Exploiting Side Information and Scalability in Compressed Sensing and Deep Learning

by

Xing Wang

M.Sc., Beijing University of Posts and Telecommunications, 2012
B.Sc., Huazhong Universrity of Science and Technology, 2009

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Engineering Science
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Fall 2016

Approval

Name:	**Xing Wang**
Degree:	**Doctor of Philosophy (Engineering Science)**
Title:	***Exploiting Side Information and Scalability in Compressed Sensing and Deep Learning***
Examining Committee:	**Chair:** Ivan Bajić Associate Professor

Jie Liang
Senior Supervisor
Professor _____

Zhaosong Lu
Supervisor
Associate Professor
Department of Mathematics _____

Faisal Beg
Internal Examiner
Professor _____

Z. Jane Wang
External Examiner
Professor
Department of Electrical and Computer Engineering
University of British Columbia _____

Date Defended: November 2, 2016

Abstract

There is a tremendous demand for increasingly efficient ways of both capturing and processing high-dimensional datasets of large size. When capturing such datasets, a promising recent trend has developed based on the recognition that, many high-dimensional datasets have low-dimensional structures. For example, the notion of sparsity is a requisite in the compressed sensing (CS) field, which allows for accurate signal reconstruction from sub-Nyquist sampled measurements given certain conditions. When processing such datasets, the recently developed deep learning is a powerful tool, able to extract high-level and complex abstractions from massive amounts of data.

CS has a wide range of applications that include imaging, radar and many more. Much effort has been put on developing more accurate and efficient reconstruction algorithms. In this thesis, first, we are interested in how to incorporate the side information into CS reconstruction when there is an initial estimation of the sparse signal available from other sources. Rigorous theoretical analysis was proposed for the first time in this field. Sufficient number of measurements is required for accurate CS reconstruction. We may have to wait for a long time to do the reconstruction until we receive enough measurements, which could incur undesired delays. Moreover, state-of-the-art CS reconstruction algorithms are still inefficient for signals of large size, e.g., images. Inspired by the multi-resolution or scalable reconstruction in multimedia transmission, such as JPEG 2000 and H.264/SVC, in the second part of this thesis, we analyzed scalable CS reconstruction problem and proposed to reconstruct a low-resolution signal if the number of measurements is too small.

Deep learning or deep neural networks (DNNs) has evolved into the state-of-the-art technique for many artificial intelligence tasks including computer vision, speech recognition and natural language processing. However, DNNs generally involve many layers with millions of parameters, making them difficult to be deployed and updated on devices with limited resources such as mobile phones and other smart embedded systems. Moreover, if the DNN needs to be updated, usually via wireless communications, downloading the large amount of network parameters will cause excessive delay. In the final part of this thesis, we propose a scalable representation of the network parameters, so that different applications can select the most suitable bit rate of the network based on their own storage constraints.

Keywords: Compression; side information; scalability; compressed sensing; deep neural network

Acknowledgements

During the completion of this doctoral work, I have been accompanied and supported by many people. It is my great pleasure to take this opportunity to thank all of them.

First of all, I would like to express my deepest gratitude to my senior supervisor, Professor Jie Liang, who influenced me most and guided me during the study towards my doctoral degree at Simon Fraser University. Dr. Liang is a great professor and great person. I am very grateful for the free pass he gave to explore the unknown and the generous support he offered to present my work in academic conferences. Without his supervision, none of my research work in this thesis would have been finished.

I would also like to express my sincere gratitude to my supervisor Zhaosong Lu, for his constructive advices and comments that have made substantial contributions to this thesis. I would also like to thank Professor Mirza Faisal Beg for serving as the internal examiner of my thesis.

I also gratefully thank the external examiner, Professor Z. Jane Wang at University of British Columbia, and defense chair Professor Ivan V. Bajić, for their precious time to attend the defense in the midst of their busy activities.

I would thank all the amazing members at the Multimedia Communication Laboratory, for making my life during the last four years a fun, challenging and memorable one. Especially, I would like to thank Yu Gao, Dong Zhang, Chongyuan Bi, Xiao Luo, Siyu Wu, Yijian Wang, Him Wai Ng, Rui Wang, Lei Liu, Mehdi Seyfi, Mohammad Akbari, Golara Javadi, Omar Alhussein and Setareh Dabiri to name a few, for their insightful comments and helpful discussions.

Finally, I would like to dedicate this thesis to my parents, for their endless love, limitless encouragement and unselfish sacrifice throughout my doctoral education.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, there have been massive increases in both the dimensionality and sample sizes of data due to ever-increasing demand coupled with relatively inexpensive sensing technologies. As of March 2014, it was estimated that 90 % of all data was generated over the course of past two years [102]. Without question, society has entered the era of "Big Data". These high-dimensional datasets bring challenges, along with numerous opportunities.

One critical challenge concerns the acquisition of such data. Many camera systems are capable of generating gigabytes of raw data in a short period of time-data which is then immediately compressed in order to discard unnecessary and redundant information. In some applications, this sample-then-compress paradigm is acceptable, while in others, where hardware costs dominate, it is much more preferable to compress while sampling, that is, collect only the truly necessary information. Compressed sensing (CS) is such kind of technique that combines sampling with compression and can sample the signal at a rate much lower than Nyquist sampling rate.

With "Big Data" also comes many opportunities. "Big Data" has become important as many organizations both public and private have been collecting massive amounts of domain-specific information, which can contain useful information about problems such as national intelligence, fraud detection, marketing, and medical informatics. Deep neural networks (DNNs) or deep learning have become ubiquitous in applications ranging from computer vision [60] to speech recognition [5] and natural language processing [20]. A key benefit of Deep Learning is the analysis and learning of massive amounts of data, making it a valuable tool for Big Data Analytics. One key factor that makes deep learning so powerful is large dataset. Take ImageNet [82], which is one of the most famous datasets in computer vision community, as an example. It has 1.3M training images and 50K validation images. Large dataset makes deep learning be able to extract high-level, complex abstractions.

In this thesis, we put our focus on two compression concepts in "Big Data", one is about the data acquisition and compression in "Big Data", closely related to CS, and the other

one is about the deep learning model compression in "Big Data". Our main goal is to tackle some important topics in these two compression concepts from image/vide coding's point of view. The first topic is side information-aided compressed sensing problem where a sparse signal is sampled via a noisy underdetermined linear observation system, and an additional initial estimation of the signal is available during the reconstruction. Our goal here is to take advantage of this side information and improve the CS reconstruction performance. There is plenty of work in this area, but only loose theoretical bounds are provided and rigorous theoretical analysis is missing. Motivated by distributed source coding, we model this initial estimation as the side information for the decoder if we treat CS as a coding problem. Then we incorporate this prior information into CS reconstruction, and present theoretical analysis to measure the gain brought by the side information. More details can be found in Chapter 2 and 3.

The next two topics, scalable compressed sensing and scalable compression of deep neural networks, are motivated by scalable image/video coding.

The first one is scalable compressed sensing. According to CS theory, in order to get bounded reconstruction error, the number of samples has to be larger than a minimal requirement. For high-resolution images, a large number of CS samples are stilled needed. In applications composed of transceivers, undesired delay is inevitable since the receiver has to wait for a long time to start the reconstruction. If we turn to recover the corresponding low-resolution preview instead of original high-resolution image, can we get bounded reconstruction error? And how can we determine the resolution of this low-resolution preview? Chapter 4 is devoted to answer these questions. A multi-resolution CS problem is formulated and a multi-resolution algorithm is developed. More importantly, strong theoretical guarantees are presented to prove the efficiency of proposed multi-resolution algorithm.

Next, current deep learning model compression methods can only achieve single compressed bit rate. However, different devices have different bit constraints. To take every device's constraint into consideration, a scalable compression pipeline is needed to make the compression scalable and flexible. Motivated by scalable image/video coding, a scalable compression method of deep neural networks is proposed in Chapter 5. Similar to base layer and enhancement layer in scalable image/video coding, we introduce hierarchical quantization, composed of base quantization layer and enhancement quantization layers, to quantize the weights in DNN models. Next, motivated by the layer formulation in JPEG2000 scalable image compression, we design backward search bit allocation scheme to select the bits for each network layer based on the bit constraints.

1.1 Background

To facilitate the understanding of our contributions, we review some important background knowledge that is closely related to the work in this thesis, including compressed sensing, multiview image and video, and deep learning.

1.1.1 Compressed Sensing

Shannon-Nyquist theory informs us that sample acquired uniformly in time or space at twice the highest signal bandwidth (or desired resolution) can be used to accurately reconstruct the signal through a simple, computationally inexpensive process known as sinc interpolation. CS differs from Shannon-Nyquist sampling in several important respects. First, it is primarily studied as a finite-dimensional,digital-to-digital sampling scheme, although continuous-time sampling is possible within the theory [4]. Second, CS requires a more complex sampling process; rather than acquiring point samples uniformly in time/space, CS collects samples in the form of inner products between the complete signal and a series of "test waveforms." Third, unlike sinc interpolation, the CS inverse problem is highly nonlinear in nature, requiring more complex reconstruction algorithms. In exchange for these trade-offs, CS offers the ability to dramatically reduce the number of samples that must be acquired without sacrificing reconstruction fidelity.

Mathematically, CS is the problem of reconstructing a sparse signal from its noisy underdetermined linear measurement. In this case, the observations $\mathbf{y} \in \mathcal{R}^m$ can be written as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1.1}$$

where $\mathbf{x} \in \mathcal{R}^n$ is a k-sparse signal, *i.e.*, with k nonzero entries ($k \ll n$). $\mathbf{A} \in \mathcal{R}^{m \times n}$ is a known linear measurement matrix, and $\mathbf{w} \in \mathcal{R}^m$ is an additive white Gaussian noise with variance σ^2, *i.e.*, $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

In this thesis, the following ratios are frequently used:

$$\delta = m/n, \;\; \varepsilon = k/n, \;\; \rho = \varepsilon/\delta = k/m. \tag{1.2}$$

When $m < n$, the problem is underdetermined and has been studied extensively recently via the compressed sensing (CS) theory. It is shown in [15] that when $\mathbf{A}$ satisfies certain condition and m is larger than some bound, ℓ_1-based algorithms can successfully recover the sparse signal, written as

$$\arg\min_{\mathbf{x}} \|\mathbf{x}\|_1, \;\; \text{s.t.} \;\; \mathbf{y} = \mathbf{A}\,\mathbf{x}. \tag{1.3}$$

Many reconstruction algorithms have been developed to estimate the sparse signal $\mathbf{x}$ from $\mathbf{y}$, including, *e.g.*, convex optimization [15], greedy method [95], and iterative thresholding algorithm [11].

Estimation theory can also be used to analyse the performance of CS. In [78], with the help of the replica method from statistical physics, a sharp prediction is derived for the performance of the LASSO or Basis Pursuit Denoising method (BPDN) [18, 93]

$$\arg\min_{\mathbf{x}} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \tau \|\mathbf{x}\|_1 \right), \tag{1.4}$$

where τ is a weighting parameter that enforces the sparsity constraint, which is an ℓ_1-regularized least-square optimization problem. However, the replica assumption is not rigorous and it cannot be checked for specific problems.

In [31, 32, 68, 72], an approximate message passing (AMP) algorithm is developed for Gaussian sampling matrices, which reduces the complexity of classic message passing [61]. More importantly, the AMP is rigorous and can predict the final reconstruction performance accurately. Some generalizations of AMP have been developed. For example, in [76], a generalized AMP (GAMP) is developed to handle arbitrary noise distributions and arbitrary prior distributions. In [100], the Gaussian mixture model and expectation-maximization (EM) algorithm are used to learn the distribution of the signal's nonzero coefficients.It is also shown empirically and theoretically that AMP-type solvers work well with various types of matrices, such as Rademacher matrices and Fourier matrices [10, 78, 100].

1.1.2 Multiview Image and Video

In the past decade, multiview imaging (MVI) has attracted increasing attention, thanks to the rapidly dropping cost of digital cameras. This opens a wide variety of interesting research topics and applications, such as virtual view synthesis, 3DTV, and Free Viewpoint TV (FTV) [62]. Conventional two-dimensional (2D) video provides a fixed viewpoint of recorded objects where viewers can only watch a video playback passively, as the viewpoint remains the same throughout video playback. In contrast, multiview video (MMV) consists of video sequences of the same scene captured time-synchronously by multiple closely spaced cameras from different observation viewpoints. This means that each viewer of the same video content can observe various viewpoints of a scene from different angles and locations, which further generate a free viewpoint video (FVV) or create realistic three-dimensional (3D) perceptions.

Several prototypes of such MMV systems have demonstrated a much improved viewing experience compared to 2D video. As seen in the movie The Matrix [3], successive switching of multiple real images captured at different angles can give the sensation of a flying viewpoint. In addition, MMV system EyeVision [1] was used for broadcasting Super Bowl XXXV, in which 33 cameras were arranged around the stadium and the camera directions

were controlled mechanically to track the target scene. In these systems, no new virtual images are generated, and the movement of the viewpoint is limited to the predefined original camera positions. However, some MMV systems, such as FTV, do not impose the constraint that a selected viewpoint corresponds to one existing camera, but instead, allows the selection of an arbitrary viewpoint within 3D scene. New virtual views are generated from neighbouring captured views using 3D geometry.

1.1.3 Deep Learning

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations [43]. Deep learning has evolved into the state-of-the-art technique for many artificial intelligence tasks including computer vision [49, 50, 60, 86], speech recognition [5] and natural language processing [20]. For 1000-class ImageNet image classification challenge [82], the top-5 error has been reduced from 25.7% with hand-crafted features to 17.0% with AlexNet in 2012 and more recently, 4.94% with ResNet in 2015 [49], which has surpassed human-level performance. For spontaneous speech recognition, the word error rate is decreased from nearly 25% with Hidden Markov Models to almost 5% with deep neural network in 2012 [23].

In this thesis, we focus on the convolutional neural network (CNN), the most widely used DNN, which was originally developed in 1998 by LeCun et al. [63] with less than 1M parameters to classify handwritten digits. CNN has attracted much attention since 2012 when AlexNet [60], the winner of ILSVRC2012 (Large Scale Visual Recognition Challenge 2012), was first proposed and outperformed the second place where hand-crafted feature extraction was used significantly. To name a few, CNN-based architectures have become the state-of-the-art in object detection [39, 40, 79], face recognition [75] and image segmentation [67]. The filters in the first convolution layer of AlexNet is visualized in Figure 1.1. We can see that edge and color information are well captured.

Different from the traditional model of pattern recognition where hand-crafted feature extractor and trainable classifier are used, deep learning is end-to-end learning, where the features and the classifier are automatically learned and trained simultaneously. CNN models usually consist of several convolutional (CONV) layers , pooling layers and fully connected (FC) layers, which are stacked up with one on top of another. There are five CONV layers and three FC layers in AlexNet [60]. The second place of ILSVRC2014, VGG-16 [86], has thirteen CONV layers and three FC layers. The recent winner of ILSVRC2015, ResNet [49] has more than one hundred layers.

Besides CNN, there are some other basis deep learning architectures, e.g., auto-encoder [101], RBM (Restricted Boltzmann Machine) [83], and LSTM (long short-term memory) [51], which are beyond the scope of this thesis.

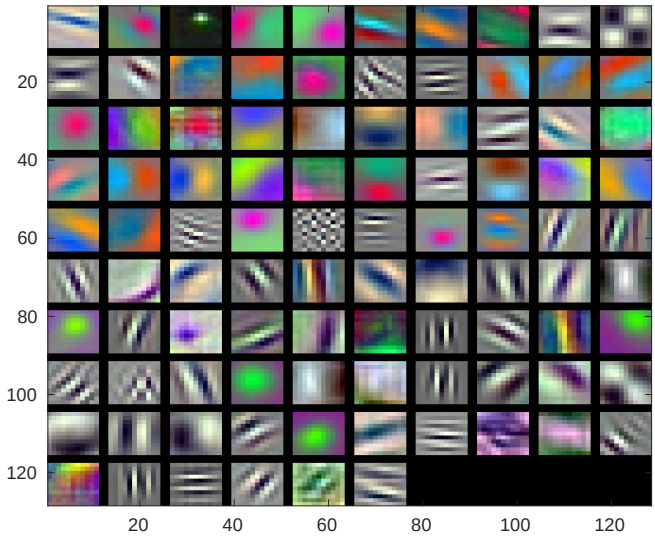


Figure 1.1: Visualization of filters in CONV1 of AlexNet

1.2 Related Work

In this thesis, some novel and not well-understood topices for CS and DNN are studied. For CS, we focus on side information (SI)-aided CS reconstruction and scalable CS reconstruction. For DNN, we are interested in the scalable compression of DNN models. Scalability idea, which is originally inspired by multi-resolution or scalable reconstruction in multimedia transmission, e.g., JPEG2000 [92] and H.264/SVC [85], is applied to both CS and DNN.

