




Article

Compression Reconstruction Network with Coordinated Self-Attention and Adaptive Gaussian Filtering Module

Zhen Wei ¹, Qiurong Yan ^{1,*}, Xiaoqiang Lu ², Yongjian Zheng ¹, Shida Sun ¹ and Jian Lin ¹¹ School of Information Engineering, Nanchang University, Nanchang 330031, China² Qiyuan Lab, Beijing 100095, China

* Correspondence: yanqiurong@ncu.edu.cn; Tel.: +86-0791-8396-9680

Abstract: Although compressed sensing theory has many advantages in image reconstruction, its reconstruction and sampling time is very long. Fast reconstruction of high-quality images at low measurement rates is the direction of the effort. Compressed sensing based on deep learning provides an effective solution for this. In this study, we propose an attention-based compression reconstruction mechanism (ACRM). The coordinated self-attention module (CSAM) is designed to be embedded in the main network consisting of convolutional blocks and utilizes the global space and channels to focus on key information and ignore irrelevant information. An adaptive Gaussian filter is proposed to solve the loss of multi-frequency components caused by global average pooling in the CSAM, effectively supplementing the network with different frequency information at different measurement rates. Finally, inspired by the basic idea of the attention mechanism, an improved loss function with attention mechanism (AMLoss) is proposed. Extensive experiments show that the ACRM outperforms most compression reconstruction algorithms at low measurement rates.

Keywords: compressed sensing (CS); attention mechanism; adaptive Gaussian filter; loss function; deep learning

MSC: 68T07



Citation: Wei, Z.; Yan, Q.; Lu, X.; Zheng, Y.; Sun, S.; Lin, J.

Compression Reconstruction Network with Coordinated Self-Attention and Adaptive Gaussian Filtering Module. *Mathematics* **2023**, *11*, 847. <https://doi.org/10.3390/math11040847>

Academic Editor: Jakub Nalepa

Received: 30 December 2022

Revised: 27 January 2023

Accepted: 3 February 2023

Published: 7 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In 2004, E.J. Candes, J. Romberg, T. Tao, and D.L. Donoho proposed the compressed sensing theory. The theory shows that if a signal is sparse or compressible in a certain transform domain, it can be recovered accurately from measurements less than those of the Nyquist sampling theorem [1–5]. The compressed sensing theory defines a new paradigm for signal acquisition and reconstruction [6]. Our work focuses on compressed sensing techniques for fast reconstruction of high-quality images at low measurement rates, such as single pixel imaging techniques [7–9] and MRI [10,11].

Despite compressed sensing having many advantages, its sampling and reconstruction time is very long. Traditional compressive reconstruction algorithms such as convex relaxation methods [12], greedy matching pursuit methods [13], and iterative thresholding algorithms are too computationally complex, and the reconstruction time increases exponentially with the resolution of the image [14,15]. In recent years, researchers have begun to use deep learning methods to solve the reconstruction problem of compressed measurements. Deep learning can learn the structural features of the signal and adjust the network weights in a data-driven manner adaptively, enabling the reconstruction of raw images from low-dimensional measurement data without extensive computation [16]. Compared with the traditional iterative algorithm, the reconstruction method based on deep learning not only avoids the huge amount of computation effectively but also obtains better reconstruction quality. This provides a solution for image reconstruction based on compressed sensing. Therefore, compressed sensing technology based on deep learning has also attracted much attention and become a research hotspot.

Most of the existing deep learning networks for compressed sensing rely on massive data streams and deeper convolutional layers to obtain higher receptive fields and high-quality reconstructions. Many data calculations cause data redundancy, which reduces the reconstruction rate and efficiency greatly [17]. The reconstruction quality of the important regions of the image may be worse, thereby lowering the reconstruction quality.

In this study, an attention-based compression reconstruction mechanism (ACRM) is proposed, which enables the network to learn important information of images and suppress useless information autonomously. The reconstruction quality of the network is improved greatly. Extensive experiments show that the ACRM outperforms most compression reconstruction algorithms at a low measurement rate. Our proposed attention mechanism and adaptive Gaussian filtering also outperforms other mainstream attention modules in this field. Our contributions are as follows:

- We propose a coordinated self-attention module (CSAM), which not only introduces information into the channel and the spatial domain but also captures the global information of the image, improving the network's ability to capture long-range relationships for better imaging results.
- We propose an adaptive Gaussian filter sub-network in the frequency domain to make up for the defect of global average pooling in the CSAM. It can capture information on different frequency components of the image selectively when the measurement rate is changed.
- We propose a loss function with attention based on the traditional MSE-Loss (AMLoss) to match the gradient descent algorithm with the attention mechanism and focus more on the important parts of the image during optimization. Extensive experiments prove that the AMLoss can significantly improve the reconstruction quality.

2. Background and Related Work

This section mainly introduces the development of the compressed sensing imaging and the development of the attention mechanism.

2.1. Deep Learning Based on Compressed Sensing Reconstruction

The process of image reconstruction is the process of finding solutions to underdetermined equations. There are infinite solutions to this problem in theory. However, the reconstruction of image x can be carried out by solving the L_0 norm problem based on compressed sensing theory. Let x equal $\Psi\alpha$, where Ψ is an orthonormal basis and the number of non-zero elements of α is much less than N (N refers to the dimension of image x). The image reconstruction problem can be expressed as

$$y = \min \|\alpha\|_0 \text{ s.t. } \Phi\Psi\alpha. \quad (1)$$

where Φ is the observation matrix, which projects the high-dimensional signal x into the low-dimensional space.

Traditional compressed sensing reconstruction methods are based on sparse prior knowledge to iteratively reconstruct the original signal by solving an optimization problem. They are mainly convex relaxation methods [12], greedy matching pursuit methods [13], and Bayesian methods [18], aiming at the problem that the sparse assumption model in traditional compressive sensing theory is not fully satisfied in practical applications. The deep learning method uses a data-driven approach to learn signal structure features, relaxes the assumptions on the sparseness of the original signal, and learns the specific structure of the actual signal by adaptively adjusting the network weights. At the same time, the measurement and reconstruction process can be turned into an end-to-end framework.

In 2015, Mousavi et al. used stacked denoising autoencoders as unsupervised feature learners to achieve fast reconstruction of compressed sensing images [19]. In 2016, Kulka-rni et al. proposed the ReconNet model based on image super-resolution reconstruction, which improved the accuracy of reconstructed images [16]. In 2017, Mousavi et al. pro-

posed the DeepInverse deep learning model [20]. In the same year, Yao et al. used the ResNet structure to build a deep residual reconstruction network named DR²-Net based on the ReconNet, which once again improved the accuracy of the reconstructed images [17]. In 2018, S. Lohit et al. proposed a variant of the ReconNet that used an adversarial loss to further improve the reconstruction quality [21]. In 2020, Yang et al. proposed the ADMM-SCNet, which used traditional model-based compressed sensing methods and data to drive deep learning methods for reconstructing images from sparsely sampled measurements. The method achieves good reconstruction accuracy at fast computation speed [22]. In the same year, inspired by generative networks and attention mechanisms, Yuan et al. proposed a down-sampled MRI reconstruction method based on SARA-GAN. The method applies the relative average discriminator theory to make full use of prior knowledge. At the same time, adding the self-attention mechanism in the upper layers of the generator can overcome the problem of limited convolution kernel size [23]. In 2021, Zhang et al. proposed a deep learning system for attention-guided dual-layer image compression (AGDL), which advanced the state of the art in perceptual image compression [24]. In the same year, Barranca formulated a new framework for learning improved sampling paradigms for compressed sensing in a bio motivated manner, significantly improving the quality of signal reconstruction across multiple connection weight penalty schemes and signal classes [25].

