.

[33] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, *arXiv:1606.01583*.

[34] Z. Zhou *et al.*, "Activation maximization generative adversarial nets," in *Proc. Int. Conf. Learn. Representations*, 2018.

[35] B. Li, Q.-R. Yan, Y.-F. Wang, Y.-B. Yang, and Y.-H. Wang, "A binary sampling Res2net reconstruction network for single-pixel imaging," *Rev. Sci. Instruments,* vol. 91, no. 3, 2020, Art. no. 033709.

[36] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," 2016, *arXiv:1602.02830*.