1.2.1 Side Information-Aided Compressed Sensing

There have been some efforts on exploiting various initial estimations in CS. One example is the CS problem with partially known support [97], which shows that by finding the signal that satisfies the measurement constraint and is the sparsest outside the partially known support, the CS reconstruction can be improved, and bounds on the reconstruction error are derived. However, the method is time-consuming. Another relevant approach is to recover the estimation error instead of the sparse signal [94], based on the assumption that the prediction error between the initial estimation and the sparse signal is sparser than the signal itself, and is thus easier to be recovered, but this method lacks theoretical analysis. It is also possible that the prediction error is denser than the original sparse signal, if the initial estimation has poor quality.

In [12], the belief-propagation-based CS framework (BPCS) in [9] is used to exploit the SI from neighboring cameras in multiview imaging systems, where the SI is used as the starting point for belief propagation. In [103], a squared-error-constrained penalty term is added to

the CS of multiview images. It also considers a more general case, where the variances of the prediction errors are different at different entries. A fast solution is developed based on the Gradient Projection for Sparse Reconstruction (GPSR) algorithm [35].

The sparsity-constrained dynamic system estimation scheme proposed in [17] and the dynamic compressed sensing via approximate message passing (DCS-AMP) proposed in [108,109] are closely related to our framework. In [17], a prediction of the signal is obtained from the state evolution model, and the norm of the prediction error is added as a penalty term in the objective function of LASSO or BPDN method. In [108,109], the sparse signal is modeled as the Bernoulli-Gaussian distribution and the correlation between the active amplitudes in different time slots is assumed to be a stationary steady-state Gaussian-Markov process. The EM and AMP are applied to learn the hidden parameters and perform the inference. Although the model in [108,109] is similar to ours, it relies on sequential data to learn the hidden parameters, and cannot be applied to solve the problem discussed here directly. In fact, it is not clear how to extend the method in [109] to solve the problem in this paper.

Several papers have also studied the theoretical contribution of the prior knowledge [54, 97]. In [97], the authors have provided some sharp bounds on the necessary number of CS measurements to successfully reconstruct the original sparse signal, based on null space property and geometry interpretations. However, it is mainly on the noiseless case. The performance of noisy case remains unknown. Kamilov et al. have taken the first step towards a theoretical understanding of EM-based algorithms [54, 108, 109], although the complete analysis is still not available.

1.2.2 Scalable Compressed Sensing

It is proven in [15], [95] and [31] that a minimum number of samples is required to ensure stable and accurate CS reconstruction. Therefore, in applications in which a large number of CS samples need to be transmitted to a receiver, undesired delay is inevitable. Although the need for multi-resolution (MR) or scalable reconstruction has been well recognized in multimedia transmission, leading to the development of standards such as JPEG 2000 and H.264/SVC [85, 92], the problem has received little attention in CS.

In [53], some rules are proposed to design efficient up-/down-sampling matrices for MR reconstruction, and the number of nonzero entries of the LR image in the transform domain is shown to be no larger than that of the HR image. Therefore, the required sampling rate for stable LR image reconstruction is less than that of HR reconstruction. However, the analysis in [53] is qualitative, and only some loose bounds are provided. Moreover, the impact of the MR design on the quality of the measurement matrix, which can be measured by, e.g., the restricted isometry property (RIP) constant [15] and mutual coherence [95], is not studied. Moreover, only the noiseless case is considered in [53].

A similar problem is studied in [107], where two solutions are proposed. In the first method, the sampling matrix is designed to have non-uniform sampling, which is quite restrictive because the matrix should be redesigned whenever a new result with a different resolution is needed. The second method modifies the sampled data of the HR image to be similar to the data acquired directly from the target LR image. Although it works empirically, there is no theoretical guarantee of its performance. In addition, although it is mentioned in [107] that the CS sampling rate for the LR reconstruction is increased, the change in the sparsity rate is not considered. Recently, a special two-resolution CS reconstruction scheme was proposed in [41], where the sampling matrix is designed such that an LR reconstruction can be obtained via direct matrix inversion.

The MR concept has also been used in certain CS schemes, such as [16, 52, 74, 90, 96], with different purposes from ours. In [52], Bayesian CS is used to detect the primary user in cognitive radio. The method first performs the detection in LR and then refines the signal around the detected primary user spectrum. In [96], a CS-based two-layer scalable image coding is proposed, where the encoder employs two measurement matrices with different sizes, and inter-layer prediction is used to reduce the bit rate. In [16], the authors extended the Kronecker CS [33] to MR measurements such that the sensing is performed on the LR image, and the goal is to recover the HR signal from LR measurements. In [74], a multiscale framework is proposed for the CS of videos. The motion vectors are estimated at different resolutions and serve as the input to higher resolution frame recovery. The sensing is applied to different resolutions for the same frame.

1.2.3 Compression of Deep Neural Networks

Although DNNs have recently led to significant improvement in countless areas of machine learning, its application in low-end devices such as mobile phones or smart hardware faces some challenges. For example, many devices have limited storage spaces. Therefore storing millions of DNN parameters on these devices could be a problem. If the DNN network needs to be updated, usually via wireless communications, downloading the large amount of network parameters will cause excessive delay. Moreover, running large-scale DNNs with floating-point parameters could consume too much energy and slow down the algorithm. Therefore, efficient compression of the DNN parameters without sacrificing too much the performance becomes an important topic.

There have been some recent works on the compression of neural networks. Vanhoucke et al. [65] proposed a fixed-point implementation with 8-bit integer (vs 32-bit floating-point) activations. Denton et al. [25] exploited the linear structure of the neural network by finding an appropriate low-rank approximation of the parameters and keeping the accuracy within 1% of the original model. Kim et al. [58] applied tensor decomposition to the network parameters and proposed an one-shot whole network compression scheme that can achieve significant reductions in model size, runtime and energy consumption.

Much work has been focused on binning the network parameters into buckets, and only the values in the bucket need to be stored. HashedNets [19] is a recent technique to reduce model size by using a hash function to randomly group connection weights, so that all connections within the same hash bucket share a single parameter value. Gong et al. [42] compressed deep convnets using vector quantization, which resulted in 1% accuracy loss. Both methods studied the fully-connected (FC) layer in the CNN, but ignored the convolutional (CONV) layers. Recently, Han et al. [45] introduced a deep neural network compression pipeline by combining pruning, quantization and Huffman encoding, which can reduce the storage requirement of neural network by 35 $\times$ or 49 $\times$ without affecting their accuracy.

1.2.4 Sparsity-Constrained Deep Learning

It has been shown that current state-of-the-art deep CNN models are redundant [24]. Many researchers have tried to reduce this redundancy with sparsity constraint. In [46], small weights in the pretrained models are pruned to zero without any accuracy loss after fine-tuning. In [66], maximum sparsity is obtained by exploiting both inter-channel and intra-channel redundancy. More than 90% of parameters are zeroed out with less than 1% accuracy drop on the ILSVRC2012 dataset. Moreover, an efficient sparse matrix multiplication algorithm on CPU is proposed. Structured sparsity learning for structures of filters, channels, filter shapes and depth in DNNs is proposed in [105]. First, a compact structure from a bigger DNN is learned. Second, a great speedup of CONV layer computation is achieved. Finally, regularizing the DNN structure with structured sparsity can improve classification accuracy. In [104], deep double sparsity encoder is developed to simultaneously sparsify the output features and the learned model parameters. Also, a compact model size and low complexity is achieved.

1.3 Contributions

The contributions of this thesis include:

- the design of new recovery algorithms for SI-aided CS problem that an initial estimation of the sparse signal is available;
- the analysis of these new SI-based recovery algorithms to provide performance guarantees, as related to the distortion of the recovered signal, the number of linear measurements required for recovery and the quality of SI;
- the design of new recovery algorithms for multi-resolution CS problem that prefer to recover the LR preview of the target signal, rather than the original HR signal, when the number of linear measurements is insufficient for HR signal recovery;

- the analysis of new multi-resolution CS recovery algorithms, and provide performance guarantees about the distortion of the recovered LR signal and the number of linear measurements;

- the design of scalable compression framework of DNNs that represents the DNN parameters in a scalable fashion such that we can easily update the representation of the network according to the storage constraint.

1.4 Outline

This thesis is organized as follows.

Chapter 2 describes initial work on the use of side information in hybrid multi-view imaging system to improve the performance of CS. We provide algorithms that exploit the view interpolation confidence map generated by view interpolation algorithms, incorporate this information into CS reconstruction and present experimental evidence of the advantage by the use of this side information.

Chapter 3 builds on the work in Chapter 2 by presenting a theoretical and algorithmic framework for the use of initial estimation in CS. We derive the rigorous theoretical analysis on the reconstruction performance bound with respect to the number of linear measurements required for recovery and the quality of side information, and show this new bound outperforms the one corresponding to the scenario that either linear measurements or the initial estimation exist. Moreover, we also present a parameterless version of proposed algorithm that no parameters need to be manually tuned. Extensive applications of the proposed algorithm are discussed.

Chapter 4 introduces the multi-resolution CS problem and provides a theoretical and algorithmic framework to solve this problem. In addition to the reduced complexity, our method can choose to recover an LR signal of a proper resolution stably based on the number of linear CS measurements at hand, even when the reconstruction of HR signal is unstable. We then apply the algorithm to image reconstruction using either soft-thresholding or a total variation denoiser and develop three pairs of up/down-sampling operators in the transform or spatial domain. We also present experimental results that validate the advantages of these algorithms for synthetic datasets and for the real-word data.

Chapter 5 provides a scalable compression framework of DNN. We represent the DNN parameters in a scalable fashion so that different applications can select the most suitable bit rate of the network based on their own storage constraints. Moreover, when a device needs to upgrade to a high-rate network, the existing low-rate network can be reused, and only some incremental data are needed to be downloaded. Experimental results on various DNNs show that our method can achieve scalable compression with graceful degradation in the performance.

Finally, we conclude with a summary of our findings and a discussion of future work in **Chapter 6**.

This PhD research study resulted in the following publications:

Conference Papers:

1. X. Wang and J. Liang, View interpolation confidence-aided compressed sensing of multiview images, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1651-1655, May 2013.

2. X. Wang and J. Liang, Side information-aided compressed sensing reconstruction via approximate message passing, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 3354-3358, May 2014.

3. X. Wang and J. Liang, Multi-resolution compressed sensing reconstruction via approximate message passing, in Proc. IEEE Conf. on Image Proc. (ICIP), pp. 4352-4356, Sept. 2015.

4. X. Wang and J. Liang. Scalable compression of deep neural networks. In ACM Multimedia Conference (ACM MM), Oct. 2016

Journal Papers:

1. X. Wang and J. Liang, Approximate message passing-based compressed sensing reconstruction with generalized elastic net prior, Signal Processing: Image Communication, vol. 37, pp. 19-33, Sep. 2015.

2. X. Wang and J. Liang, Multi-resolution compressed sensing via approximate message passing, IEEE Trans. on Computational Imaging, vol. 2, no. 3, pp. 218-234, Sep. 2016.

1.5 Acronyms and Notations

In this section, we summarize the acronyms and some common notations used throughout this thesis.

1D	one-dimensional
2D	two-dimensional
CS	compressed sensing
GPSR	gradient projection for sparse reconstruction
LASSO	least absolute shrinkage and selection operator
BPDN	basis pursuit denoising method
AMP	approximate message passing
EM	expectation-maximization
SI	side information
PTC	phase transition curve
ST	soft-thresholding denoiser
TV	total variation denoiser
GENP	generalized elastic net prior
MR	multi-resolution
HR	high-resolution
LR	low-resolution
TVAL3	TV Minimization by Augmented Lagrangian and Alternating Direction Algorithms
PSNR	peak-signal-to-noise ratio
MSE	mean-squared-error
DCT	discrete cosine transform
SURE	Stein's unbiased risk estimate
DL	deep learning
DNN	deep neural network
CNN	convolutional neural network
CONV	convolutional layer
FC	fully-connected layer
ILSVRC	Large Scale Visual Recognition Challenge

Table 1.1: Lists of acronyms.

Chapter 2

View Interpolation Confidence-Aided Compressed Sensing

Multiview images are captured by a group of cameras from slightly different locations. Together with new display technologies such as free view-point TV and autostereoscopic displays, an immersive viewing experience can be achieved. However, multiview systems require higher costs for data acquisition, storage and transmission. Fortunately, in most multiview applications, there exist strong correlations between neighboring views. Therefore view-interpolation-based methods can be used to improve the compression efficiency [47, 106]. It can also be used to reduce the acquisition cost. In this paper, a hybrid multiview imaging system is considered, where traditional high-resolution cameras and emerging low-cost compressed sensing (CS) cameras are interleavingly placed. The key idea of the CS theory is that if a signal is sparse in some basis, it can be reconstructed with high quality via simple random sampling at the encoder and ℓ_1-norm optimization at the decoder [27]. Therefore the cost of the CS cameras can be lower than traditional cameras.

However, existing multiview imaging systems in [12, 94] have not fully exploited all information in view interpolation. First, it is known that view interpolation quality is highly dependent on the scene composition. Therefore, based on the overall frame-level confidence of the interpolated image provided by the view interpolation algorithm, we should have a mechanism to adjust the influence of the view interpolation result on the CS reconstruction.

Secondly, many view interpolation algorithms also provide confidence information at pixel level [47], in terms of the number of matching points a pixel of the interpolated view can have in the two neighboring views. Usually, pixels with two matching points have higher reconstruction quality. Pixels with only one matching point are occluded in one neighboring view, thereby having lower interpolation quality. For pixels without any correspondence in the neighboring views (corresponding to holes in the initial interpolated

image), various impainting methods have to be used to estimate their values. Therefore these pixels generally have the lowest confidence.

If these issues are not addressed properly, existing view interpolation-aided multiview CS reconstruction methods could perform even worse than direct CS reconstruction. In this part, we propose a modified GPSR algorithm by adding another term to the objective function. The term measures the squared error between the CS-based and view-interpolation-based reconstructions. The weighting parameters of this term are determined by both the frame-level and pixel-level confidences of the view interpolation result. We show that the modified method can still be converted to the GPSR framework. Simulation results demonstrate that the framework is very flexible and can outperform existing methods.

2.1 Background of CS and GPSR

The problem of reconstructing $\mathbf{x}$ from $\mathbf{y}$ is underdetermined. However, since $\mathbf{x}$ is sparse, the ℓ_1 optimization can be used. The problem can be efficiently solved via linear programming. However, for large-scale applications, the speed of the optimization algorithms can be very slow. Recently, a fast Gradient Projection for Sparse Representation (GPSR) algorithm has been developed [35], which starts with the unconstrained convex optimization problem in Eq. (1.4).

To solve this, it first decomposes $\mathbf{x}$ into its positive and negative parts.

$$\mathbf{x} = \mathbf{u} - \mathbf{v}, \mathbf{u} \geq 0, \mathbf{v} \geq 0. \tag{2.1}$$

The problem can then be converted to the following bound-constrained quadratic programming (BCQP) formulation of basis pursuit or similar problems [27].

$$\begin{aligned} &\min_{\mathbf{z}} \mathbf{c}^T\mathbf{z} + \tfrac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} \equiv F(\mathbf{z}), \\ &\text{s. t. } \mathbf{z} \geq 0, \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \ \mathbf{b} = \mathbf{A}^T\mathbf{y}, \ \mathbf{c} = \tau\mathbf{1}_{2N} + \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} \mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{A} \\ -\mathbf{A}^T\mathbf{A} & \mathbf{A}^T\mathbf{A} \end{bmatrix}. \end{aligned} \tag{2.3}$$

The solution to (2.2) is equal to the solution of (1.4) if the free parameter τ is much less than 1.

It is shown in [35] that good solutions can be obtained very fast by using gradient projection, special line search and termination techniques, making the GPSR method very attractive.

2.2 Generalized GPSR with View Interpolation Confidence

In this section, we propose a generalized optimization framework to consider the occlusions and holes in the interpolated image. We then show that the framework can be converted into the standard BCQP format, which can be efficiently solved by the GPSR algorithm.

2.2.1 Generalized Optimization Framework

Our goal is to reconstruct the middle image $\mathbf{I}_j$ from its linear CS measurements $\mathbf{y}$, with the help of the interpolated middle image $\widehat{\mathbf{I}}_j$ generated from the left and right reference images $\mathbf{I}_{j-1}, \mathbf{I}_{j+1}$ (given by conventional cameras). Due to the strong correlation between images in multi-view image system, the final reconstructed image should be generally close to the interpolated image. However, the quality of the interpolated image $\widehat{\mathbf{I}}_j$ is affected by the number of occlusion pixels and the size of the holes in it. Hence, if we reconstruct the difference image between $\mathbf{I}_j$ and $\widehat{\mathbf{I}}_j$ and add it back to $\widehat{\mathbf{I}}_j$ to get the reconstructed image, the performance could be even worse than directly reconstructing the middle image from its CS measurement, because the sparsity of the difference image could be larger than the sparsity of the original middle image in this case.