2.2. Attention

In recent years, attention mechanisms have been seen in various types of tasks, such as image processing, machine translation, and natural language processing. The attention mechanism, as its name implies, draws on the unique brain processing signal mechanism of human vision [26]. In computer vision, the attention mechanism can filter out important regions of the input image and extract important information from key parts [27]. It is a means of sifting out high-value information from a large volume of information, which is widely used in the field of remote sensing [28,29].

Convolutional neural networks (CNNs) have been shown to be effective models for a wide range of vision tasks. They can generate image representations that capture hierarchical patterns and obtain global theoretical receptive fields. If researchers want to find the correlation between two parts of an image that are far apart, then they generally continue to increase the number of convolutional layers and the number of pooling layers, which will increase the cost of computation and the number of parameters. Then, Wang et al. proposed the non-local network in the spatial domain based on the self-attention mechanism. Each pixel on the image has an attention map, which increases its ability to obtain global information [30]. Global information refers to the information containing the dependencies between image elements. However, this model requires a large amount of computation. Cao et al. found in the experiment that the attention map of each pixel in an image is the same, so a simple module GC-Net that shares the attention map was constructed. This module reduces the amount of computation greatly [31]. Many of the previous works have proposed structures in the spatial domain that can improve network performance, and the SE-Net is the first attention mechanism module proposed at the channel domain. It can complete the weight labeling of features in the channel dimension dynamically and adaptively, paying attention to the dependencies at the channel level of the model. Most importantly, the SE-Net is very simple in construction and easy to deploy without introducing new functions or layers [32]. However, the SE-Net module needs to reduce the amount of calculation by reducing the dimensionality of the fully connected layer, which will weaken the learning of weights. Therefore, Wang et al. proposed the ECA-Net based on the SE-Net, which uses a local cross-channel interaction strategy without dimensionality reduction. It can improve cross-channel interactive learning while avoiding dimensionality reduction, making the module lighter [33]. Today, global average pooling in the SE module has become a common spatial information-encoding method. However, it saves the global spatial information in a channel descriptor, which makes it difficult to

save the position information. Hou et al. then proposed to decompose the global average pooling into one-dimensional average pooling in parallel along the horizontal and vertical directions. Perceptual feature maps along two independent directions are obtained, which in turn encode the feature maps into two attention maps. Then the position information is stored in the generated attention map [34]. This method can capture not only inter-channel information, but also orientation-aware and position-sensitive information.

Inspired by the coordinated attention module and the global context block that can improve long-distance capture capability [34,35], we designed the coordinated self-attention module (CSAM). While capturing inter-channel information and spatial location information, this module also enhances feature transformation and feature aggregation capabilities, which improves the reconstruction capabilities of the network ultimately.

3. The Proposed Method

The main innovations of our work are to introduce a new attention mechanism into the field of compressed sensing imaging and propose a new interpretable loss function. In this section, we will describe the proposed method in detail, including model formula and the proof of loss function principle.

3.1. Overall Network Framework

This part mainly introduces the structure of ACRM as shown in Figure 1. The network consists of two fully connected layers, four ACR submodules, and an adaptive Gaussian filter. The first fully connected layer is a sampling network used to down sample the original image to generate CS measurements [36]. The second fully connected layer is used for the preliminary reconstruction of the CS measurements [36]. Related studies have shown that using fully connected layers as linear mapping networks can reconstruct high-quality primary images [17] so that the subsequent deep convolution can proceed smoothly. The ACR submodule includes three convolutional layers and the CSAM. The convolution kernel sizes and channels of the three convolutional layers from left to right are $11 \times 11 \times 64$, $7 \times 7 \times 32$, and $3 \times 3 \times 1$, respectively. After each convolutional layer, the LeakyReLU function is used to improve the ability of convolutional nonlinear feature extraction. Each ACR submodule uses a residual structure to cope with vanishing and exploding gradients. Next, we focus on the CSAM, the adaptive Gaussian filtering sub-network and the loss function used for network training.

3.2. Coordinated Self-Attention Module

Ma et al. studied the classical attention module and established the framework of the attention module in a broad sense, thinking that the attention block is composed of three parts: context extraction, transformation, and fusion [37]. The structure of our designed attention block also conforms to this framework, as shown in Figure 2.

Context extraction is used to collect relevant feature information from the feature map of the internal relationship of the image. We assume that the feature map obtained by the previous convolution block is $x \in \mathbb{R}^{C \times H \times W}$. We perform one-dimensional average pooling in two directions, so the outputs of the c -th channel at height h and width w are expressed as [34]

$$z_c^h = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \quad (2)$$

$$z_c^w = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w) \quad (3)$$

where C is the number of channels of the feature map, and H and W are the height and width of the feature map, respectively.

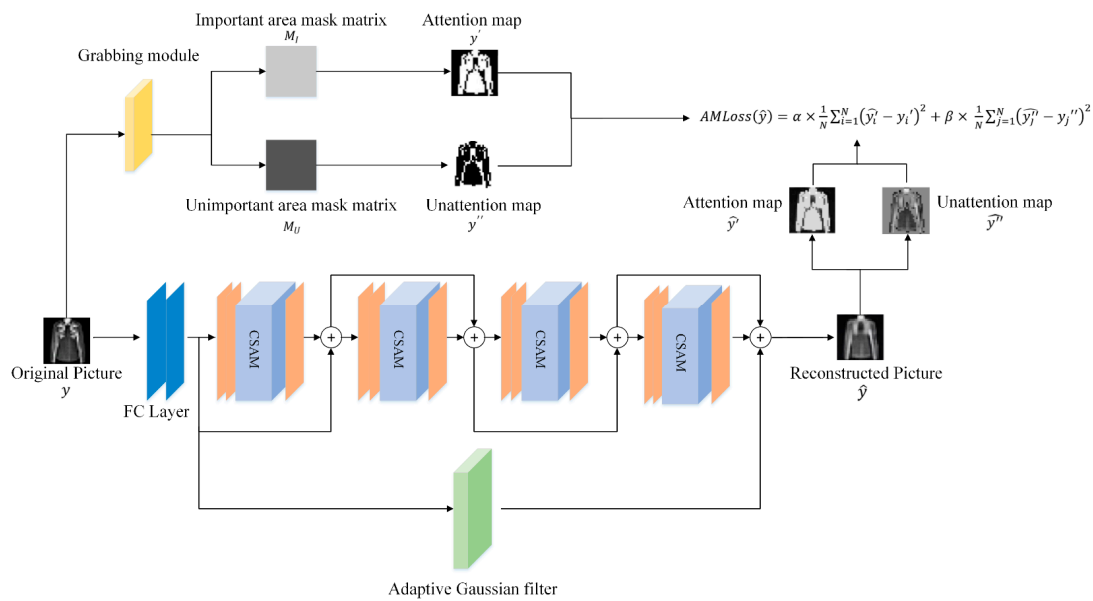


Figure 1. ACRM structure diagram. It contains 2 fully connected layers, 4 CSAMs, 12 convolutional layers, and an adaptive Gaussian filter. Each CSAM module and 3 convolutions make up the ACR sub-module. The kernel sizes and channels of the three convolutional layers from left to right are $11 \times 11 \times 64$, $7 \times 7 \times 32$, and $3 \times 3 \times 1$, respectively. In addition to sequential connections between each module, there are skip connections. The grab module generates the corresponding important area mask matrix M_I and unimportant area mask matrix M_U . Then, we multiply the \hat{y} and y with the M_I and M_U to obtain y' , y'' , \hat{y}' , and \hat{y}'' . Finally, the AMLoss for backpropagation optimization is obtained.

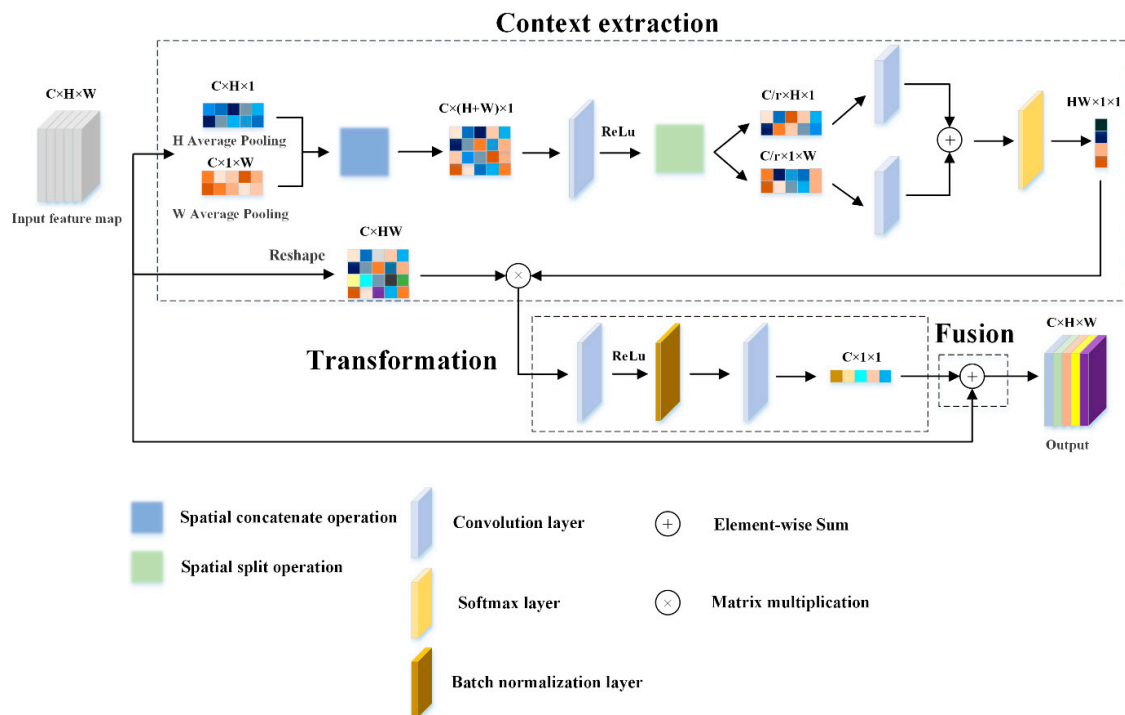


Figure 2. CSAM structure diagram. The module contains three parts: context extraction, transformation, and fusion. Among them, “H Average Pooling” and “W Average Pooling” refer to 1D horizontal average pooling and 1D vertical average pooling, respectively. “ReLU” is the nonlinear activation function $ReLU$; “r” is a reduction ratio to reduce the amount of computation.

The above results in two independent orientation-aware feature maps, which not only capture the long-distance dependencies in their respective orientations well but also preserve spatial location information from each other.

Then z^h and z^w are used with convolution *Conv1* to generate the aggregated feature map [34]:

$$f = \delta\left(\text{Conv1}\left(\left[z^h, z^w\right]\right)\right) \tag{4}$$

where

Conv1 is a convolutional layer with a convolution kernel size of 1×1 ,

$[\cdot, \cdot]$ represents the connection operation along one dimension,

$z^h \in \mathbb{R}^{C \times H \times 1}$ and $z^w \in \mathbb{R}^{C \times 1 \times W}$ are the input feature maps,

δ is the nonlinear activation function *ReLU*,

$f \in \mathbb{R}^{C/r \times (H+W) \times 1}$ is the output feature map,

r is a reduction ratio to reduce the amount of computation.

Then we cut it into separate tensors along the two spatial dimensions f^h and f^w . Then we use two convolutions *Conv2* and *Conv3* to restore the tensors as dimension $C \times H \times W$ consistent with the input dimension. Finally, the two tensors are connected along the two spatial dimensions to form an attention map with long-distance dependencies initially:

$$G = \text{Conv2}\left(f^h\right) \oplus \text{Conv3}\left(f^w\right) \tag{5}$$

where *Conv2* and *Conv3* are convolutional layers with a convolution kernel size of 1×1 , respectively, \oplus indicates that the two matrices are added along different dimensions, $f^h \in \mathbb{R}^{C/r \times H \times 1}$ and $f^w \in \mathbb{R}^{C/r \times 1 \times W}$ are the cut feature vectors, and $G \in \mathbb{R}^{C \times H \times W}$ is the generated attention map.

Unlike the coordinated attention module, the interaction between the attention map and the original image are used to calculate the position in the graph to capture long-range correlations after initially generating an attention map. Therefore, based on having a certain amount of attention, we further improve the context modeling ability, and aggregate the features of all positions to obtain global context features.

Inspired by [31], all pixels in the image share an attention map. The relationship between positions i and j can be expressed as

$$e_i = \sum_{j=1}^N \frac{e^{Gx_j}}{\sum_{p=1}^N e^{Gx_p}} x_j \tag{6}$$

where N (N equals H times W) represents the total number of pixels in the image.

Transformation aims to capture the channel and space dependencies and transform the extracted features on the nonlinear attention space to obtain the attention map z_f . The output z_f can be expressed as [31]

$$z_f = \text{Conv5}\left(\delta\left(\text{BN}\left(\text{Conv4}\left(e_i\right)\right)\right)\right) \tag{7}$$

where *Conv4* and *Conv5* are convolutional layers with a convolution kernel size of 1×1 , respectively, and δ represents the nonlinear activation function *ReLU*. *BN* represents batch normalization processing.