To resolve this potential issue, we propose the following generalized optimization framework.

$$\arg\min_{\mathbf{x}} \left(\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{X}\|_2^2 + \tau \|\mathbf{x}\|_1 + \frac{\mu}{2} \sum_{i=1}^{N} w_i \Big(I_{i,j} - \widehat{I}_{i,j} \Big)^2 \right), \tag{2.4}$$

where $\mathbf{x}$ is the sparse representation of $\mathbf{I}_j$ in basis $\mathbf{\Psi}$, *i.e.*, $\mathbf{I}_j = \mathbf{\Psi}\mathbf{x}$. The last squared-error term is new compared to the original GPSR in (1.4). $I_{i,j}$ and $\widehat{I}_{i,j}$ denote the i-th pixel of the target image $\mathbf{I}_j$ and the interpolated image $\widehat{\mathbf{I}}_j$, respectively. μ is a weighting parameter that is determined by the overall frame-level confidence of the view-interpolation algorithm, and w_i is the weighting parameter for the i-th pixel, which is determined by the pixel-level view interpolation confidence.

A larger value of μ can be used if the overall view interpolation has higher quality. In this case, the view interpolation is more trustworthy. On the other hand, μ should be smaller if there are many occlusion pixels and holes in the view interpolation; hence the CS reconstruction should rely more on the linear measurement from the CS camera.

Similarly, the pixel-level weighting parameter w_i should be larger when a pixel in the middle view has two point correspondences in the neighboring views. A smaller w_i should be used when there is only one point correspondence, *i.e.*, the pixel is occluded in one view. Finally, the smallest w_i should be used when no point correspondence can be found, as the

pixel is in a hole in the initial view interpolation. The occluded pixels and holes usually occur near the edges of objects in an image.

The impacts of μ and w_i will be studied in Sec. 2.3.

2.2.2 Conversion to the Standard BCQP Format

Let $\widehat{\mathbf{x}}$ be the sparse representation of the interpolated image $\widehat{\mathbf{I}}_j$ in basis $\mathbf{\Psi}$, $\boldsymbol{\psi}_i$ the i-th row of $\mathbf{\Psi}$, and $\mathbf{R}_i = \boldsymbol{\psi}_i^T \boldsymbol{\psi}_i$, which is a symmetric matrix. Each squared error in the last term of (2.4) can be written as

$$(I_{i,j} - \widehat{I}_{i,j})^2 = (\mathbf{x} - \widehat{\mathbf{x}})^T \mathbf{R}_i (\mathbf{x} - \widehat{\mathbf{x}}) \ . \tag{2.5}$$

As in (2.1), we split $\mathbf{x}$ and $\widehat{\mathbf{x}}$ into their positive and negative parts. The generalized framework in (2.4) can thus be converted to the standard BCQP format in (2.2), with the following definitions:

$$\begin{aligned}
\mathbf{z} =& \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \ \ \mathbf{b} = \mathbf{A}^T \mathbf{y} + \mu \sum_{i=1}^{N} w_i \mathbf{R}_i (\widehat{\mathbf{u}} - \widehat{\mathbf{v}}), \\
\mathbf{c} =& \tau \mathbf{1}_{2N} + \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}, \\
\mathbf{B} =& \begin{bmatrix} \mathbf{A}^T\mathbf{A} + \mu \sum\limits_{i=1}^{N} w_i \mathbf{R}_i & -(\mathbf{A}^T\mathbf{A} + \mu \sum\limits_{i=1}^{N} w_i \mathbf{R}_i) \\ -(\mathbf{A}^T\mathbf{A} + \mu \sum\limits_{i=1}^{N} w_i \mathbf{R}_i) & \mathbf{A}^T\mathbf{A} + \mu \sum\limits_{i=1}^{N} w_i \mathbf{R}_i \end{bmatrix}.
\end{aligned} \tag{2.6}$$

The GPSR algorithm can then be used to solve this BCQP problem.

2.3 Simulation Results

In this section, we present some simulation results to compare our proposed algorithm with other GPSR-based algorithms. The orthonormal basis in the CS is chosen as the DCT. In the image acquisition step of compressed sensing, the scrambled block Hadamard ensemble (SBHE) method proposed in [37] is used. The size B and free parameter τ are chosen according to [37]. The view interpolation method in [106] is used.

In the following experiments, View-Interp represents the view interpolation result given by the method in [106], which will be included in the figures as a reference. Direct-GPSR refers to a method similar to the scheme in [56], where the view interpolation result $\widehat{\mathbf{x}}$ is directly used as the initial value of GPSR reconstruction. Diff-GPSR is the generalization of [26] to multiview image systems, where the GPSR method is used to recover the residual frame, which is then added back to the interpolated image to get the final reconstruction.

VIC-GPSR denotes the proposed view interpolation confidence aided GPSR method. Its initial value is also chosen as $\widehat{\mathbf{x}}$. The frame-level weighting parameter μ is selected for each multiview image data set, as will be described below. The pixel-level weighting parameter w_i is chosen to be 1, 0, and 0 respectively, if a target pixel has two, one or zero point correspondence in view interpolation. That is, in the interpolated image, we only trust the pixels with two point correspondences when evaluating the CS reconstruction.

FVIC-GPSR is another special case of the proposed method, where all w_i's are fixed to be 1. In this case, the pixel-level confidence information is not exploited, and only the frame-level weighting parameter μ is in effect.

In the following, the multiview video datasets Akko & Kayo, Christmas and Teddy are used, with frame size of 640×480, 640×480, and 448×352, respectively. The first and third views of each dataset are assumed to be given by traditional cameras, and the second view is assumed to be sampled by a CS camera and reconstructed by different CS algorithms. Only the first frame of each view is tested.

Fig. 2.1 (a) shows the reconstruction PSNR versus CS sampling subrate M/N of different methods with the multiview image dataset Akko & Kayo. The view interpolation result shows that the number of target pixels with two, one and zero point correspondences is 285879, 20317, and 1004, respectively. The weight parameter μ is chosen to be 1.

Some observations can be made from Fig. 2.1 (a). First, at low subrate, *Direct-GPSR* is much worse than other methods. As the number of samples M increases, Direct-GPSR can get close to and eventually outperform other methods, including our proposed VIC-GPSR. The reason is that the parameter μ is fixed to 1, which essentially gives the same weight to the first and the last term in Eq. (2.4).

Secondly, the proposed VIC-GPSR and *FVIC-GPSR*, as well as *Diff-GPSR* can always have better results than the interpolated view. Our methods also always achieve better results than *Diff-GPSR*, and the gain increases with the subrate (more than 3 dB when the subrate is greater than 0.3), which shows the power of considering the view interpolation confidence information.

Third, VIC-GPSR has better performance than FVIC-GPSR, thanks to the contribution of the pixel-level confidence information. The gain also increases with the subrate.

Fig. 2.1 (b) are the results using the dataset Christmas. The view interpolation result shows that the number of target pixels with two, one and zero point correspondences is 272428, 31029, and 3743, respectively. This means that the view interpolation of this dataset is not as good as that in Akko & Kayo, as indicated by the PSNRs of the view interpolation method in Fig. 2.1 (a) and Fig. 2.1 (b). In our methods, μ is set to be 1.

Fig. 2.1 (b) shows that *Direct-GPSR* is worse than View-Interp when subrate is less than about 0.33. This verifies that if the view interpolation does not have good quality, directly using it as the initial value of GPSR could lead to even worse result than the interpolated view. Our methods and *Diff-GPSR* can still outperform the view interpolation. Note that

Diff-GPSR only has limited gain over the view interpolation method, and the gain of our methods over *Diff-GPSR* can be more than 4 dB when the subrate is greater than 0.3.

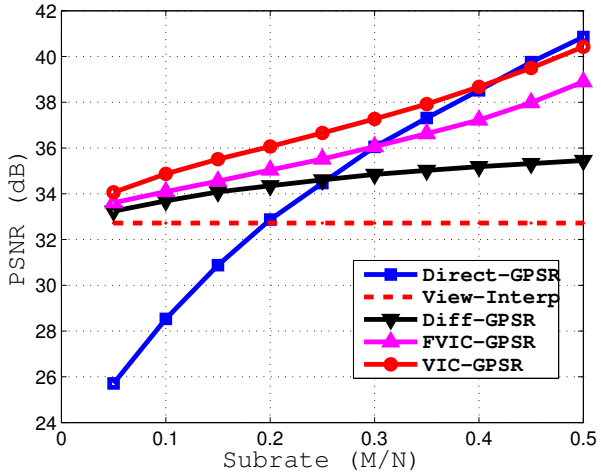
Fig. 2.1 (c) shows the results with the Teddy dataset. The number of target pixels with two, one and zero point correspondences is 141946, 15289, and 461, respectively. Therefore, we choose the weighting parameter μ to be 5.

It can be seen from Fig. 2.1 (c) that Diff-GPSR achieves almost the same result as View-Interp, sometimes even worse when the subrate is low, because the subrate is not enough to recover the difference image accurately, and fails to capture the edges in the middle image. Our proposed algorithm always achieves the best performance.

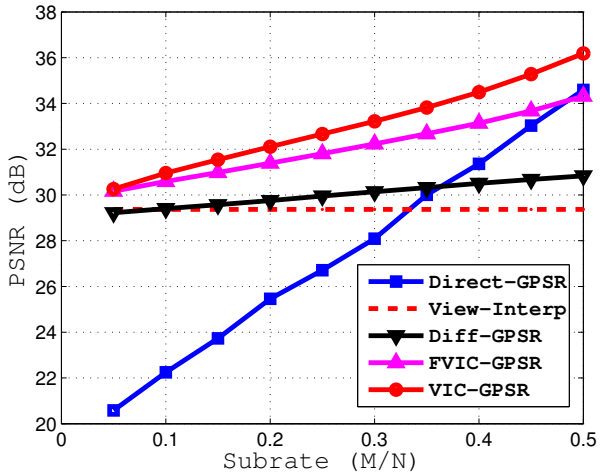
Fig. 2.2 shows portions of the final reconstruction errors using different CS methods. The proposed method has much smaller errors near the edges in the images. This verifies that our method can avoid the adverse impact of the occlusion and holes in view interpolation.

2.4 Summary

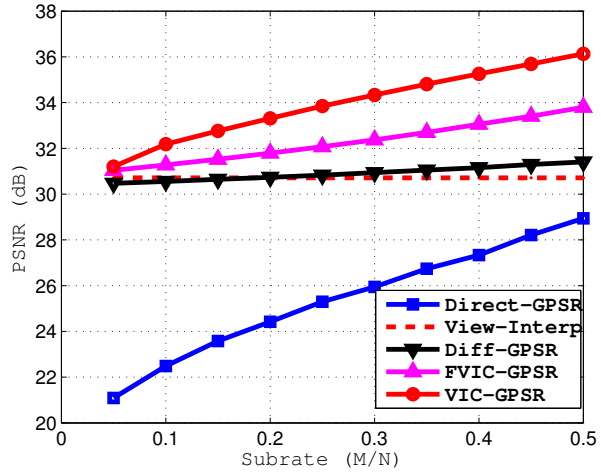
In this part, we consider view interpolation-aided compressed sensing of multiview images. Different from existing methods, we exploit the knowledge of occlusions and holes in the interpolated view when performing the CS reconstruction, by assigning more weights to the view interpolation result when its quality is satisfactory, and vice versa. Experimental results show that our method outperforms existing CS-based multiview image systems.



(a)

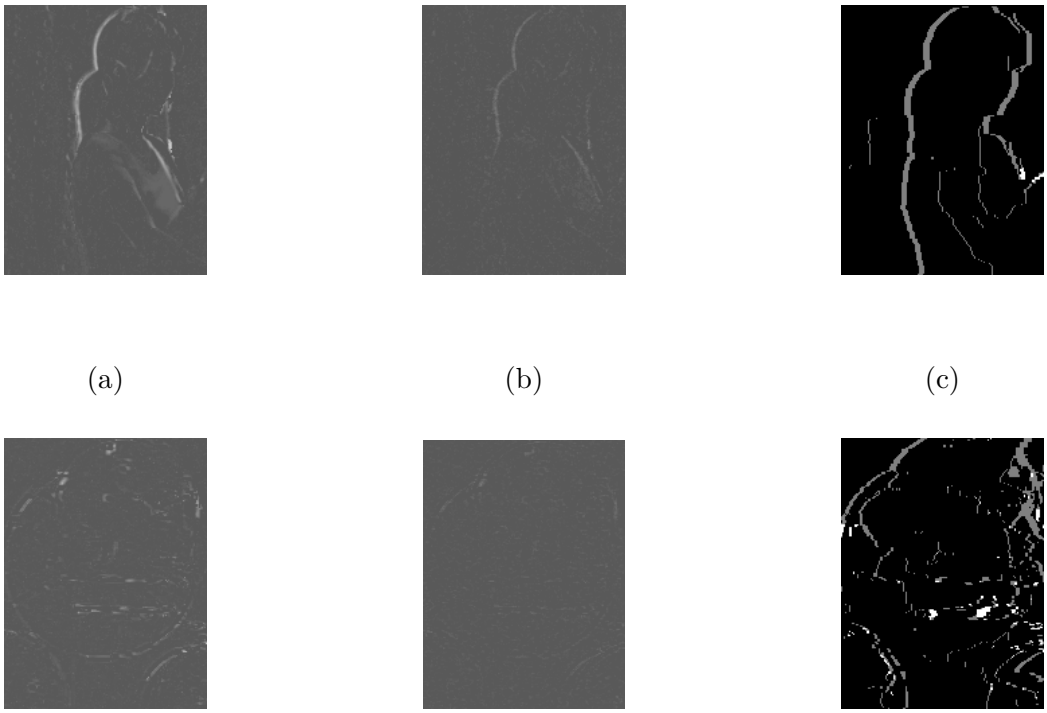


(b)



(c)

Figure 2.1: PSNRs versus sampling subrate of different methods. (a) Akko & Kayo. (b) Christmas. (c)Teddy.



(a) (b) (c)

(d) (e) (f)

Figure 2.2: Portions of the reconstruction errors of Akko & Kayo (a), (b) and Christmas (d), (e), with subrate=0.4 using different methods. (a), (d) Diff-GPSR. (b), (e) VIC-GPSR and their corresponding confidence map, (c) Akko & Kayo, (f) Christmas.

Chapter 3

Approximate Message Passing-based Compressed Sensing Reconstruction with Generalized Elastic Net Prior

In Chapter 2, we discussed the view interpolation confidence-aided compressed sensing problem. There are some questions remain to be answered. First, the regularization parameter τ in Eq. (2.4) is strongly correlated to the quality of interpolated middle image and plays an important role in the final reconstruction performance. In Chapter 2, we set τ empirically for the test images. It is unclear how to adjust τ adaptively based on the quality of view interpolation. Next, there is no theoretical analysis to prove the image reconstructed by solving the optimization problem in Eq. (2.4) has better quality than the one for Eq. (1.4). This chapter is devoted to answer these questions and generalize the conclusions made in Chapter 2.

In this chapter, we study the compressed sensing reconstruction problem with generalized elastic net prior (GENP), where a sparse signal is sampled via a noisy underdetermined linear observation system, and an additional initial estimation of the signal (the GENP) is available during the reconstruction. We first incorporate the GENP into the LASSO and the approximate message passing (AMP) frameworks, denoted by GENP-LASSO and GENP-AMP respectively. We then investigate the parameter selection, state evolution, and noise-sensitivity analysis of GENP-AMP. We show that, thanks to the GENP, there is no phase transition boundary in the proposed frameworks, *i.e.*, the reconstruction error is bounded in the entire plane. The error is also smaller than those of the standard AMP and scalar denoising. A practical parameterless version of the GENP-AMP is also developed, which does not need to know the sparsity of the unknown signal and the variance of the GENP. Simulation results are presented to verify the efficiency of the proposed schemes.

Throughout this chapter, we model the initial estimation or SI of the signal as a noisy version of the unknown sparse signal, and modify the LASSO and AMP frameworks to incorporate the initial estimation. After developing the frameworks of GENP-LASSO and GENP-AMP, we focus on the GENP-AMP, and investigate its parameter selection, state evolution, asymptotic prediction performance and noise-sensitivity analysis. We show that there is no phase transition boundary in our scheme, *i.e.*, the mean-squared error (MSE) of the reconstruction is bounded in the entire plane, thanks to the generalized elastic net prior. As far as the authors' knowledge, this is the first result that demonstrates that a CS scheme could have such a property. Moreover, the MSE of GENP-AMP is smaller than those of the standard AMP and scalar denoising.