Fusion aims to combine the obtained attention map with the feature map of the original convolutional block. According to (2) to (7), the process of aggregating global context features into features of each location can be expressed as

$$z_i = x_i + \text{Conv5}\left(\delta\left(\text{BN}\left(\text{Conv4}\left(e_i\right)\right)\right)\right)$$

$$\begin{aligned}
 e_i &= \sum_{j=1}^N \frac{e^{(\delta(\text{Conv1}([z^h, z^w]))^h \oplus \delta(\text{Conv1}([z^h, z^w]))^w) x_j}}{\sum_{p=1}^N e^{(\delta(\text{Conv1}([z^h, z^w]))^h \oplus \delta(\text{Conv1}([z^h, z^w]))^w) x_p}} x_j \\
 z_c^h &= \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \\
 z_c^w &= \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w)
 \end{aligned} \tag{8}$$

The attention module we designed not only acquires channel-level global features in the channel domain, but it also has a stronger ability to capture global information in the spatial domain. As mentioned above, we use the attention map in two directions to reflect whether there is a place to pay attention between the corresponding rows and columns, and then we encode the established attention map with itself. Then the global information is exploited to locate the areas that need attention accurately in the image. This enhances the power of context extraction and transformation nicely compared to general self-attention networks. Moreover, we also analyze the insufficiency of the global average pooling method in CSAM and propose an adaptive Gaussian filter to optimize this attention module (this is described in detail in the next section). Finally, the overall reconstruction ability of the network is improved. We showed this in experiments.

3.3. Adaptive Gaussian Filter Sub-Networks

Although the method of decomposing the 2D global average pooling into two parallel 1D feature encodings improves the ability of the network to utilize spatial information effectively in CSAM, Qin et al. pointed out that this method cannot capture rich input representation [38]. It was demonstrated that the global average pooling method is the lowest frequency component of the discrete cosine transform. In other words, in the frequency domain, only a single component is used, and other useful components are ignored. Usually, the low-frequency part of the image mainly contains general image information, while the high-frequency part mainly contains the detailed information of the image. These details are very important for the reconstruction of the image. So, an adaptive Gaussian filtering sub-network is proposed to solve this problem.

The specific implementation of two-dimensional Gaussian filtering assigns different Gaussian weight values to the surrounding pixel values within a certain range. It obtains the result of the current point after the weighted average. The two-dimensional Gaussian function is

$$G(x, y) = A e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} \tag{9}$$

where A is the amplitude, x_0 and y_0 are the center point coordinates, and σ_x and σ_y are the standard variances.

The amplitude of the two-dimensional Gaussian function is inversely proportional to the standard variances σ . The larger the σ , the wider the frequency band of the Gaussian filter and the better the smoothness. However, the two-dimensional Gaussian filter is mainly used as a controllable filter, so the amplitude is fixed to one.

Usually, the low-frequency part of the image mainly contains general image information, while the high-frequency part mainly contains detailed information of the image. When the measurement rate is lower, the compressed sampled image contains more low-frequency information, so the bandwidth of the filter can be set smaller when the measurement rate is higher. Therefore, the standard variances σ of the Gaussian filter sub-network we designed can vary with the measurement rate of the sampling network. In this way, the Gaussian filtering sub-network can supplement more frequency domain component information to the main network under different measurement rate conditions, which increases the expressive ability in the frequency domain. This method makes up

for the deficiency of the global average pooling method and adopts the skip connection method as shown in Figure 1 to reduce the computational complexity.

3.4. *AMLoss (Attention MSE Loss)*

The MSE (mean square error) loss function has the advantages easily using the gradient descent algorithm and is conducive to the convergence of the function. However, the MSE loss function has the characteristics of giving a larger penalty to the larger error and a smaller penalty to the smaller error. If there are outliers in the sample, the MSE loss function will give higher weights to the outliers, thereby ignoring the influence of the image content itself, which will reduce the overall performance of the model ultimately. Although there are some works on the combination of attention and loss function [39–41], unfortunately, the combination of attention and MSE loss function is still very rare. Therefore, we propose an improved loss function based on the traditional MSE loss, which is called *AMLoss* (attention MSE loss). Its expression is

$$AMLoss(\hat{y}) = \alpha \times \frac{1}{N} \sum_{i=1}^N (\hat{y}'_i - y'_i)^2 + \beta \times \frac{1}{N} \sum_{j=1}^N (\hat{y}''_j - y''_j)^2 \quad (10)$$

where N is the number of pixels in the image, α and β are the set hyperparameters, respectively, and \hat{y}' and \hat{y}'' are the pixel values of the important area and the non-important area of the predicted value output \hat{y} by the network, respectively, y' and y'' are the pixel values of the important area and the non-important area of the real value input y by the network, respectively.

In the training process, as shown in Figure 1, the designed adaptive grabbing module is used to extract its important area part before the input image enters the network. After that, it generates the corresponding important area mask matrix M_I and unimportant area mask matrix M_U . Finally, we multiply the predicted value \hat{y} and the real value y with the masks M_I and M_U to obtain the respective important and non-important regions. Take MNIST dataset as an example. The characteristic of this dataset is that the main part of the image is in the center of the image, and the value of pixels is very large. According to this feature, we design the capture module as follows: First, we normalize the image. Then we calculate the average pixel value of all pixels in the picture and compare the value of all pixels in the picture with the average pixel value. If the value of the pixel point is greater than the average pixel value, the position of the matrix corresponding to the point is assigned as 1. On the contrary, the position of the matrix corresponding to the point is assigned as 0. Finally, the important area matrix is generated. We believe that the important part of the image is determined according to the work task. If the important part of the picture is artificially defined, the method of extraction is not unitary. Different datasets also have different methods.

In this way, the MSE-loss function will increase the penalty for important areas of the image and reduce the penalty for non-important areas of the image. This will not only alleviate the defects of the above MSE-loss function but also make the loss function have “attention”, paying more attention to the important parts of the image. The error of the important part has a greater impact on the update of parameters such as weights and biases of the network after the back-propagation algorithm. The proof process is as follows:

The last activation function used by the last convolution of this network is *LeakyReLU*:

$$LeakyReLU(z) = \begin{cases} z & z > 0 \\ \gamma z & z \leq 0 \end{cases} \quad (11)$$

This function solves *Relu*'s neuron death problem with a small positive slope γ in negative regions, so it can backpropagate even for negative input values. We assume an input vector y , which is transformed by the *LeakyReLU* function to obtain a vector r , and

propagates forward to obtain an error value e . We only solve the gradient of e to y and do not solve the update of specific parameters in the network after e backpropagation.

The known condition is:

$$y = (y_1, y_2, y_3, y_4, \dots, y_n) \tag{12}$$

$$r = \text{LeakyReLU}(y) \tag{13}$$

$$e = \text{AMLoss}(r) \tag{14}$$

According to (11) and (13), we obtain:

$$\frac{dr_i}{dy_i} = \begin{cases} 1 & y_i > 0 \\ \gamma & y_i \leq 0 \end{cases} \tag{15}$$

According to (13), the gradient of r to y is:

$$\frac{dr}{dy} = \left(\frac{dr_1}{dy_1}, \frac{dr_2}{dy_2}, \frac{dr_3}{dy_3}, \dots, \frac{dr_n}{dy_n} \right) \tag{16}$$

According to (10) and (14), the gradient of e to r is:

$$\begin{aligned} \frac{\partial e}{\partial r} &= \left(\frac{\partial e}{\partial r_1}, \frac{\partial e}{\partial r_2}, \frac{\partial e}{\partial r_3}, \dots, \frac{\partial e}{\partial r_n} \right) \\ &= \frac{\partial \text{AMLoss}(r)}{\partial r} \\ &= (\alpha(r'_1 - y'_1) + \beta(r''_1 - y''_1), \frac{\alpha}{2}(r'_2 - y'_2) + \frac{\beta}{2}(r''_2 - y''_2), \frac{\alpha}{3}(r'_3 - y'_3) \\ &\quad + \frac{\beta}{3}(r''_3 - y''_3), \dots, \frac{\alpha}{n}(r'_n - y'_n) + \frac{\beta}{n}(r''_n - y''_n)) \end{aligned} \tag{17}$$

According to the chain rule, we obtain:

$$\frac{\partial e}{\partial y_i} = \frac{\partial e}{\partial r_i} \frac{dr_i}{dy_i} \tag{18}$$

According to (16) and (18), the gradient of e to y is:

$$\begin{aligned} \frac{\partial e}{\partial y} &= \left(\frac{\partial e}{\partial y_1}, \frac{\partial e}{\partial y_2}, \frac{\partial e}{\partial y_3}, \dots, \frac{\partial e}{\partial y_n} \right) \\ &= \frac{\partial e}{\partial r} \odot \frac{dr}{dy} \end{aligned} \tag{19}$$

where \odot represents the multiplication of parity elements.

It can be seen from (19) that when $\alpha > 1 \geq \beta$, the weight of errors in important areas will be increased, and the weight of errors in non-important areas will remain the same or decrease. It is worth mentioning that the AMLoss is suitable for networks with an attention mechanism, because networks with an attention mechanism filter out important parts of the image. The AMLoss is more inclined to reduce the loss value for important parts of the image when it is minimized by backpropagation.

4. Experiments and Results

In this section, we design three experiments to test the reconstruction performance of the ACRM. They are a comparison of different attention mechanism modules, an optimization comparison of the AMLoss in different attention mechanism networks, and a comparison of different compressed sensing networks. Our experiments were implemented in the TensorFlow 2.0 framework and run on CUDA11 for accelerated processing. The computer we used was equipped with an Intel Xeon W-2133 CPU, an Nvidia GEFORCE RTX 2080Ti 11GB graphics card (in particular, we used Nvidia RTX A6000 48G for CelebA

dataset), 64 GB of RAM, and a 2 T hard drive. The configuration of specific network parameters is explained in the following subsections.

4.1. Comparison of Different Attention Mechanism Modules

The feature-rich MNIST handwriting dataset and Fashion-MNIST dataset were used as training data. Taking the Fashion-MNIST dataset as an example, 40,000 images and 10,000 images were selected as training and testing sets, respectively, according to the hold-out method. The batch size of each entry into the network was 64. Each training was 500 batches. One round of training was 32,000 pictures. Total training was 100 rounds. The optimizer chose ADAM, and the learning rate was set to 10^{-4} . We tested with images from the test set and calculated the average of the metrics.

In this experiment, the main purpose was to verify the improvement effect of the attention mechanism module on the reconstruction ability of the general compressed sensing network. We compared the CSAM with the classical attention mechanism module to verify the superiority of that in the compression reconstruction network. To highlight the superiority of the attention module, we simplified the convolutional network in the ACRM and built a basic convolutional network (BC-Net) consisting of two fully connected layers and eight convolutional connections as the main network. The eight convolution kernels were divided into four BC modules, and each group of BC modules was composed of convolution kernel size and channel number of $11 \times 11 \times 64$ and $7 \times 7 \times 1$, respectively. The *LeakyReLU* function was used after each convolutional layer. The SE-Net [32], the CBAM [42], the coordinated attention module [34], and the GC-Net [31] were embedded in it, respectively, as shown in Figure 3. The modules surrounded by dotted lines in Figure 3 were used to place different attention mechanism modules in the experiments. At the same time, we also set up a group of ablation experiments to verify the effectiveness of the sub network. The dataset adopted fashion MNIST, and the network and dataset settings were consistent with the above. Finally, all networks used the general MSE function as the loss function. The signal-to-noise ratio (PSNR) and structural similarity (SSIM) were used as evaluation metrics for reconstruction performance. The calculation formulas of PSNR and SSIM are as follows:

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right) \quad (20)$$

$$MSE = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \|X(i, j) - Y(i, j)\|^2 \quad (21)$$

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \quad (22)$$

where H and W are the height and width of the image, respectively, and MSE represents the mean square error of the current image X and the reference image Y . μ_X and μ_Y are the mean of X and Y , respectively, σ_X and σ_Y are the variances of X and Y , respectively, σ_{XY} is the covariance of X and Y , and c_1 and c_2 are constants.

The experimental results are shown in Tables 1 and 2. The measurement rate (MR) is the ratio M/N of the sampling points M of the image to the total image pixels N . The CSAM outperformed in all comparisons from the table. The PSNR at all measurement rates was better than the other networks. It can be seen from Table 3 that under different measurement rates, the reconstruction effect was improved by using the adaptive Gaussian filter sub network, which proved the effectiveness of the sub network. Figure 4 shows the differences in the PSNR values of several models under different datasets. Other attention modules cannot maintain consistent performance in different datasets, which indicates that the CSAM is more robust than other attention modules.

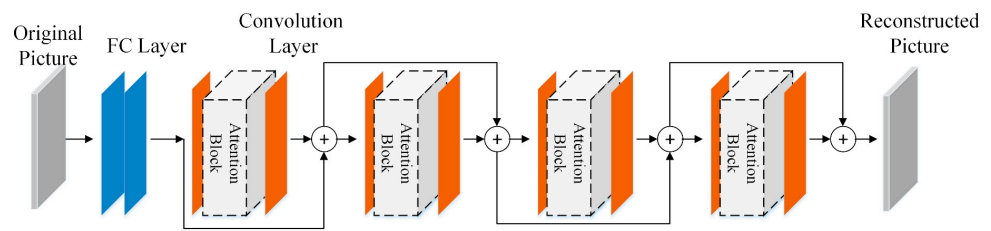


Figure 3. The structure diagram of BC-Net. It contains 2 fully connected layers and 8 convolutional layers. The 8 convolutional layers form 4 BC modules. In addition to the sequential connections between each module, there are skip connections. After the first convolutional layer of each module, the attention mechanism modules can be connected sequentially to form an attention-compressed sensing network.

Table 1. PSNR and SSIM of algorithms embedded with different attention mechanisms on the MNIST dataset at different measurement rates.

Methods	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01		MR = 0.005	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BC	19.344	0.925	16.619	0.859	14.964	0.798	10.563	0.578	8.288	0.364
BC + SE	19.591	0.932	16.894	0.882	15.195	0.821	10.332	0.561	8.412	0.381
BC + CBAM	18.918	0.879	16.819	0.86	15.658	0.842	10.599	0.557	7.737	0.294
BC + GC	21.091	0.953	18.043	0.912	15.652	0.844	10.679	0.584	8.499	0.382
BC + CA	18.15	0.89	16.362	0.845	14.788	0.78	10.11	0.524	8.446	0.39
BC + CSAM	21.448	0.953	18.199	0.92	15.836	0.856	10.789	0.595	8.592	0.416

Note: Bold numbers indicate the best value for that measurement.

Table 2. PSNR and SSIM of algorithms embedded with different attention mechanisms on the Fashion-MNIST dataset at different measurement rates.