The theoretical analyses require the knowledge of the sparsity of the unknown sparse signal and the variance of the generalized elastic net prior. In practices, these parameters have to be estimated. In [73], a parameterless AMP is developed using Stein's unbiased risk estimate (SURE). Inspired by [73], we apply the SURE theory to GENP-AMP and develop a parameterless version of GENP-AMP.

The rest of this chapter is organized as follows. Sec. 4.1 reviews the necessary background of minimax MSE of soft thresholding algorithm. Sec. 3.2 formulates the GENP-LASSO problem. Sec. 3.3 formulates GENP-AMP, studies its connection with GENP-LASSO, and presents its parameter selection and state evolution. In Sec. 3.4, we derive the noise sensitivity analysis of the GENP-AMP. The parameterless GENP-AMP is developed in Sec. 3.5. Simulation results with both 1-D data and multiview images are presented in Sec. 3.6, and the proofs of some main results are given in the Appendix.

3.1 Background: Minimax MSE of Soft Thresholding Algorithm

In this section, we briefly review the minimax MSE of the soft thresholding algorithm [29,32], which plays an important role in AMP. Suppose we need to recover a k-sparse n-vector $\mathbf{x}^0 = (x^0(i) : i \in [n])$ (where $[n] \equiv \{1, \ldots, n\}$) contaminated by a Gaussian white noise, *i.e.*,

$$y(i) = x^0(i) + z^0(i),\ i \in [n],$$

where $z^0(i) \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed. One way to estimate the signal is to solve the following LASSO or ℓ_1-regularized least-square problem,

$$\hat{\mathbf{x}}^\lambda = \arg\min_{\mathbf{x}}\ \frac{1}{2}\left\|\mathbf{y} - \mathbf{x}\right\|_2^2 + \lambda\|\mathbf{x}\|_1. \tag{3.1}$$

An important fact is that the solution of this problem is equivalent to that of the well-known soft thresholding algorithm in wavelet denoising [29],

$$\hat{x}^{\lambda}(i) = \eta(y(i); \lambda),\ i \in [n],$$

where the soft thresholding operation with threshold θ is

$$\eta(x;\theta) = \begin{cases} x-\theta & \text{if } x > \theta, \\ 0 & \text{if } -\theta \leqslant x \leqslant \theta, \\ x+\theta & \text{if } x < -\theta. \end{cases} \tag{3.2}$$

A reasonable choice of the threshold λ in (3.1) is a scaled version of the noise standard deviation, *i.e.*, $\lambda = \alpha\sigma$. The MSE of the soft thresholding algorithm can thus be written as

$$\mathrm{mse}(\sigma^2; p, \alpha) \equiv \mathbb{E}\{[\eta(X + \sigma Z; \alpha\sigma) - X]^2\}, \tag{3.3}$$

where the expectation is with respect to independent random variables $Z \sim \mathcal{N}(0,1)$ and $X \sim p$.

The soft thresholding method is scale-invariant [32], *i.e.*,

$$\mathrm{mse}(\sigma^2; p, \alpha) = \sigma^2 \mathrm{mse}(1; p_{1/\sigma}, \alpha), \tag{3.4}$$

where p_s is a scaled version of p, $p_s(S) = p(\{x :\ sx \in S\})$. Therefore we only need to focus on $\sigma = 1$, and the notation $\mathrm{mse}(1; p, \alpha)$ can be simplified into $\mathrm{mse}(p, \alpha)$.

Since x^0 is k-sparse, we can define the following set of probability measures with small non-zero probability,

$$\mathcal{F}_{\varepsilon} \equiv \{p :\ p \text{ is a probability measure with } p(\{0\}) \geqslant 1-\varepsilon\}, \tag{3.5}$$

where $\varepsilon = k/n$ is defined in (1.2).

The minimax threshold MSE is thus defined as [32]

$$M^{\pm}(\varepsilon) = \inf_{\alpha > 0} \sup_{p \in \mathcal{F}_{\varepsilon}} \mathrm{mse}(p, \alpha), \tag{3.6}$$

which is the minimal MSE of the worst distribution in $\mathcal{F}_{\varepsilon}$, where $\pm$ means a nonzero estimand can take either sign.

For a given α, the worst case MSE in (3.6) is given by [32]

$$\sup_{p \in F_{\varepsilon}} \mathrm{mse}(p, \alpha) = \varepsilon(1+\alpha^2) + (1-\varepsilon)[2(1+\alpha^2)\Phi(-\alpha) - 2\alpha\phi(\alpha)], \tag{3.7}$$

with $\phi(z) = \exp(-z^2/2)/\sqrt{2\pi}$ being the standard normal density, and $\Phi(z) = \int_{-\infty}^{z} \phi(x)dx$ the Gaussian cumulative distribution function. Moreover, the supremum can be achieved by the following three-point probability distribution on the extended real line $\mathcal{R} \cup \{-\infty, \infty\}$

$$p_\varepsilon^* = (1-\varepsilon)\delta_0 + \frac{\varepsilon}{2}\delta_\infty + \frac{\varepsilon}{2}\delta_{-\infty},$$

where δ_t is a Dirac delta function at t. In practice, we are more interested in the near-worse-case signals with finite values. It is known that the following c-least-favorable distribution can achieve a MSE that is a fraction of $(1-c)$ of the worst case,

$$p_{\varepsilon,c} = (1-\varepsilon)\delta_0 + \frac{\varepsilon}{2}\delta_{h^{\pm}(\varepsilon,c)} + \frac{\varepsilon}{2}\delta_{-h^{\pm}(\varepsilon,c)}, \tag{3.8}$$

where $h^{\pm}(\varepsilon, c) \sim \sqrt{2\log(\varepsilon^{-1})}$ as $\varepsilon \to 0$.

3.2 GENP-aided LASSO

In this chapter, we study the generalized elastic net prior (GENP)-aided CS reconstruction, where in addition to the CS sampling as in (1.1), an initial estimation of $\mathbf{x}$, denoted by $\tilde{\mathbf{x}}$, is available during reconstruction, which can be seen as a noisy version of $\mathbf{x}$. The error of this estimation, $\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x}$, is assumed to be i.i.d. additive white Gaussian with variance σ_s^2, *i.e.*, $e \sim \mathcal{N}(\mathbf{0}, \sigma_s^2\mathbf{I})$. This Gaussian noise model is decently accurate in applications such as image acquisition with poor illumination, high temperature, or transmission error, and has been widely used in image denoising [22]. The ratio between the noise variance of the GENP and that of the compressed sampling noise in Eq. (1.1) will be used later for noise sensitivity analysis.

$$\gamma_s^2 = \sigma_s^2/\sigma^2. \tag{3.9}$$

In this section, we formulate the GENP-aided reconstruction from estimation theory, in particular, the maximum a posteriori (MAP) criterion, and develop an GENP-LASSO framework. In Sec. 3.3, based on the results in this section, a fast GENP-aided AMP algorithm is developed to reduce the complexity of recovering the signal.

By the Bayesian rule, the posterior probability is proportional to

$$p(\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}) \;\propto\; p(\mathbf{x})p(\mathbf{y}, \tilde{\mathbf{x}}|\mathbf{x}) \overset{(a)}{=} p(\mathbf{x})p(\mathbf{y}|\mathbf{x})p(\tilde{\mathbf{x}}|\mathbf{x}), \tag{3.10}$$

where (a) is due to the conditional independence of $\tilde{\mathbf{x}}$ and $\mathbf{y}$ given $\mathbf{x}$. The simplest choice for the prior $p(\mathbf{x})$ is a product of identical factors $p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$, which can be easily generalized, *e.g.*, a non-uniformly sparsity model is considered in [87], where different coefficients have different nonzero probabilities. Based on the assumption that $p(\tilde{\mathbf{x}}|\mathbf{x})$ is the

product of identical factors, *i.e.*, $p(\tilde{\mathbf{x}}|\mathbf{x}) = \prod_{i=1}^{n} p(\tilde{x}_i|x_i)$, the posterior pdf can be written as

$$p_{\sigma,\sigma_s}(\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}}) = \frac{\exp(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2)\prod_{i=1}^{n} p(x_i)p(\tilde{x}_i|x_i)}{Z(\mathbf{y},\tilde{\mathbf{x}})}, \tag{3.11}$$

where $Z(\mathbf{y},\tilde{\mathbf{x}})$ is the normalization constant. We call $\prod_{i=1}^{n} p(x_i)p(\tilde{x}_i|x_i)$ the *joint prior*, which includes contributions from both the source and the initial estimation. By the MAP criterion, we have

$$\hat{\mathbf{x}} = \underset{\mathbf{z}\in\mathcal{R}^n}{\arg\min} \left(\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2^2 + \sum_{i=1}^{n} g(z_i) + \sum_{i=1}^{n} f(\tilde{x}_i, z_i)\right). \tag{3.12}$$

When g is convex, (3.12) can be easily solved. In particular, if $g(z_i) = \lambda|z_i|$, and $f(\tilde{x}_i, z_i) = \frac{\tau_s}{2}(\tilde{x}_i - z_i)^2$, we have

$$\begin{aligned}\hat{\mathbf{x}}(\lambda,\tau_s) = \underset{z\in\mathcal{R}^n}{\arg\min} \Bigg(&\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2^2 \\ &+\lambda\|\mathbf{z}\|_1 + \frac{\tau_s}{2}\|\tilde{\mathbf{x}}-\mathbf{z}\|_2^2\Bigg),\end{aligned} \tag{3.13}$$

which is a generalized version of the LASSO in (3.1) with an additional ℓ_2 penalty term caused by the initial estimation $\tilde{\mathbf{x}}$. When $\tilde{\mathbf{x}} = 0$, the problem reduces to the elastic net-regularized LASSO in [110]. Therefore we call $\tilde{\mathbf{x}}$ generalized elastic net prior (GENP), and the problem in Eq. (3.13) generalized elastic net prior-aided LASSO (GENP-LASSO).

In LASSO, the ratio ρ in Eq. (1.2) cannot be larger than 1, *i.e.*, the number of selected atoms is bounded by the number of samples, whereas it is shown in [110] that in the elastic net-regularized LASSO, the quadratic penalty term removes this limitation. Our noise sensitivity analysis in Sec. 3.4 will show that $\rho < 1$ is also not necessary in the GENP-LASSO.

The parameters λ and τ_s in Eq. (3.13) are closely related to σ_s^2, the noise variance of the GENP. How to tune the two parameters λ and τ_s will be addressed later in this chapter. The proposed GENP-LASSO in (3.13) is a convex optimization problem and can be solved by, *e.g.*, the interior point methods (as used in the CVX package [44]) and the gradient methods. For example, to incorporate the GENP into the Orthant-Wise Limited-memory Quasi-Newton (OWLQN) algorithm [7], which is a popular gradient-based method for large-scale LASSO problems, we can replace the ℓ_2 regularization term $\|\mathbf{z}\|_2^2$ in it by the quadratic penalty term $\|\tilde{\mathbf{x}}-\mathbf{z}\|_2^2$. However, both interior point and gradient methods are quite slow for large-scale problems.

In this chapter, we solve the GENP-LASSO problem by modifying the fast AMP algorithm, which enjoys several advantages, *e.g.*, low complexity and the capability of predicting the final performance accurately.

3.3 GENP-aided Approximate Message Passing

In this section, we present the formulae of GENP-AMP, study its connections with the GENP-LASSO, and derive its corresponding parameter selections and state evolution.

3.3.1 The Formula of GENP-AMP

In [72], the following iterative formulas of AMP are obtained after simplifying the traditional min-sum-based message passing algorithm using the quadratic approximation.

$$\begin{aligned} \hat{\mathbf{x}}_0^t &= \mathbf{x}^t + \mathbf{A}^T\mathbf{r}^t, \\ \mathbf{x}^{t+1} &= \eta(\hat{\mathbf{x}}_0^t; \theta_t), \end{aligned} \tag{3.14}$$

$$b_t = \frac{1}{m}\left\|\mathbf{x}^t\right\|_0, \tag{3.15}$$

$$\mathbf{r}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + b_t\mathbf{r}^{t-1}. \tag{3.16}$$

Each iteration of AMP only needs to update the estimate $\mathbf{x}^t$ in (3.14) and the residual $\mathbf{r}^t$ in (3.16), which have only $m+n$ entries. The complexity is thus much lower than traditional message passing methods that need $2mn$ updates. Note that the AMP is parameterized by two sequences of scalar parameters: the thresholds $\{\theta_t\}_{t\geq 0}$ and the forgetting factors $\{b_t\}_{t\geq 0}$.

To incorporate the GENP into AMP, we modify the local message of each AMP variable node from $\lambda\|\mathbf{z}\|_1$ to $\lambda\|\mathbf{z}\|_1 + \frac{\tau_s}{2}\|\tilde{\mathbf{x}} - \mathbf{z}\|_2^2$. By the same simplifications and derivations in [72], we can get the following iterative estimate of the n-vector signal x. The details are skipped due to space limitation.

$$\hat{\mathbf{x}}_0^t = \frac{u_t}{1+u_t}\tilde{\mathbf{x}} + \frac{1}{1+u_t}(\mathbf{x}^t + \mathbf{A}^T\mathbf{r}^t), \tag{3.17}$$

$$\mathbf{x}^{t+1} = \eta(\hat{\mathbf{x}}_0^t;\ \theta_t), \tag{3.18}$$

$$b_t = \frac{1}{1+u_{t-1}}\frac{\|\mathbf{x}^t\|_0}{m}, \tag{3.19}$$

$$\mathbf{r}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + b_t\mathbf{r}^{t-1}. \tag{3.20}$$

Compared to AMP, $\hat{\mathbf{x}}_0^t$ in our scheme is a linear combination of $\mathbf{x}^t + \mathbf{A}^T\mathbf{r}^t$ and the GENP, adaptively controlled by a new sequence of scalar parameters, $\{u_t\}_{t\geq 0}$. The forgetting factor b_t is also affected by u_{t-1}. When $u_t = 0$, $\tilde{\mathbf{x}}$ has no contribution, and the proposed

framework reduces to the standard AMP in [31, 32, 68, 72]. The iteration is applied to each entry. Hence, if the variances of different $\tilde{x}_i$ are different, the method can still be applied by changing the scalar u_t to vector $\mathbf{u_t} = [u_{t,1}, u_{t,2}, ..., u_{t,n}]$ and the scalar θ_t to its vector case.

3.3.2 Connections to GENP-LASSO

As shown in [72], the parameters $\{\theta_t\}_{t\geqslant 0}$ and $\{b_t\}_{t\geqslant 0}$ are constrained by its connection with the min-sum algorithm. This is also true for the new parameter $\{u_t\}_{t\geqslant 0}$. However, the following proposition shows that GENP-AMP provides a very general solution for the GENP-LASSO problem in Eq. (3.13). When there is no GENP ($u_t = 0$), the proposition reduces to Prop. 5.1 in [72] for LASSO.

Proposition 3.3.1. *Let $(\mathbf{x}^*, \mathbf{r}^*)$ be the fixed point of the GENP-AMP algorithm given by (3.17) and (3.20) for fixed $\theta_t = \theta$, $u_t = u$, and $b_t = b$. Then $\mathbf{x}^*$ is also a minimum of the GENP-LASSO problem in (3.13) with*

$$\lambda = (1+u)\theta(1-b), \tag{3.21}$$

$$\tau_s = u(1-b). \tag{3.22}$$

Proof. The fixed-point condition of Eq. (3.17) is

$$\mathbf{x}^* = \frac{u}{1+u}\tilde{\mathbf{x}} + \frac{1}{1+u}(\mathbf{x}^* + \mathbf{A}^T\mathbf{r}^*) - \theta\mathbf{v}^*, \tag{3.23}$$

where $v_i^* = sign(x_i^*)$ if $x_i^* \neq 0$ and $v_i^* \in [-1, +1]$ otherwise. Similarly, from (3.20), we get $(1-b)\mathbf{r}^* = \mathbf{y} - \mathbf{A}\mathbf{x}^*$, or $\mathbf{r}^* = (\mathbf{y} - \mathbf{A}\mathbf{x}^*)/(1-b)$. Plugging into the equation above, we get

$$(1+u)\theta(1-b)\mathbf{v}^* + u(1-b)(\mathbf{x}^* - \tilde{\mathbf{x}}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^*).$$

On the other hand, in Eq. (3.13), by setting the derivative of the GENP-LASSO objective function with respect to z to zero, we get the stationary condition

$$\lambda\mathbf{v}^* + \tau_s(\mathbf{x}^* - \tilde{\mathbf{x}}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^*). \tag{3.24}$$

Comparing the two equations above leads to the conclusion. □

3.3.3 GENP-AMP State Evolution and Parameter Selection

In this part, we derive the state evolution of GENP-AMP and investigate its parameter selection. The state evolution was first developed to describe the asymptotic limit of the AMP estimates as $m, n \to \infty$ for any fixed t, but with the same sample ratio $\delta = m/n$, as

defined in (1.2) [72]. It enables the accurate prediction of the MSE of AMP by solving a fixed-point equation. This part is based on Sec. IV of [32].