Methods	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01		MR = 0.005	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BC	17.284	0.769	15.88	0.718	15.031	0.678	12.686	0.563	10.729	0.428
BC + SE	17.412	0.782	15.857	0.723	14.947	0.685	12.584	0.551	10.493	0.409
BC + CBAM	17.493	0.792	15.947	0.731	15.093	0.687	12.34	0.543	10.79	0.453
BC + GC	17.481	0.787	16.053	0.729	15.137	0.689	12.653	0.568	10.665	0.42
BC + CA	17.303	0.784	15.924	0.726	14.695	0.659	12.657	0.556	10.805	0.441
BC + CSAM	17.636	0.791	16.151	0.739	15.183	0.689	12.885	0.656	10.821	0.45

Note: Bold numbers indicate the best value for that measurement.

Table 3. Ablation study on the adaptive Gaussian filter (AGF).

Methods	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01		MR = 0.005	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BC + CSAM	17.636	0.791	16.151	0.739	15.183	0.689	12.885	0.656	10.821	0.45
BC + CSAM + AGF	17.964	0.802	16.334	0.752	15.231	0.702	12.975	0.675	10.894	0.47
	$\sigma = 6$		$\sigma = 6$		$\sigma = 5$		$\sigma = 2.75$		$\sigma = 1$	

Note: Bold numbers indicate the best value for that measurement. σ is the standard variance of the Gaussian filter at different measurement rates.

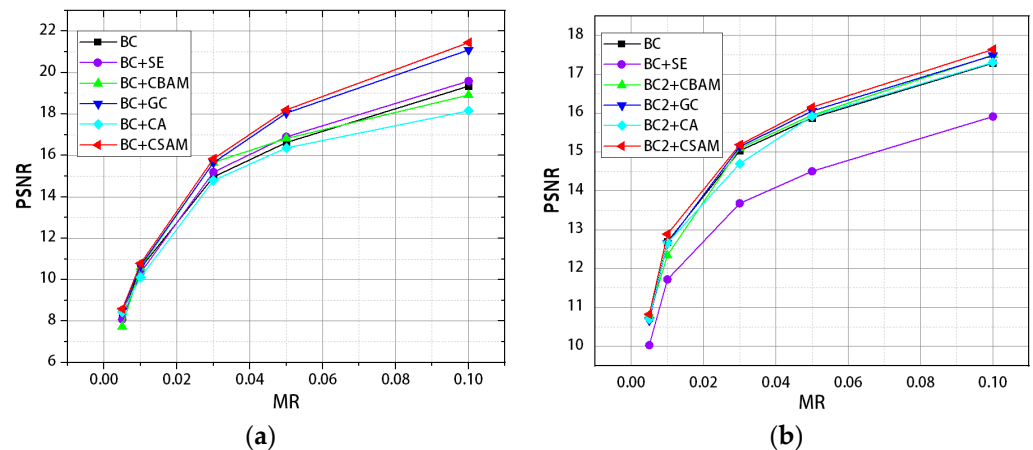


Figure 4. (a) Different attention mechanism algorithms test the PSNR of MNIST pictures at different measurement rates. (b) Different attention mechanism algorithms test the PSNR of Fashion-MNIST pictures at different measurement rates.

4.2. Optimization Comparison of the AMLoss in Different Attention Mechanism Networks

In this subsection, we designed five sets of controlled experiments to verify that the AMLoss can optimize the compressed sensing network with an attention mechanism. The backbone network of this experiment used the BC-Net proposed in the previous section. The configuration of training parameters was the same as in the previous section. In this experiment, only the MNIST dataset was used for experiments. A set of controlled experiments was set up for each network, which were trained using the MSE-Loss and the AMLoss as loss functions, respectively. We tested with images from the test set and calculated the average of the metrics.

The experimental results are shown in Table 4. The AMLoss could improve the attention compressed sensing network significantly when we chose a suitable α . This boost was most pronounced at higher measurement rates. Figure 5 also shows that the reconstructed image using the AMLoss network contained more detail and less noise. The above experiments showed that the loss function with attention could help the network to pay more attention to the important part during the training process and add more details to the part. Finally, the overall quality of the reconstructed image was improved.

Table 4. Performance comparison of the AMLoss and MSE-Loss acting on different attention compressed sensing networks in the MNIST dataset.

Methods	Loss	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01		MR = 0.005	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BC + SE	MSE-Loss	19.591	0.932	16.894	0.882	15.195	0.821	10.332	0.561	8.412	0.381
	AMLoss (2)	20.814	0.943	17.601	0.894	15.223	0.821	10.503	0.564	8.529	0.382
BC + CBAM	MSE-Loss	18.918	0.879	16.819	0.86	15.658	0.842	10.599	0.557	7.737	0.294
	AMLoss (1.1)	21.042	0.953	17.512	0.894	15.824	0.845	10.721	0.559	8.32	0.404
BC + GC	MSE-Loss	21.091	0.953	18.043	0.912	15.652	0.844	10.679	0.584	8.499	0.382
	AMLoss (1.2)	21.210	0.955	18.244	0.92	15.739	0.845	10.758	0.588	8.553	0.386
BC + CA	MSE-Loss	18.15	0.89	16.362	0.845	14.788	0.78	10.11	0.524	8.446	0.39
	AMLoss (1.2)	20.395	0.944	17.6	0.909	15.042	0.822	10.24	0.571	8.521	0.41
BC + CSAM	MSE-Loss	21.448	0.953	18.199	0.92	15.836	0.856	10.789	0.595	8.592	0.416
	AMLoss (1.15)	22.182	0.959	18.485	0.931	16.12	0.86	10.847	0.605	8.646	0.418

Note: The bold numbers represent the best value for a network at that measurement rate. The value in parentheses next to “AMLoss” is the value of α (β is 1 by default).

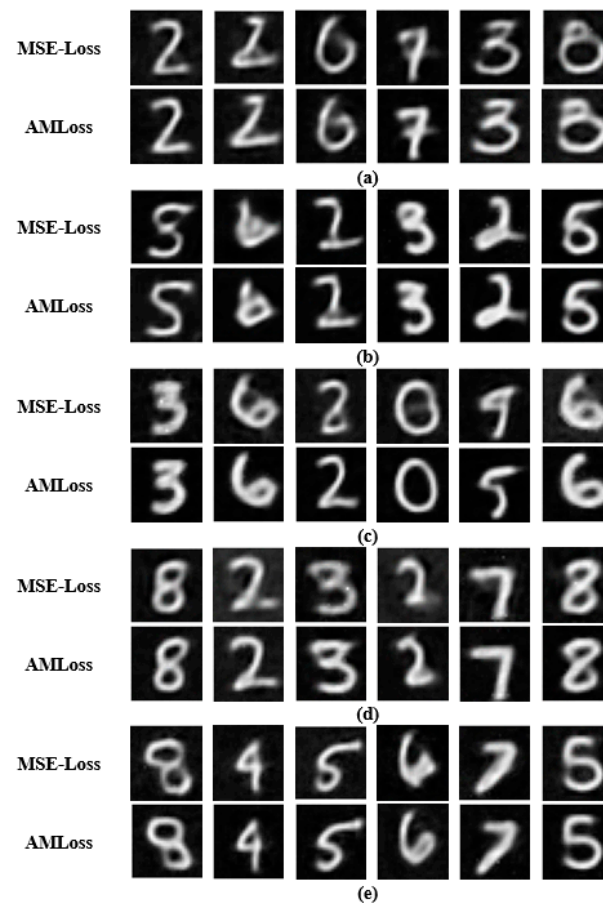


Figure 5. Model images were reconstructed on the MNIST dataset, $MR = 0.01$. The top-down compressed sensing networks are (a) BC + SE, (b) BC + CBAM, (c) BC + GC, (d) BC + CA, and (e) BC + CSAM.

4.3. Comparison of Different Compressed Sensing Networks

In this subsection, we compared ACRM with other algorithms and demonstrated the advantages of our network in reconstruction tasks. In this experiment, the Fashion-MNIST and CelebA datasets were used as datasets. For the Fashion-MNIST dataset, our parameter settings were the same as in the first subsection. For the CelebA dataset, scaled and cropped to a standard size of 64×64 , 200,000 images and 700 images were selected as training and test sets, respectively. The batch size of each entry into the network was 64. Each training was 3125 batches. Total training was 20 rounds. Regardless of the dataset used, we tested with images from the test set and calculated the average of the metrics. A simulated sampling network (FC layer) was added to all the networks used for comparison to ensure the fairness of the experiments. Related work proved that this method improves the image reconstruction accuracy of the network [36], so our modification did not affect the performance of the original network. The networks used for comparison in the experiments were Recon-Net [16], DR2-Net [17], and Bsr2-Net [43].

The experimental results are shown in Tables 5 and 6. The ACRM achieved the highest PSNR and SSIM values among all comparative experiments. Figure 6 shows the PSNR difference and loss convergence for each reconstruction model under different datasets. Figures 6 and 7 show the reconstruction samples of different reconstruction networks. They show that ACRM improved the learning ability and expressive ability of the network. In addition, it can be seen from (b) and (d) in Figure 6 that the ACRM had good efficiency on different datasets. Although the decline rate of loss in the early stage was slightly lower than that of Recon-Net, it overtook Recon-Net in the later stage.

Table 5. PSNR and SSIM of different algorithms are performed on the Fashion-MNIST dataset with different measurement rates.

Methods	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Recon-Net	17.601	0.796	15.039	0.693	14.37	0.639	12.094	0.519
DR2-Net	17.784	0.804	15.956	0.72	15.046	0.683	12.741	0.56
Bsr2-Net	17.885	0.796	16.304	0.749	15.357	0.695	13.261	0.598
ACRM (1.1)	18.12	0.817	16.673	0.757	15.743	0.719	13.438	0.603
	$\sigma = 6$		$\sigma = 6$		$\sigma = 4.75$		$\sigma = 4$	

Note: Bold numbers indicate the best value at that sample rate. The value in parentheses next to “ACRM” is the value of α (β is 1 by default); σ is the standard variance of the Gaussian filter.

Table 6. PSNR and SSIM of different algorithms are performed on the CelebA dataset with different measurement rates.

Methods	MR = 0.1		MR = 0.05		MR = 0.03		MR = 0.01	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Recon-Net	22.347	0.843	20.507	0.764	18.91	0.681	16.136	0.524
DR2-Net	20.893	0.776	19.602	0.712	18.49	0.651	15.949	0.509
Bsr2-Net	20.833	0.772	19.834	0.726	18.727	0.669	15.969	0.51
ACRM (1.2)	22.38	0.843	20.543	0.767	19.027	0.69	16.188	0.535
	$\sigma = 7$		$\sigma = 6$		$\sigma = 5.75$		$\sigma = 2$	

Note: Bold numbers indicate the best value at that sample rate. The value in parentheses next to “ACRM” is the value of α (β is 1 by default); σ is the standard variance of the Gaussian filter.

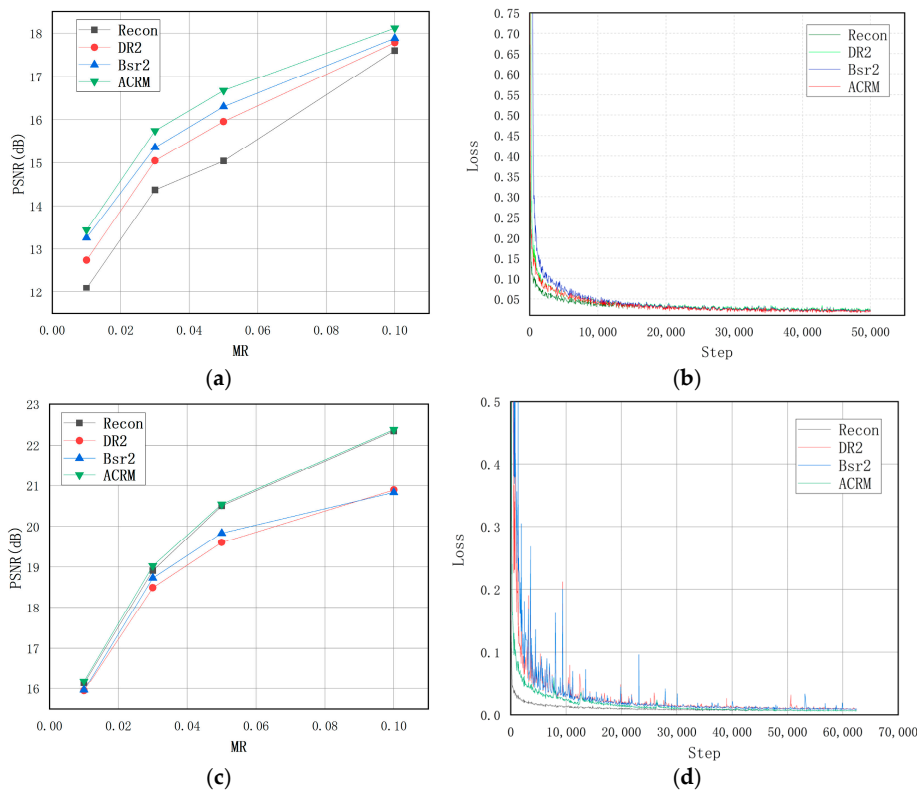


Figure 6. (a) PSNR of different algorithms at different measurement rates on the Fashion-MNIST

dataset. (b) Loss values of test images of different algorithms on the Fashion-MNIST dataset. (c) PSNR of different algorithms at different measurement rates on the CelebA dataset. (d) Loss values of test images of different algorithms on the CelebA dataset.



Figure 7. Model images are reconstructed on the CelebA dataset, MR = 0.1. From top to bottom are the (a) original image, (b) Recon-Net reconstructed image, (c) DR2-Net reconstructed image, (d) Bsr2-Net reconstructed image, and (e) ACRM reconstructed image.

5. Conclusions

In this study, we propose a deep learning-based compressed sensing network (ACRM) for single-pixel imaging. The ACRM combines a coordinated self-attention mechanism and uses an adaptive Gaussian filtering method to make up for the insufficiency of global average pooling. Then, the AMLoss, which incorporates attention ideas, is used for optimization. A series of experiments shows that the combination of the CSAM and adaptive Gaussian filter can fully utilize the global spatial information and channel information. Compared with the traditional compressed sensing neural network, the ACRM has a better reconstruction effect and reconstruction accuracy. It retains rich semantic information and has good network efficiency at a low measurement rate. It performs well in different datasets, indicating its strong robustness. After theoretical and experimental demonstrations, the AMLoss has a better reconstruction effect for the attention mechanism.