First, we define the MSE map Ψ as

$$\Psi(q^2, u, \delta, \sigma, \sigma_s, \alpha, p) \equiv \mathrm{mse}(\mathrm{npi}(q^2, u; \delta, \sigma, \sigma_s); p, \alpha),$$

which is the MSE of the soft thresholding as defined in (3.3) with npi (noise-plus interference) as the noise variance, where q^2 is the variance of the thresholded estimator, and npi is the variance of the un-thresholded estimator in (3.17), which can be written as (see Appendix A for the derivation)

$$\mathrm{npi}(q^2, u; \delta, \sigma, \sigma_s) = (\frac{u}{1+u})^2\sigma_s^2 + (\frac{1}{1+u})^2(\sigma^2 + \frac{q^2}{\delta}). \tag{3.25}$$

As pointed out in [72], the choice of the AMP parameter θ_t can be quite flexible. A good option is $\theta_t = \alpha\xi_t$, where $\alpha > 0$, and ξ_t is the root MSE of the un-thresholded estimation $\hat{\mathbf{x}}_0^t$ in (3.17). From this, based on the i.i.d. normalized distribution of $\mathbf{A}$ and the large system limit [32], it can be shown that

$$\xi_t^2 = \mathrm{npi}(q_t^2, u_t^2; \delta, \sigma, \sigma_s) \approx \left(\frac{u_t}{1+u_t}\right)^2\sigma_s^2 + \left(\frac{1}{1+u_t}\right)^2\frac{\left\|\mathbf{r}^t\right\|_2^2}{m}. \tag{3.26}$$

Besides, we have $\left\|\mathbf{x}^t\right\|_0/n \approx \mathbb{E}\{\eta'(x_0 + \sigma^t Z; \alpha\sigma^t)\}$. According to Eq. (3.19, 3.21, 3.22), Prop. 3.3.1 can be rewritten as

$$\begin{aligned}
\lambda &= (1+u_*)\alpha\xi_* \left[1 - \frac{1}{1+u_*}\frac{\mathbb{E}\{\eta'(x_0 + \xi_* Z; \alpha\xi_*)\}}{\delta}\right], \\
\tau_s &= u_* \left[1 - \frac{1}{1+u_*}\frac{\mathbb{E}\{\eta'(x_0 + \xi_* Z; \alpha\xi_*)\}}{\delta}\right],
\end{aligned} \tag{3.27}$$

where $\xi_* = \lim_{t\to\infty}\xi_t$. Since the computation of q^2 is nontrivial, Eq. (3.26) is useful for practical algorithm design, whereas Eq. (3.25) is mainly for theoretical analysis.

The state of GENP-AMP is defined as a 7-tuple $(q^2, u; \delta, \sigma, \sigma_s, \alpha, p)$. The state evolution follows the rule

$$\begin{aligned}
(q_t^2, u_t; \delta, \sigma, \sigma_s, \alpha, p) &\mapsto (\Psi(q_t^2, u_t), \Upsilon(q_t^2, u_t); \delta, \sigma, \sigma_s, \alpha, p), \\
t &\mapsto t+1,
\end{aligned}$$

where q_t^2 and u_t are the MSE and the weighting parameter in the t-th iteration, and Ψ and Υ are the evolution functions of q_t^2 and u_t, respectively. As $(\delta, \sigma, \sigma_s, \alpha, \upsilon)$ are fixed during the evolution, we only need the following state evolutions of q_t^2 and u_t (See Appendix A.1

for the derivation).

$$\begin{aligned} q_t^2 \mapsto q_{t+1}^2 &\equiv \Psi(q_t^2, \frac{\sigma^2 + q_t^2/\delta}{\sigma_s^2}), \\ u_t \mapsto u_{t+1} &= \Upsilon(q_t^2, u_t) = \frac{\sigma^2 + \Psi(q_t^2, (\sigma^2 + q_t^2/\delta)/\sigma_s^2)/\delta}{\sigma_s^2}, \end{aligned} \tag{3.28}$$

where the formula for u_t is the result of the following proposition.

Proposition 3.3.2. *The optimal weighting parameter u_t that combines the GENP $\tilde{x}$ and the previous iteration result in the GENP-AMP is given by*

$$u_t = \frac{\sigma^2 + q_t^2/\delta}{\sigma_s^2}. \tag{3.29}$$

Proof. The optimal u_t should minimize the MSE between the original sparse signal and the un-thresholded estimation $\hat{x}_0^t$ in (3.17), which can be obtained by minimizing $(\frac{u_t}{1+u_t})^2\sigma_s^2 + (\frac{1}{1+u_t})^2(\sigma^2 + \frac{q_t^2}{\delta})$ over u_t. □

Replacing u in Eq. (3.25) by Eq. (3.29), $\mathrm{npi}(q^2, u; \delta, \sigma, \sigma_s)$ can be simplified into

$$\mathrm{npi}(q^2) = \frac{\sigma_s^2(\sigma^2 + q^2/\delta)}{\sigma_s^2 + \sigma^2 + q^2/\delta}. \tag{3.30}$$

The fixed point condition of the state evolution is

$$q_*^2 = \Psi(q_*^2, \frac{\sigma^2 + q_*^2/\delta}{\sigma_s^2}) = \mathrm{mse}(\mathrm{npi}(q_*^2); p, \alpha). \tag{3.31}$$

If we treat $\xi^2 = \mathrm{npi}(q_*^2)$ as an unknown variable, plugging (3.31) into (3.30) yields a fixed-point equation for ξ^2,

$$\xi^2 = \frac{\sigma_s^2(\sigma^2 + \mathrm{mse}(\xi^2; p, \alpha)/\delta)}{\sigma_s^2 + \sigma^2 + \mathrm{mse}(\xi^2; p, \alpha)/\delta} \equiv F(\xi^2, \alpha). \tag{3.32}$$

The following result shows that with an appropriate choice of α, the fixed-point equation has a unique solution, from which we can predict the final MSE performance of the GENP-AMP algorithm.

Proposition 3.3.3. *Let $\alpha_{\min} = \alpha_{\min}(\delta, \gamma_s)$ be the unique non-negative solution of the equation*

$$(1 + \alpha^2)\Phi(-\alpha) - \alpha\phi(\alpha) = \frac{\delta}{2}\frac{(\gamma_s^2 + 1)^2}{\gamma_s^4}, \tag{3.33}$$

where $\phi(z)$ and $\Phi(z)$ are defined after Eq. (3.7), and γ_s^2 is defined in Eq. (3.9). Then for any $\alpha > \alpha_{\min}(\delta, \gamma_s)$, the fixed-point equation $\xi^2 = F(\xi^2, \alpha)$ in (3.32) admits a unique solution $\xi_* = \xi_*(\alpha)$, and $\lim_{t\to\infty}\xi_t = \xi_*(\alpha)$.

Proof. This proof is an extension of Case $\chi = \pm$ in Appendix C of [31]. It is easy to find that if γ_s^2 goes to ∞, the whole equation is exactly the one in [72].

Since we want to have $F < \xi^2$, following the same setup as the one in Case $\chi = \pm$ in Appendix C of [31], we need to consider the boundary point, which can be found by solving the boundary condition $\frac{dF}{d\xi^2}|_{\xi^2=0} = 1$. This leads to $\frac{\sigma_s^4 d(\Psi/\delta)/d\xi^2}{(\sigma_s^2+\sigma^2+\Psi/\delta)^2}|_{\xi^2=0} = 1$. If $\xi^2 \to 0$, we know that $q^2/\delta = 0$, and the expression of $\frac{d(q^2/\delta)}{d\xi^2}$ can be obtained as in [31]. Then the problem is transformed into

$$\frac{d(q^2/\delta)}{d\xi^2}|_{\xi^2=0} = \frac{\left(1+\gamma_s^2\right)^2}{\gamma_s^4}. \tag{3.34}$$

The numerator of Eq. (3.34) becomes $\frac{\left(1+\gamma_s^2\right)^2}{\gamma_s^4}(1 - \frac{\gamma_s^4}{(1+\gamma_s^2)^2}\frac{2}{\delta}[(1+\alpha^2)\Phi(-\alpha) - \alpha\phi(\alpha)])$ instead of $1 - \frac{2}{\delta}[(1+\alpha^2)\Phi(-\alpha) - \alpha\phi(\alpha)]$ as in the classical case in Eq. (6.6) of [72]. Comparing these two expressions, from Proposition 6.2 in [72], we can reach the conclusion. □

If the threshold α and the distribution p_0 of X_0 are given, we can obtain the fixed point ξ_* by solving Eq. (3.32). Therefore, the MSE performance of the GENP-AMP algorithm can be predicted.

Based on Prop. 3.3.1, λ and τ_s can be determined if the necessary parameters are known. Conversely, if either λ or τ_s is given, combining Eq. (3.33) with Eq. (3.27), we can get the corresponding α and ξ_*. Thus the other parameter can be uniquely determined.

3.4 Noise Sensitivity Analysis of GENP-AMP

The noise sensitivity phase transition is a curve in the (ρ, δ) plane [32], where $\rho = k/m$ and $\delta = m/n$, as defined in (1.2). For many classical compressed sensing algorithms, the MSE is bounded below the phase transition curve, and unbounded above the curve. It is known that ℓ_1-based methods (such as the CVX package [44]) enjoys the best phase transition performance, and the fast AMP can achieve the same phase transition performance [32]. For large-scale problems, the OWLQN algorithm in [7] has similar empirical phase transition boundary to ℓ_1 methods, but its complexity is higher.

In this section, we show that there is no phase transition boundary for GENP-AMP, *i.e.*, its MSE is bounded in the entire plane, thanks to the GENP. We also prove that $\rho < 1$ is no longer needed, which agrees with Lemma 1 in [110] for the elastic net-regularized LASSO.

First, for the GENP-LASSO problem in (3.13), we define the MSE per entry when the empirical distribution of the signal converges to p_0:

$$\mathrm{MSE}(\sigma^2; \sigma_s^2, p_0, \lambda, \tau_s) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}\{\|\hat{\mathbf{x}}(\lambda, \tau_s) - \mathbf{x}_0\|_2^2\}, \tag{3.35}$$

where the limit is taken along a converging sequence. Since the class $\mathcal{F}_\epsilon$ in (3.5) is scale-invariant, where $\varepsilon = k/n = \rho\delta$ according to (1.2), the minimax risk of the GENP-LASSO can be written as

$$\inf_{\lambda,\tau_s} \sup_{p_0 \in F_{\rho\delta}} \mathrm{MSE}(\sigma^2; \sigma_s^2, p_0, \lambda, \tau_s) = M^*(\delta, \rho, \gamma_s^2)\sigma^2, \tag{3.36}$$

which indicates the sensitivity of the GENP-LASSO to the noise variance in the measurements, where γ_s^2 is defined in Eq. (3.9), and the expression of noise sensitivity $M^*(\delta, \rho, \gamma_s^2)$ is given by the following proposition. We also give closed-form expressions of the tuning parameters λ and τ_s that achieve the minimax risk bound.

Before presenting the proposition, we first define the formal mean square error (fMSE) and formal noise-plus interference level (fNPI), following Definitions 3.1−3.4 in [32]. fMSE is defined as the MSE of an observable in a large system framework $\mathrm{LSF}(\delta, \rho, \sigma, \gamma_s, p)$, where $\mathrm{LSF}(\delta, \rho, \sigma, \gamma_s, p)$ denotes a sequence of problem instances $(\mathbf{y}; \mathbf{A}, \mathbf{x})_{m,n}$ as per Eq. (1.1) indexed by the problem sizes, and m and n grow proportionally such that $m/n = \delta$. fNPI is expressed as

$$\begin{aligned} \mathrm{fNPI} &= (\frac{u^*}{1+u^*})^2\sigma_s^2 + (\frac{1}{1+u^*})^2(\sigma^2 + \mathrm{fMSE}/\delta), \\ u^* &= \frac{\sigma^2 + \mathrm{fMSE}/\delta}{\sigma_s^2}. \end{aligned}$$

Its minimax value is $\mathrm{NPI}^*(\delta, \rho, \gamma_s^2) \equiv \frac{\gamma_s^2\sigma^2(1+M^*(\delta,\rho,\gamma_s^2)/\delta)}{\gamma_s^2+1+M^*(\delta,\rho,\gamma_s^2)/\delta}$ by replacing fMSE in the equation above with its minimax risk $M^*(\delta, \rho, \gamma_s^2)$.

Proposition 3.4.1. *(1) For any point in the surface, i.e., $\rho \leqslant 1/\delta$ (since $\delta\rho = \varepsilon \leqslant 1$), the minimax risk of GENP-LASSO is bounded, and $M^*(\delta, \rho, \gamma_s^2)$ is given by*

$$M^*(\delta, \rho, \gamma_s^2) = \frac{-G(\delta, \rho, \gamma_s^2) + \sqrt{G(\delta, \rho, \gamma_s^2)^2 + 4\delta\gamma_s^2 M^{\pm}(\delta\rho)}}{2}, \tag{3.37}$$

where $G(\delta, \rho, \gamma_s^2) = \delta\gamma_s^2 + \delta - \gamma_s^2 M^{\pm}(\delta\rho)$.

(2)For $c > 0$, define

$$h^*(\delta, \rho, \gamma_s^2; c) \;\equiv\; h^{\pm}(\delta\rho, c) \;\cdot\; \sqrt{NPI^*}.$$

Then similar to Eq. (3.8), the distribution $p \in \mathcal{F}_{\delta\rho}$ with a fraction $(1 - \delta\rho)$ of its mass at zero and the remaining mass equally at $\pm h^*(\delta, \rho, \gamma_s^2; c)$ is c-nearly-least-favorable, i.e., the formal noise sensitivity of $\hat{x}(\lambda, \tau_s)$ is

$$\frac{-G(\delta, \rho, \gamma_s^2; c) + \sqrt{G(\delta, \rho, \gamma_s^2)^2 + 4(1-c)\delta\gamma_s^2 M^{\pm}(\delta\rho)}}{2}, \tag{3.38}$$

where $G(\delta, \rho, \gamma_s; c) = \delta\gamma_s^2 + \delta - (1-c)M^{\pm}(\delta\rho)\gamma_s^2$.

(3) The formal minimax parameters are given by

$$\begin{aligned}\lambda(\upsilon;\delta,\rho,\sigma,\sigma_s) &\equiv (1+u^*)\cdot\alpha^{\pm}(\delta\rho)\cdot\sqrt{fNPI(\alpha^{\pm};\delta,\rho,\sigma,\sigma_s,\upsilon)}\\ &\quad\times(1-\frac{1}{1+u^*}EqDR(\upsilon;\alpha^{\pm}(\delta\rho))/\delta),\\ \tau_s(\upsilon;\delta,\rho,\sigma,\sigma_s) &\equiv u^*(1-\frac{1}{1+u^*}EqDR(\upsilon;\alpha^{\pm}(\delta\rho))/\delta),\end{aligned}\tag{3.39}$$

where EqDR is the equilibrium detection rate, i.e., the asymptotic fraction of coordinates that are estimated to be nonzero, i.e., EqDR $= P\{\eta(x_\infty;\theta_\infty)\neq 0\}$*, as in Eq. (4.5) in [32].*

Proof. The proof is given in Appendix A.2. □

To show that the noise sensitivity analysis presented here is indeed a generalized result, we next discuss three special cases and show that the result here degrades to the existing known conclusions. First, let $\gamma_s^2=\infty$. In this case, Eq. (3.37) degrades to the formulae of the bounded MSE below the phase transition boundary of AMP, *i.e.*, Eq. (4.8) in [32] . The phase transition boundary only exists in this extreme case for GENP-AMP. Second, if $\gamma_s^2=0$, *i.e.*, $\tilde{\mathbf{x}}=\mathbf{x}$, we do not need to run the AMP; hence the MSE is 0, which coincides with Eq. (3.37) when $\gamma_s^2=0$. Last, if $\delta=0$, which means there is no compressed measurement, solving the minimization problem in Eq. (3.13) is equivalent to scalar denoising, and the minimax MSE is $M^{\pm}(\rho\delta)\sigma_s^2$, which also agrees with the denoising of scalars introduced in Sec. 4.1.

When there is no initial estimation $\tilde{\mathbf{x}}$, the formal MSE noise sensitivity above the phase transition is infinite. However, this is no longer the case in the presence of the GENP, as we can at least assign τ_s to ∞ while keeping λ to be finite, and the formal MSE noise sensitivity is thus bounded by γ_s^2. We can do even better by exploiting the measurement and the sparsity of the original signal, as shown below.

It is easy to verify that $\partial M^*(\delta,\rho,\gamma_s^2)/\partial\gamma_s^2$ is positive, so $M^*(\delta,\rho,\gamma_s^2)$ is a monotonically increasing function of γ_s^2. Since GENP-AMP reduces to AMP when $\gamma_s^2=\infty$, this means that the minimax bound of GENP-LASSO is no greater than that of LASSO, *i.e.*,

$$M^*(\delta,\rho,\gamma_s^2)\leqslant M^b(\delta,\rho),\tag{3.40}$$

where $M^b(\delta,\rho)=\frac{M^{\pm}(\delta\rho)}{1-M^{\pm}(\delta\rho)/\delta}$ is the bound of LASSO minimax risk.

Besides, we can also verify that for a fixed sparsity, i.e., $\varepsilon=\delta\rho$ is a constant, $\partial M^*(\delta,\rho,\gamma_s^2)/\partial\delta$ is non-positive (only equal to 0 when $\delta=0$), and $M^*(\delta,\rho,\gamma_s^2)$ is a monotonically decreasing function of δ. Since GENP-AMP reduces to denoising via soft-thresholding described in Sec. 4.1 when $\delta=0$, we conclude that the minimax bound of GENP-LASSO is no greater than that of scalar denoising,

$$M^*(\delta,\rho,\gamma_s^2)\leqslant M^{\pm}(\delta\rho)\gamma_s^2.\tag{3.41}$$

In fact, Eq. (3.40) and (3.41) have proved that GENP-AMP outperforms AMP and the scalar denoising via soft-thresholding. More importantly, Eq. (3.40) measures the benefit brought by the generalized elastic net prior while Eq. (3.41) measures the benefit brought by the linear CS measurements.

We can find more properties of this minimax risk bound. For a fixed δ, the only function of ρ is $M^{\pm}(\delta\rho)$. From [32], we know that $M^{\pm}(\delta\rho)$ is monotonically increasing with respect to ρ, and $M^{\pm}(0) \to 0,\ M^{\pm}(1) \to 1$. Besides, we can find that $M^*(\delta, \rho, \gamma_s^2)$ is monotonically increasing with respect to $M^{\pm}(\delta\rho)$. The maximum value of $M^{\pm}(\delta\rho)$ is 1. The maximum value of $M^*(\delta, \rho, \gamma_s^2)$ is thus

$$\begin{aligned} &\max_{M^{\pm}(\delta\rho)} M^*(\delta, \rho, \gamma_s^2) \\ &= \frac{\sqrt{(\delta\gamma_s^2 - \gamma_s^2 + \delta)^2 + 4\delta\gamma_s^2} - (\delta\gamma_s^2 - \gamma_s^2 + \delta)}{2}, \end{aligned} \tag{3.42}$$

where the maximum is achieved at $\rho = 1/\delta$.

3.5 Parameterless GENP-AMP

In the GENP-AMP proposed above, two parameters need to be known in advance: (1) the sparsity of the signal, $\varepsilon = k/n$, in order to select the appropriate thresholding parameter in soft thresholding function in Sec. 4.1; (2) the variance of the prior $\tilde{\mathbf{x}}$, σ_s^2, in order to determine the weighting parameter u_t as in Prop. 3.3.2. This makes the algorithm impractical.

The original AMP also needs to know the sparsity. However, recently two types of parameterless AMP algorithms have been developed in [73] and [99, 100]. In [73], Stein's unbiased risk estimate (SURE) framework is used to automatically determine the optimal thresholding parameter in AMP using the gradient descent method. The methods in [99,100] are both based on the GAMP [76], and try to approximate the MMSE result by learning the prior distribution of the sparse signal through Expectation Maximization (EM) method.

In this part, we follow the approach in [73] due to its theoretical guarantee, since the complete analysis of the EM algorithm used in [99, 100] is still not available. However, the method in [73] cannot be applied in this chapter directly since it does not consider the GENP. In the following proposition, using the SURE theory, we develop a practical parameterless version of the GENP-AMP (P-GENP-AMP) that can simultaneously select the thresholding parameter and estimate the variance of the GENP.

Proposition 3.5.1. *The variance of the GENP $\tilde{\mathbf{x}}$ can be approximated by*

$$\sigma_s^2 \approx \frac{\|\tilde{\mathbf{x}} - \mathbf{x}_{AMP}\|_2^2 - \lim\limits_{t\to\infty} \hat{r}(\theta^t)}{n}, \tag{3.43}$$

where $\mathbf{x}_{AMP}$ *is the sparse signal estimated by the AMP with the same setup (fixed* $\mathbf{A}$*,* δ*, and* ρ*),* $\lim_{t\to\infty} \hat{r}(\theta^t)/n$ *is the MSE of AMP predicted by the SURE method in [73], and*

$$\frac{\hat{r}(\theta_t)}{n} = \frac{1}{n}\left\|\eta(\hat{\mathbf{x}}_0^t;\theta_t) - \hat{\mathbf{x}}_0^t\right\|_2^2 + \sigma_t^2 + \frac{1}{n}\sigma_t^2[\mathbf{1}^T(\eta'(\hat{\mathbf{x}}_0^t;\theta_t) - 1)] \tag{3.44}$$

is Eq. (13) in [73], in which σ_t^2 *is the noise-plus interference level in the* t*-th iteration of the standard AMP.*

Proof. The proof is given in Appendix A.3. □

In fact, thanks to the state evolution analysis, the choice of $\mathbf{x}_{\text{AMP}}$ can be quite flexible. Another good choice is $\hat{\mathbf{x}}_0^*$, the un-thresholded estimator in the last iteration of AMP, whose variance is σ_*^2, mentioned in Eq. (3.14). Then, σ_s^2 can also be approximated by

$$\sigma_s^2 \approx \frac{\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_0^*\|_2^2 - \sigma_*^2}{n}. \tag{3.45}$$

Note that as shown in Prop. 3.5.1 and its proof in Appendix A.3, the approximation of σ_s^2 relies on the approximation of the standard AMP. Therefore, above the phase transition boundary of AMP, the AMP approximation is unstable since the MSE is unbounded, making the approximation $\lim_{t\to\infty} \hat{r}(\theta^t)/n$ unbounded. A tiny mismatch between $\lim_{t\to\infty} \hat{r}(\theta^t)/n$ and MSE of AMP will cause large error when estimating σ_s^2. On the other hand, below the phase transition boundary, the MSE of AMP is bounded. The approximation is very stable.

Once σ_s^2 is estimated, the remaining problem is to determine the thresholding parameter in Eq. (3.17). Since the iteration formulae and the state evolutions of GENP-AMP are similar to those of AMP, we only need to replace the explicit expressions of σ_t^2 in Eq. (3.44) with $\text{npi}(q_t^2)$ in Eq. (3.30). The subsequent steps are exactly the same as those in [73], *i.e.*, determining the thresholding parameter θ_t using gradient descent, and updating the estimator and the residual according to Eq. (3.17) and (3.20).

3.6 Numerical Experiments

In this section, we present simulation results with both 1-D data and multiview images to demonstrate the performances of the proposed GENP-LASSO and GENP-AMP. Comparisons with some other methods are also included.

3.6.1 Performance of GENP-LASSO

We first compare the predicted and empirical MSEs of GENP-LASSO and LASSO. Note that GENP-LASSO reduces to LASSO when $\gamma_s^2 = \infty$. We generate the signal vector $\mathbf{x}_0$ by randomly choosing each entry from $\{+1, 0, -1\}$ with probabilities $P(x_{0,i} = +1) = P(x_{0,i} = -1) = 0.064$. The entries of the measurement matrix $\mathbf{A}$ are drawn from the i.i.d. Gaussian

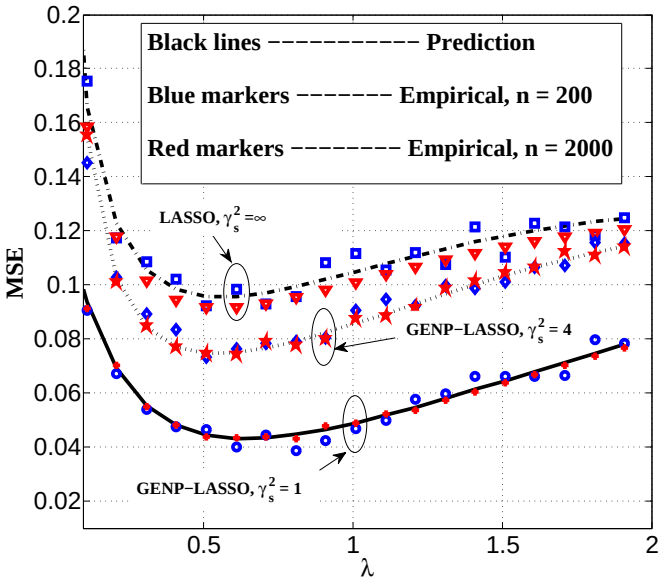


Figure 3.1: The predicted and actual MSEs of LASSO and GENP-LASSO with different regularization parameter λ. The sample rate is $\delta = 0.64$.

δ	ρ	h^*	λ^*	τ^*	fMSE (GENP -AMP)	eMSE (GENP-OWLQN)	eMSE (GENP -AMP)	fMSE (AMP)	eMSE (OWLQN)	eMSE (AMP)	fMSE (DN)	eMSE (DN)
0.100	0.095	2.828	2.585	0.995	0.033	0.032	0.033	0.136	0.119	0.128	0.058	0.062
0.100	0.142	2.807	2.359	0.993	0.047	0.044	0.048	0.380	0.394	0.430	0.079	0.081
0.100	0.170	2.801	2.256	0.992	0.055	0.057	0.056	1.045	1.199	1.089	0.090	0.093
0.100	0.180	2.799	2.223	0.992	0.058	0.058	0.058	2.063	1.958	3.159	0.094	0.103
0.100	1.900	2.656	0.919	0.951	0.405	0.405	0.406	UB	UB	UB	0.486	0.479
0.250	0.134	2.581	2.025	0.995	0.086	0.091	0.088	0.374	0.369	0.366	0.150	0.151
0.250	0.201	2.547	1.796	0.994	0.120	0.121	0.123	1.028	1.213	1.137	0.201	0.203
0.250	0.241	2.533	1.694	0.993	0.139	0.137	0.139	2.830	2.708	2.910	0.228	0.226
0.250	0.254	2.529	1.663	0.992	0.145	0.145	0.148	5.576	6.665	5.680	0.236	0.236
0.250	1.900	2.276	0.511	0.973	0.619	0.625	0.626	UB	UB	UB	0.797	0.790
0.500	0.193	2.362	1.512	0.995	0.182	0.184	0.184	0.853	0.845	0.856	0.315	0.316
0.500	0.289	2.314	1.279	0.992	0.245	0.245	0.245	2.329	2.343	2.412	0.410	0.415
0.500	0.347	2.291	1.172	0.993	0.280	0.275	0.280	6.365	7.232	6.312	0.459	0.465
0.500	0.366	2.285	1.140	0.993	0.291	0.296	0.290	12.427	15.665	12.165	0.475	0.476
0.500	1.900	1.253	0.047	0.986	0.689	0.689	0.696	UB	UB	UB	0.978	0.972

Table 3.1: Empirical and predicted MSEs of different methods for different points in the sampling space.

distribution $\mathcal{N}(0, 1/m)$. The sampling noise $\mathbf{w}$ are drawn from $\mathcal{N}(0, 0.2)$, and the noise $\mathbf{e}$ of the GENP $\tilde{\mathbf{x}}$ are drawn from $\mathcal{N}(0, 0.2\gamma_s^2)$. The simulation setup is the same as that in [72], except for the GENP.

As shown in Sec. 3.3, the MSE of GENP-LASSO is controlled by two regularization parameters λ and τ_s, but they are connected by the hidden parameter u. If one of them is given, using Prop. 3.3.1, Prop. 3.3.2, and Prop. 3.3.3, the other parameters can be uniquely determined.

Fig. 3.1 shows the predicted and the empirical MSEs of LASSO and GENP-LASSO with different λ. Three γ_s^2 are tested, each with two different values of n. In this example,

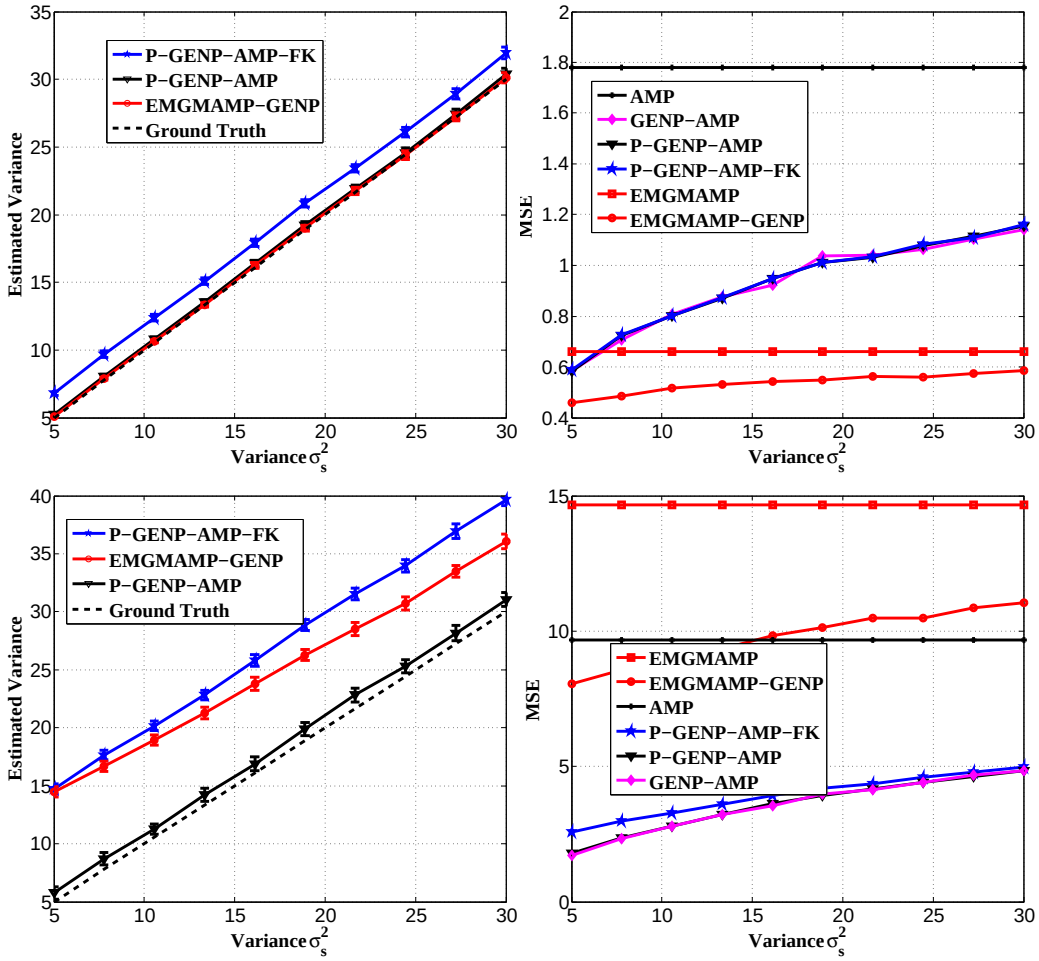


Figure 3.2: Performances of parameterless algorithms with $\delta = 0.5$ and $\varepsilon = 0.2$. First row (from left to right): (a) Estimated σ_s^2 with SNR=20 dB. The confidence level of the error bar is 0.95. (b) MSEs with SNR=20 dB. Second row: (c) Estimated σ_s^2 with SNR=5 dB. The confidence level of the error bar is 0.95. (d) MSEs with SNR=5 dB.

the predicted MSEs of GENP-LASSO are given by the state evolution of GENP-AMP. The empirical results of LASSO and GENP-LASSO for $n = 200$ are obtained by the Matlab-based CVX package [44]. The empirical results of LASSO for $n = 2000$ are obtained by the OWLQN algorithm [7], which is written in C++. The empirical results of GENP-LASSO for $n = 2000$ are obtained by modifying the OWLQN to incorporate the GENP, as described in Sec. 3.2. We denote this as GENP-OWLQN.

It can be seen from Fig. 3.1 that the predicted MSE is quite accurate in both LASSO and GENP-LASSO. The result of LASSO (with $\gamma_s^2 = \infty$) is the same as Fig. 9 in [72]. When $\gamma_s^2 = 4$ or $\gamma_s^2 = 1$, the minimal MSE of GENP-LASSO can be reduced by about 20% and 50%, respectively, compared to the standard LASSO without any prior.

3.6.2 Comparison of AMP, GENP-AMP and Denoising

We now compared the performances of AMP, GENP-AMP and scalar denoising via soft thresholding of the initial estimation when they are operated at different points of the sampling plane, including points below and above the phase transition boundary of the standard AMP. We will compare the predicted and empirical MSEs of GENP-AMP and AMP using the nearly-least-favorable signal generated by Eq. (3.8). We also use OWLQN and GENP-OWLQN to find the LASSO solution $\hat{\mathbf{x}}(\lambda)$ and the GENP-LASSO solution $\hat{\mathbf{x}}(\lambda, \tau_s)$ for Eq. (3.13), but OWLQN-based methods could not predict the MSE, and the regularized parameters need to be chosen manually. The number of iterations of GENP-AMP and AMP for empirical results is fixed as 60.