Author Contributions: Conceptualization, Z.W., Q.Y. and X.L.; Methodology, Z.W., Q.Y. and Y.Z.; Software, Z.W., Y.Z., S.S. and J.L.; Validation, Z.W., Y.Z., S.S. and J.L.; Formal analysis, Z.W., Q.Y., S.S. and J.L.; Investigation, Z.W., Q.Y., X.L., S.S. and J.L.; Resources, Q.Y. and X.L.; Data curation, Z.W. and Q.Y.; Writing—original draft, Z.W.; Writing—review & editing, Z.W. and Q.Y.; Visualization, Z.W.; Supervision, Q.Y., X.L. and Y.Z.; Project administration, Q.Y. and X.L.; Funding acquisition, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number No. 62165009.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kasin, B. The widths of certain finite-dimensional sets and classes of smooth functions. *Izv. Akad. Nauk SSSR Ser. Mat.* **1977**, *41*, 334–351.
2. Candes, E.; Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse Probl.* **2007**, *23*, 969. [[CrossRef](#)]
3. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
4. Tsaig, Y.; Donoho, D.L. Extensions of compressed sensing. *Signal Process.* **2006**, *86*, 549–571. [[CrossRef](#)]

5. Candès, E.J. Compressive sampling. In Proceedings of the International Congress of Mathematicians, Madrid, Spain, 22–30 August 2006; pp. 1433–1452.
6. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [[CrossRef](#)]
7. Duarte, M.F.; Davenport, M.A.; Takhar, D.; Laska, J.N.; Sun, T.; Kelly, K.F.; Baraniuk, R.G. Single-pixel imaging via compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 83–91. [[CrossRef](#)]
8. Li, W.; Hu, X.; Wu, J.; Fan, K.; Chen, B.; Zhang, C.; Hu, W.; Cao, X.; Jin, B.; Lu, Y. Dual-color terahertz spatial light modulator for single-pixel imaging. *Light Sci. Appl.* **2022**, *11*, 191. [[CrossRef](#)]
9. Lin, J.; Yan, Q.; Lu, S.; Zheng, Y.; Sun, S.; Wei, Z. A Compressed Reconstruction Network Combining Deep Image Prior and Autoencoding Priors for Single-Pixel Imaging. *Photonics* **2022**, *9*, 343. [[CrossRef](#)]
10. Lustig, M.; Donoho, D.L.; Santos, J.M.; Pauly, J.M. Compressed sensing MRI. *IEEE Signal Process. Mag.* **2008**, *25*, 72–82. [[CrossRef](#)]
11. Vasudeva, B.; Deora, P.; Bhattacharya, S.; Pradhan, P.M. Compressed Sensing MRI Reconstruction with Co-VeGAN: Complex-Valued Generative Adversarial Network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 672–681.
12. Liu, X.-J.; Xia, S.-T.; Fu, F.-W. Reconstruction guarantee analysis of basis pursuit for binary measurement matrices in compressed sensing. *IEEE Trans. Inf. Theory* **2017**, *63*, 2922–2932. [[CrossRef](#)]
13. Nguyen, N.; Needell, D.; Woolf, T. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Trans. Inf. Theory* **2017**, *63*, 6869–6895. [[CrossRef](#)]
14. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [[CrossRef](#)]
15. Li, C.; Yin, W.; Jiang, H.; Zhang, Y. An efficient augmented Lagrangian method with applications to total variation minimization. *Comput. Optim. Appl.* **2013**, *56*, 507–530. [[CrossRef](#)]
16. Kulkarni, K.; Lohit, S.; Turaga, P.; Kerviche, R.; Ashok, A. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 449–458.
17. Yao, H.; Dai, F.; Zhang, S.; Zhang, Y.; Tian, Q.; Xu, C. Dr2-net: Deep residual reconstruction network for image compressive sensing. *Neurocomputing* **2019**, *359*, 483–493. [[CrossRef](#)]
18. Babacan, S.D.; Molina, R.; Katsaggelos, A.K. Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **2009**, *19*, 53–63. [[CrossRef](#)]
19. Mousavi, A.; Patel, A.B.; Baraniuk, R.G. A deep learning approach to structured signal recovery. In Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 29 September–2 October 2015; pp. 1336–1343.
20. Mousavi, A.; Baraniuk, R.G. Learning to invert: Signal recovery via deep convolutional networks. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2272–2276.
21. Lohit, S.; Kulkarni, K.; Kerviche, R.; Turaga, P.; Ashok, A. Convolutional neural networks for noniterative reconstruction of compressively sensed images. *IEEE Trans. Comput. Imaging* **2018**, *4*, 326–340. [[CrossRef](#)]
22. Yang, Y.; Sun, J.; Li, H.; Xu, Z. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 521–538. [[CrossRef](#)]
23. Yuan, Z.; Jiang, M.; Wang, Y.; Wei, B.; Li, Y.; Wang, P.; Menpes-Smith, W.; Niu, Z.; Yang, G. SARA-GAN: Self-attention and relative average discriminator based generative adversarial networks for fast compressed sensing MRI reconstruction. *Front. Neuroinform.* **2020**, *14*, 611666. [[CrossRef](#)]
24. Zhang, X.; Wu, X. Attention-guided image compression by deep reconstruction of compressive sensed saliency skeleton. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13354–13364.
25. Barranca, V.J. Neural network learning of improved compressive sensing sampling and receptive field structure. *Neurocomputing* **2021**, *455*, 368–378. [[CrossRef](#)]
26. Hayhoe, M.; Ballard, D. Eye movements in natural behavior. *Trends Cogn. Sci.* **2005**, *9*, 188–194. [[CrossRef](#)]
27. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [[CrossRef](#)]
28. Zheng, X.; Wang, B.; Du, X.; Lu, X. Mutual attention inception network for remote sensing visual question answering. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5606514. [[CrossRef](#)]
29. Zheng, X.; Chen, W.; Lu, X. Spectral super-resolution of multispectral images using spatial-spectral residual attention network. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5404114. [[CrossRef](#)]
30. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
31. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
32. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

33. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
34. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3258.
36. Guan, Y.; Yan, Q.; Yang, S.; Li, B.; Cao, Q.; Fang, Z. Single photon counting compressive imaging based on a sampling and reconstruction integrated deep network. *Opt. Commun.* **2020**, *459*, 124923. [[CrossRef](#)]
37. Ma, X.; Guo, J.; Tang, S.; Qiao, Z.; Chen, Q.; Yang, Q.; Fu, S. DCANet: Learning connected attentions for convolutional neural networks. *arXiv* **2020**, arXiv:2007.05099.
38. Qin, Z.; Zhang, P.; Wu, F.; Li, X. Fcanet: Frequency channel attention networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 783–792.
39. Abraham, N.; Khan, N.M. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 683–687.
40. Wang, W.; Su, C. Convolutional neural network-based pavement crack segmentation using pyramid attention network. *IEEE Access* **2020**, *8*, 206548–206558. [[CrossRef](#)]
41. Yu, J.; Wu, B. Attention and hybrid loss guided deep learning for consecutively missing seismic data reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5902108. [[CrossRef](#)]
42. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
43. Li, B.; Yan, Q.-R.; Wang, Y.-F.; Yang, Y.-B.; Wang, Y.-H. A binary sampling Res2net reconstruction network for single-pixel imaging. *Rev. Sci. Instrum.* **2020**, *91*, 033709. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.