We generate in each case 20 random realizations of size $n = 2000$, with parameters , $\gamma_s^2 = 1$, $\sigma^2 = 1$, $\delta \in \{0.10, 0.25, 0.50\}$, $\rho \in \{\frac{1}{2}\rho(\delta), \frac{3}{4}\rho(\delta), \frac{9}{10}\rho(\delta),$
$\frac{19}{20}\rho(\delta), 1.9\}$, where $\rho(\delta)$ represents the phase transition boundary of the standard AMP. The results are summarized in Table 3.1, where eMSE and fMSE denote the empirical MSE and predicted formal MSE respectively. DN denotes the denoising method, and UB represents unbounded MSE.

Some observations can be drawn from Table 3.1. First, the MSE of GENP-AMP is much lower than those of AMP and denoising. Secondly, the fMSE and eMSE of GENP-AMP match very well, even when the number of measurements is smaller than the sparsity. For example, for $\rho = 1.9$, the fMSE of GENP-AMP is still very close to eMSE. For AMP, this ρ is much higher than its phase transition boundary. Its MSE is thus unbounded. Thirdly, since the denoising method is equivalent to GENP-AMP with $\delta = 0$, the performance difference between GENP-AMP and denoising shows the contribution of the CS measurements. Finally, although the empirical MSE of GENP-OWLQN is very similar to that of GENP-AMP, GENP-OWLQN is much slower, since it needs to calculate the gradients in each iteration. For example, on a computer with Intel Core i7 3.07GHz CPU and 6.00 GB memory, our Matlab implementation of GENP-AMP is about 10 times faster than the C++ implementation of GENP-OWLQN.

3.6.3 Performance of the Parameterless GENP-AMP

In the previous two simulations, the sparsity ε and the variance σ_s^2 of the prior $\tilde{x}$ are assumed to be known. In this subsection, we show the performance of the parameterless GENP-AMP (P-GENP-AMP), which can estimate σ_s^2. A similar setup to the previous experiments is used, except for the following. The non-zero coefficients of the sparse signal $\mathbf{x}$ follow i.i.d. $\mathcal{N}(0, 100)$. The sampling noise w are drawn from $\mathcal{N}(0, \sigma^2)$ where the variance σ^2 is set according to signal-to-noise ratio (SNR) defined as $\text{SNR} = 10\log_{10}(\frac{1}{m}\left\|\mathbf{Ax}\right\|_2^2/\sigma^2)$, and the noise e of the GENP $\tilde{x}$ are drawn from $\mathcal{N}(0, \sigma_s^2)$. The number of Monte-Carlo simulations is 100.

For comparison purpose, we also estimate σ_s^2 using the following method

$$\sigma_s^2 \approx \frac{\mathbf{1}}{n} \|\tilde{\mathbf{x}} - \mathbf{x}_{\text{AMP}}\|_2^2, \tag{3.46}$$

i.e., we first reconstruct the sparse signal using standard CS reconstruction methods such as AMP, and then use the reconstructed signal and $\tilde{\mathbf{x}}$ to estimate σ_s^2. And we name such kind of algorithm as Parameterless GENP-AMP with faked variance (P-GENP-AMP-FK). In fact, the only difference between Eq. (3.43) and Eq. (3.46) is the term $\lim_{t\to\infty} \hat{r}(\theta^t)/n$, the estimated MSE by the SURE framework proposed in [73].

We also compare with the method in [100], denoted as EMGMAMP, using its source code from [77]. We modify its source code to incorporate the GENP, and treat the variance of GENP as an additional hidden parameter, which can also be updated by the Expectation-Maximization algorithm in [100]. This algorithm is denoted as EMGMAMP-GENP in the following figures. The updating rule follows

$$\sigma_s^2(t) = \frac{1}{n} \sum_{i=1}^{n} [(\tilde{x}_i - \hat{x}_i(t))^2 + \mu_i^x(t)^2], \tag{3.47}$$

where $\hat{x}_i(t)$ and $\mu_i^x(t)$ is the approximate MMSE result, and its standard deviation in the t-th iteration, respectively.

In the first experiment, we consider a high SNR of 20 dB. From Fig. 3.2(a), we can see that P-GENP-AMP, and P-GENP-AMP-FK can both provide good approximations of the variance σ_s^2 while the gap between the ones estimated by P-GENP-AMP and GENP-AMP is exactly the MSE of AMP shown in Fig. 3.2 (b). It can also be seen from Fig. 3.2 (b) that all GENP-based algorithms achieve better performances. EMGMAMP-GENP outperforms the others, since it can learn the prior distribution of the sparse signal through EM and thus achieves near MMSE result. Although the full understanding of EM algorithm is still not available, its efficiency can be proven empirically in this high SNR example. On the other hand, both P-GENP-AMP and P-GENP-AMP-FK perform almost the same as GENP-AMP with known GENP variance. The reason is that at high SNR, the MSE of AMP is very small. Therefore Eq. (3.43) and Eq. (3.46) are very similar.

Fig. 3.2 (c) and (d) show the results with a low SNR of 5 dB. In this case, EMGMAMP-GENP no longer achieves an accurate estimate of σ_s^2, whereas the proposed P-GEMP-AMP still performs well. Moreover, P-GENP-AMP and GENP-AMP are still very close and are much better than other algorithms. The failure of EMGMAMP-GENP is because there are many approximations in EMGMAMP, *e.g.*, using the GAMP approximated posterior as the true one and learning the hidden parameters through EM. At low SNRs, these approximations are not accurate, and the method cannot achieve near MMSE result. Its performance can be even worse than the AMP.

Test sequence	σ^2, σ_s^2	δ	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
Balloons	$1e2, 1e2$	1/5	31.27	**33.72**	**33.72**	32.65	**34.50**	27.25	32.04	32.31
		1/2	34.71	**35.63**	**35.79**	30.41	30.65	28.04	32.04	35.62
	$1e2, 1e3$	1/5	31.27	**32.71**	32.61	32.65	**33.20**	18.02	28.69	14.28
		1/2	34.71	**35.07**	**35.10**	30.43	30.20	19.45	28.69	32.91
	$1e3, 1e3$	1/5	27.83	**30.36**	**30.42**	27.08	25.70	18.01	28.69	15.38
		1/2	29.06	**30.87**	**30.94**	21.17	20.60	18.52	28.69	29.81
Kendo	$1e2, 1e2$	1/5	33.08	**35.88**	**35.82**	34.37	35.56	27.57	33.51	34.77
		1/2	36.22	**37.05**	37.04	30.79	30.89	28.28	33.51	**37.33**
	$1e2, 1e3$	1/5	33.08	34.73	**34.76**	34.37	**35.20**	18.07	30.20	16.77
		1/2	36.22	**36.63**	**36.64**	30.77	30.59	19.50	30.20	35.11
	$1e3, 1e3$	1/5	28.15	**31.86**	**32.00**	28.07	25.98	18.04	30.20	22.30
		1/2	30.26	**32.20**	**32.31**	21.32	20.64	18.57	30.20	31.04
Pantomime	$1e2, 1e2$	1/5	31.65	**34.41**	**34.20**	33.42	33.51	27.43	31.93	24.79
		1/2	**36.46**	36.24	36.36	30.89	30.29	28.20	31.93	**37.62**
	$1e2, 1e3$	1/5	31.65	33.73	**33.77**	33.42	**34.40**	18.06	29.77	24.58
		1/2	36.46	**36.62**	**36.66**	30.88	30.57	19.48	29.77	34.41
	$1e3, 1e3$	1/5	28.50	**31.39**	**31.49**	28.01	25.74	17.63	29.77	26.38
		1/2	30.32	**31.86**	**32.01**	21.34	20.66	18.56	29.77	31.11

Table 3.2: PSNRs of different methods for multiview images. For $\sigma_s^2 = 1e3$, the PSNRs of the corrupted virtual middle views are all 18.03 dB, whereas when $\sigma_s^2 = 1e2$, the PSNRs are 26.96 dB for "Balloons", 27.35 dB for "Kendo", and 27.20 dB for "Pantomime".

3.6.4 Application in Natural Imaging

In this section, we consider a two-dimensional image compressive sensing example. The target signal in this case is the image "Lena" with resolution 512×512. There is also a 128×128 low-resolution version of the same subject. Gaussian noises of different variances are added to the low-resolution version to imitate the noises in poor illumination, high temperature or transmission error. Then, this noisy low-resolution version is upsampled to the resolution of the target signal and served as the initial estimation.

The full size image is partitioned into overlapped blocks of size 48×48 pixels, with an overlap of 6 pixels vertically and horizontally to reduce the blocking artifacts. And we choose DCT as the sparsifying basis. The same i.i.d. Gaussian sensing matrix is applied on each block to obtain the linear CS measurements.

3.6.5 Application in Hybrid Multi-View Imaging System

We next apply the GENP-AMP to the hybrid multi-view imaging system [12,94,103], where a group of cameras capture the scene from different locations. Some cameras are traditional cameras, and others are low-cost CS cameras such as the single pixel cameras [34]. For each CS camera, we assume its left and right neighbouring cameras are traditional cameras. To help the reconstruction from CS sampling, the left and right views are used to generate a virtual view, which serves as the initial estimate or the GENP of the middle view. To simulate the noises in poor illumination, high temperature, or transmission error, we add Gaussian noises of different variances to the CS samples and the virtual views.

In [94], after generating the virtual view from neighboring views, the method recovers the error image between the virtual middle image and the unknown middle image using CS methods, instead of recovering the middle image directly, assuming that the error image is sparser than the original image. However, in the presence of noises, the sparse error assumption could be invalid and this method might be inefficient.

We test the multiview image sequences "Balloons", "Kendo", and "Pantomime" under various channel noise levels. Eight algorithms are compared: AMP (denoted as Alg1), P-GENP-AMP (Alg2), GENP-AMP (Alg3), EMGMAMP (Alg4), EMGMAMP-GENP(Alg5), the residual AMP according to the scheme in [94] (Alg6), denoising of the corrupted virtual middle view via soft-thresholding (Alg7), and the modified CS [97] (Alg8), which finds the sparsest signal outside the support set detected from the prior $\tilde{\mathbf{x}}$. For the denoising algorithm, the parameterless SURE framework in [73] is applied to automatically choose the tuning parameter, and σ_s^2 is assumed to be known.

We partition each image into overlapped blocks of size 48×48 pixels, with an overlap of 6 pixels vertically and horizontally to reduce the blocking artifacts. The DCT is used as the sparsifying basis, and the linear CS measurements are obtained using the same i.i.d. Gaussian sensing matrix on each block. The virtual middle image is generated by Version 3.5 of the MPEG view synthesis reference software (VSRS) [91], and the test sequences are downloaded from [2].

Table 3.2 reports the PSNRs (dB) of the reconstructions given by the eight methods under different σ^2, σ_s^2, and δ. The top-two results in each case are highlighted in bold. The following can be observed. First, almost all the top-two results are P-GENP-AMP and GENP-AMP, and there is no noticeable gap between them, verifying the efficiency of the proposed algorithms. In particular, when $\sigma^2 = 1e3$ and $\sigma_s^2 = 1e3$, *i.e.*, both the CS samples and GENP have low quality, our algorithms always perform the best. Second, when the channel noise level is low and sampling rate is high, *i.e.*, $\sigma^2 = 1e2$, $\sigma_s^2 = 1e2$, and $\delta = 1/2$, the modified CS (Alg6) is comparable to or even better than the proposed methods Alg2 and Alg3. This is as expected, since detecting the support of the virtual view $\tilde{x}$ is easier under low noise levels. However, as the noise level increases, the performance of the modified CS degrades quickly. It also requires the knowledge of σ^2, which is not needed in AMP-based algorithms. Third, at high SNR ($\sigma^2 = 1e2$), EMGMAMP-GENP outperforms the proposed P-GENP-AMP. However, at low SNR ($\sigma^2 = 1e3$), the performance of EMGMAMP-GENP is quite poor. Finally, Our methods are also about 20 times faster than the CVX-based modified CS and comparable to EMGMAMP and EMGMAMP-GENP.

Some examples of the reconstructed images are shown in Fig. 3.3. Our P-GENP-AMP and GENP-AMP provide the best visual quality. All other methods have some limitations. For example, some artifacts exist in the AMP and EMGMAMP. Blurs happen when thresholding-based denoising is used, and Gaussian noises cannot be removed by the residual AMP. Although some parts can be well recovered by the modified CS, it also introduces

severe artifacts in certain areas, due to its poor detection rate of the support set in high noise levels.

3.7 Summary

This chapter studies the generalized elastic net prior (GENP)-aided compressed sensing problem, where an additional noisy version of the original signal is available for CS reconstruction. We develop a GENP-aided approximate message passing algorithm (GENP-AMP), and study its parameter selection, state evolution, and noise sensitivity. The contribution of the GENP is also examined. We also develop a parameterless GENP-AMP that does not need to know the sparsity of the unknown signal and the variance of the GENP. Simulation results with 1-D data and multiview images demonstrate the performances of the proposed methods.

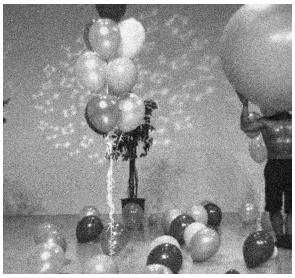
(a) original

(b) AMP (PSNR: 27.83dB)

(c) P-GENP-AMP (30.36dB)

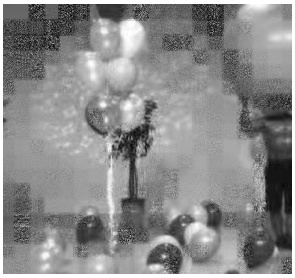
(d) GENP-AMP (30.42dB)

(e) EMGMAMP-GENP (25.70dB)



(f) Residual AMP (18.02dB)

(g) Denoising (28.69dB)



(h) Modified CS (14.28dB)

Figure 3.3: The reconstructed "Balloons" with $\sigma^2 = 1e3, \sigma_s^2 = 1e3, \delta = 1/5$.

Chapter 4

Multi-Resolution Compressed Sensing Reconstruction via Approximate Message Passing

The phase transition theory [32] in CS states that when the undersampling rate is below a certain threshold, the CS algorithm will fail to recover the signal with high probability even if there is no sampling noise. In the noisy case, the noise sensitivity, which is the minimax mean squared error (MSE) of the reconstruction, is unbounded. This is analogous to the rate-distortion bound in information theory. Therefore, in applications in which a large number of CS samples need to be transmitted to a receiver, the receiver has to wait until it receives enough samples before it can recover the signal. This can incur undesired delays.

This chapter is motivated by the following fundamental question: if in the case above we are allowed to reconstruct low-resolution (LR) previews instead of the original high resolution (HR) signal, can we recover high-quality LR signals so that we can enlarge the feasible operating region of the system? We call this framework CS with multi-resolution reconstructions, or MR-CS for short. This framework opens up many questions. For example, how does one design the sampling and reconstruction algorithms? What is the highest resolution that can be reconstructed at each sampling rate? What are the expressions of the phase transition curves for different LR reconstructions? A straightforward approach is to first reconstruct a HR signal using existing reconstruction methods and then downsample the signal. Therefore, another question is how much gain we can obtain over this simple method? Note that a carefully designed LR reconstruction algorithm should at least have lower complexity than this simple method because it can reconstruct the LR signal directly.

To answer these questions, in this chapter, we develop a general theory for MR-CS reconstruction, and propose a MR-AMP algorithm to reconstruct an LR signal if the sampling rate is too low. Our method does not impose any constraint on the measurement matrix. Therefore, it enables more LR reconstruction choices. In addition, theoretical anal-

ysis can still be obtained. Instead of having only one phase transition curve (PTC), we obtain a family of PTCs that specify the sampling rate thresholds to obtain bounded noise sensitivity with different resolutions. Moreover, the noise sensitivity is derived explicitly. The performance of the proposed scheme is verified using both synthetic data and natural images.

The remainder of this chapter is structured as follows: Sec. 4.1 presents the mathematical model of the MR-CS problem and provides the necessary conditions that the MR up/down-sampling matrices should satisfy. Sec. 4.2 is devoted to the MR-AMP algorithm and its updating rule. Sec. 4.3 establishes the theoretical analysis of MR-AMP. Sec. 4.4 discusses the application of MR-AMP to images and develops three sets of up/down-sampling matrices. Sec. 4.5 presents simulation results, validates the state evolution of MR-AMP, and gives guidelines on tuning the parameters of the algorithm. The section also compares the performance of MR-AMP to that of the original HR-AMP with different denoisers in terms of reconstruction quality and algorithm complexity.

4.1 Formulation and Conditions of MR-CS Reconstruction

The goal of the classical CS is to recover a $n_1 \times 1$ vector $\mathbf{x}$ from a $m \times 1$ noisy measurement $\mathbf{y}$ with $m < n_1$, *i.e.*,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}. \tag{4.1}$$

In this chapter, entries of the $m \times n_1$ measurement matrix $\mathbf{A}$ are i.i.d. Gaussian with zero mean and a variance of $1/m$, denoted by $\mathbb{N}(0, 1/m)$. Each entry of the noise vector $\mathbf{w}$ also follows i.i.d. Gaussian distributions with zero mean and a variance of σ_w^2. The CS undersampling ratio is defined as $\delta_1 = m/n_1$.

Because the system is underdetermined, it cannot be solved without exploiting the special structure of $\mathbf{x}$. Some examples of structured signals are given in [28], including simple sparse signals, block sparse signals, mostly constant non-decreasing signals, and piecewise constant signals. Following the notations in [28], the family of probability distributions for a particular type of structured signals over $\mathbb{R}^{n_1}$ is denoted as $\mathbb{F}_{n_1,\varepsilon_1}$, where $\varepsilon_1 \leq 1$ is a constant sparsity ratio, and the expected amount of useful structured information in the signals is at most $k_1 = n_1\varepsilon_1$. The definition of the useful structured information depends on the nature of the structure. Let υ_{n_1} denote a distribution in $\mathbb{F}_{n_1,\varepsilon_1}$, and let $\mathbf{x}$ be a signal with distribution υ_{n_1}. In this paper, we focus on the following two families of structured sparsity.

Definition 4.1.1. *The family of distributions that generates simple sparse signals is defined as (Eq. (1.2) in [28])*

$$\mathbb{F}^{SS}_{n_1,\varepsilon_1} \equiv \left\{ \upsilon_{n_1} : \mathbb{E}_{\upsilon_{n_1}} \left\{ \|\mathbf{x}\|_0 \right\} \leqslant n_1\varepsilon_1 \right\}, \tag{4.2}$$

where the ℓ_0 norm $\|\mathbf{x}\|_0$ denotes the number of nonzero entries of the vector $\mathbf{x}$. Therefore, the expected number of non-zero entries of signals in this family is at most $n_1\varepsilon_1$.

Definition 4.1.2. *The family of distributions that generate piecewise constant signals is defined as (Sec. V in [28])*

$$\begin{aligned}&\mathbb{F}^{PC}_{n_1,\varepsilon_1} \equiv \\ &\left\{\upsilon_{n_1} : \mathbb{E}_{\upsilon_{n_1}}\left\{\#\left\{t \in [1, n_1 - 1] : x_{t+1} \neq x_t\right\}\right\} \leqslant n_1\varepsilon_1\right\},\end{aligned} \tag{4.3}$$

where $\#\{\cdot\}$ denotes the number of times the condition in the operator is true. Therefore, the expected number of change points within signals of this family is at most $n_1\varepsilon_1$.

In the proposed MR-CS reconstruction framework, instead of always recovering the signal with the original resolution n_1, we allow the reconstruction of various lower resolution signals n_d ($n_d < n_1$) when the number of available CS samples is too small.

The MR downsampling factor is defined as

$$d = n_1/n_d. \tag{4.4}$$

Note that this MR downsampling factor should not be confused with the CS undersampling ratio $\delta_1 = m/n_1$. In this paper, we are interested in the case $m < n_d$, *i.e.*, the recovery of the LR signal remains an underdetermined CS problem. The equivalent CS undersampling ratio for the LR reconstruction is $\delta_d = m/n_d = d\delta_1 > \delta_1$. Let k_d be the expected amount of useful information contained in the LR signal. The expected sparsity ratio of the LR signal is $\varepsilon_d = k_d/n_d$. We also define another factor $\rho_d = \varepsilon_d/\delta_d = k_d/m$. Clearly, a signal with larger ρ_d needs more measurements (larger δ_d) to recover.

Let $\mathbf{D}_d$ be a $n_d \times n_1$ downsampling matrix, $\mathbf{U}_d$ be an $n_1 \times n_d$ upsampling matrix, and $\mathbf{x}_d = \mathbf{D}_d\mathbf{x}$ be the $n_d \times 1$ downsampled version of $\mathbf{x}$. The LR-CS problem can be formulated as [53]

$$\begin{aligned}\mathbf{y} &= \mathbf{A}\mathbf{x} + \mathbf{w} = \mathbf{A}(\mathbf{U}_d\mathbf{x}_d + \mathbf{x} - \mathbf{U}_d\mathbf{x}_d) + \mathbf{w} \\ &= \mathbf{A}\mathbf{U}_d\mathbf{x}_d + \mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x} + \mathbf{w},\end{aligned} \tag{4.5}$$

where $\mathbf{A}\mathbf{U}_d$ is the equivalent measurement matrix for the LR signal $\mathbf{x}_d$ and $\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$ is the additional approximation error term when $\mathbf{x}_d$ is the target signal to be recovered. Note that this error term depends on the signal $\mathbf{x}$.

The downsampling and upsampling matrices $\mathbf{D}_d$ and $\mathbf{U}_d$ play an important role in the MR-CS. In this paper, we require them to satisfy three conditions.

Condition 4.1.1. *The downsampling and upsampling matrices $\mathbf{D}_d$ and $\mathbf{U}_d$ should be chosen such that, if we first upsample an LR signal and then downsample the signal to the original resolution, we can retrieve the original LR signal without any error. Specifically,*

$$\mathbf{D}_d\mathbf{U}_d = \mathbf{I}_{n_d}. \tag{4.6}$$

Because $\mathbf{U}_d$ is a tall matrix, this mild condition can be easily satisfied. In [41], the authors designed a special two-resolution CS system such that a $m \times 1$ LR signal can be recovered directly from the $m \times 1$ CS sample $\mathbf{y}$. This can be considered as a special case of our setup.

The second condition concerns the quality of the measurement matrix for the LR reconstruction.

Condition 4.1.2. *The quality of the equivalent measurement matrix for the LR reconstruction should be no worse than that of the HR reconstruction.*

For different reconstruction algorithms, different criteria are used to evaluate the quality of the measurement matrix, *e.g.*, the RIP constant for the basis pursuit algorithm [15] and the mutual coherence for the orthogonal matching pursuit algorithm [95]. The solution in this paper is based on the AMP algorithm; hence, we follow the requirement in [28,31,32,70] that each entry of the LR measurement matrix should be i.i.d. Gaussian with zero mean and a variance of $1/m$.

Because $m < n_d$ in our case, the MR-CS problem here cannot be solved directly without exploiting the structure of $\mathbf{x}_d$. Moreover, the LR signal should be easier to recover than the HR signal, *i.e.*, the amount of useful information k_d contained in $\mathbf{x}_d$ should be no more than the amount k_1 in the original HR signal $\mathbf{x}$. We therefore also require the downsampling matrix $\mathbf{D}_d$ to satisfy the following condition.

Condition 4.1.3. *If* $\mathbf{x}$ *belongs to the family* $\mathbb{F}_{n_1,\varepsilon_1}$ *in the basis* $\mathbf{\Psi}$*, the downsampling matrix* $\mathbf{D}_d$ *should be chosen such that* $\mathbf{x}_d = \mathbf{D}_d\mathbf{x}$ *belongs to the family* $\mathbb{F}_{n_d,\varepsilon_d}$ *in the basis* $\mathbf{\Psi}_d = \{\mathbf{D}_d\mathbf{\Psi}\} - \{\mathbf{0}\}$ *with* $\varepsilon_d \leqslant d\varepsilon_1$.

Some results similar to Cond. 4.1.3 were reported in [53] for simple sparse vectors, which is a special case of Cond. 4.1.3, as summarized below.

Condition 4.1.4. *If* $\mathbf{x}$ *is sparse in the basis* $\mathbf{\Psi}$*, then* $\mathbf{x}_d = \mathbf{D}_d\mathbf{x}$ *is sparse in the non-zero projected low-dimension basis* $\mathbf{\Psi}_d = \{\mathbf{D}_d\mathbf{\Psi}\} - \{\mathbf{0}\}$*. The sparsity* k_d *of* $\mathbf{x}_d$ *is no larger than* k*, the sparsity of* $\mathbf{x}$*, if the columns of* $\mathbf{\Psi}_d$ *are linearly independent.*

Our condition in Cond. 4.1.3 is not restricted to simple sparse vectors and can be used for other special structures that $\mathbf{x}$ follows such as piecewise constancy.

In Sec. 4.4, we will design three pairs of up-/down-sampling matrices for images that satisfy the three conditions above perfectly or approximately. One pair of these matrices is for simple sparse signals, and the other two pairs are for piecewise constant signals. The conditions listed above can also be used to design matrices for the multi-resolution reconstructions of other types of structured sparse signals.

Note that the term "multi-resolution" in our paper is slightly different from that in the wavelet transform literature because our method only reconstructs each of these LR signals

independently, and how to use an LR reconstruction to assist in HR reconstruction is not addressed in this paper. Nevertheless, we will show in Table 4.8 that we can sometimes provide a better HR image compared to when reconstructing the HR image directly from the measurements by simply upsampling the recovered LR image to the target HR.

4.2 Multi-Resolution Approximate Message Passing

In this section, we propose an approximate message passing (AMP)-based algorithm to solve the MR-CS problem. Without loss of generality, we assume that the signal belongs to the structured sparse family $\mathbb{F}_{n_1,\varepsilon_1}$ in the canonical basis.

The main idea of the original AMP is to transform the CS reconstruction problem into a denoising problem [28], *i.e.*, estimating $\mathbf{x}_o$ from its noisy observations $\mathbf{x}_o+\sigma\mathbf{e}$, where entries of $\mathbf{e}$ are i.i.d. Gaussian with zero mean and unit variance, and σ is a constant. In each iteration of AMP, pseudo-data $\mathbf{z}^t = \mathbf{x}^t + \mathbf{A}^T r^t$ are first formed. They are then denoised by a denoising function $\eta_{\sigma^t}(\mathbf{z}^t;\tau)$, where σ^t is the standard deviation (std) of $\mathbf{z}^t$ and τ is the tuning parameter of the denoiser. Finally, the residual of the measurements is updated. Specifically,

$$\begin{aligned}\mathbf{z}^t &= \mathbf{x}^t + \mathbf{A}^T\mathbf{r}^t,\\ \mathbf{x}^{t+1} &= \eta_{\sigma^t}(\mathbf{z}^t;\tau),\\ \mathbf{r}^{t+1} &= \mathbf{y} - \mathbf{A}\mathbf{x}^{t+1} + b^t\mathbf{r}^t,\end{aligned} \tag{4.7}$$

where b^t is the Onsager term, which is related to the divergence of the denoiser by

$$b^t = \frac{1}{m}\mathrm{div}\eta_{\sigma^{t-1}}(\mathbf{u};\tau)|_{\mathbf{u}=\mathbf{z}_d^{t-1}} = \frac{1}{m}\sum_{i=1}^{n_1}\frac{\partial\eta_{\sigma^{t-1}}(\mathbf{u};\tau)}{\partial u[i]}|_{\mathbf{u}=\mathbf{z}^{t-1}}. \tag{4.8}$$

For different structured signals, different denoisers $\eta_{\sigma^t}(\cdot)$ should be used. For example, for simple sparse signals, the well-known soft-thresholding should be used, whereas a total variation (TV) denoiser is more appropriate for piecewise constant signals [28].

To apply AMP to the MR-CS problem in Eq. (4.5), we propose the following multi-resolution approximate message passing algorithm (MR-AMP):

$$\begin{aligned}\mathbf{z}_d^t &= \mathbf{x}_d^t + \mathbf{A}_d^T\mathbf{r}_d^t,\\ \mathbf{x}_d^{t+1} &= \eta_{\sigma_d^t}(\mathbf{z}_d^t;\tau),\\ \mathbf{r}_d^{t+1} &= \mathbf{y} - \mathbf{A}_d\mathbf{x}_d^{t+1} + b_d^t\mathbf{r}_d^t,\end{aligned} \tag{4.9}$$

where $\mathbf{A}_d = \mathbf{A}\mathbf{U}_d\mathbf{\Lambda}$ is the corresponding measurement matrix for the LR reconstruction, with $\mathbf{\Lambda}$ being a diagonal matrix determined by the upsampling matrix $\mathbf{U}_d$ to normalize the columns of $\mathbf{A}\mathbf{U}_d$. b_d^t is similar to Eq. (4.8) except that n_1 becomes n_d. Instead of estimating

$\mathbf{x}_o$, we attempt to estimate $\mathbf{x}_{d,o} = \mathbf{D}_d\mathbf{x}_o$ from the pseudo-data $\mathbf{z}_d^t$ with std σ_d^t using the denoising function $\eta_{\sigma_d^t}(\cdot)$.

The original AMP in Eq. (4.7) is a special case of MR-AMP in Eq. (4.9) with $d = 1$. In this paper, we denote the original AMP as high-resolution approximate message passing (HR-AMP) and MR-AMP with $d > 1$ as low-resolution approximate message passing (LR-AMP). Because the dimensions of $\mathbf{A}_d$ and $\mathbf{x}_d$ are smaller than those of $\mathbf{A}$ and $\mathbf{x}$, the complexity of LR-AMP is thus lower than HR-AMP. Note that the proposed MR-AMP does not impose any additional constraint on the measuring matrix $\mathbf{A}$ in the original AMP. It only modifies the reconstruction algorithm to obtain different LR estimates of the signal.

4.3 State Evolution and Phase Transition of MR-AMP

In this section, we analyze the theoretical performance of the proposed MR-AMP in terms of its state evolution, phase transition, and noise sensitivity.

4.3.1 State Evolution

The availability of the state evolution analysis represents an important advantage of AMP over many other CS algorithms. Empirical findings show that the MSEs of AMP with various denoisers can be predicted accurately by its state evolution [28, 70], which describes the asymptotic limit of the AMP estimates in Eq. (4.7) when $m,\ n_1 \to \infty$, for any fixed t [31]. Starting from $\theta^0 = \|\mathbf{x}_o\|_2^2/n_1$, the state evolution generates a sequence of numbers through the following iterations.

$$\begin{aligned}(\sigma^t)^2 &= \frac{1}{\delta_1}\theta^t(\mathbf{x}_o, \delta_1, \sigma_w^2, \tau) + \sigma_w^2,\\ \theta^{t+1}(\mathbf{x}_o, \delta_1, \sigma_w^2, \tau) &= \frac{1}{n_1}\mathbb{E}\left\|\eta_{\sigma^t}(\mathbf{x}_o + \sigma^t\mathbf{e}; \tau) - \mathbf{x}_o\right\|_2^2,\end{aligned} \tag{4.10}$$

where the expectation is with respect to $\mathbf{e} \sim \mathbb{N}(0, \mathbf{I})$. For large values of m and n_1, the state evolution predicts the MSE of the AMP algorithm in Eq. (4.7), *i.e.*, $\theta^t(\mathbf{x}_o, \delta_1, \sigma_w^2, \tau) \approx \frac{1}{n_1}\left\|\mathbf{x}^t - \mathbf{x}_o\right\|_2^2$.

To obtain the state evolution of the proposed MR-AMP, we start from $\theta_d^0 = \|\mathbf{x}_{d,o}\|_2^2/n_d$, where $\mathbf{x}_{d,o}$ is the target LR signal. Let $\sigma_{d,w}^2$ denote the variance of the MR-AMP noise in Eq. (4.5), including contributions from the approximation error and measurement noise, which is equal to $(\sigma_w^2 + 1/m\,\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\|_2^2)$, as will be shown in Sec. 4.3.3. The state evolution of the MR-AMP is thus given by the following iterations.

$$\begin{aligned}(\sigma_d^t)^2 &= \frac{1}{\delta_d}\theta_d^t(\mathbf{x}_{d,o}, \delta_d, \sigma_{d,w}^2, \tau) + \sigma_{d,w}^2,\\ \theta_d^{t+1}(\mathbf{x}_{d,o}, \delta_d, \sigma_{d,w}^2, \tau) &= \frac{1}{n_d}\mathbb{E}\left\|\eta_{\sigma_d^t}(\mathbf{x}_{d,o} + \sigma_d^t\mathbf{e}; \tau) - \mathbf{x}_{d,o}\right\|_2^2,\end{aligned} \tag{4.11}$$

where σ_d^t is the predicted std of the estimate $\mathbf{z}_d^t$ in Eq. (4.9). If $d = 1$, Eq. (4.11) reduces to that of AMP in Eq. (4.10).

Note that the state evolution of AMP is only proved rigorously for scalar denoisers and not for non-scalar denoisers such as total-variation-based denoisers and other more advanced denoisers [8,70,90]. However, similar to observations in these papers, the empirical findings in Sec. 4.5 show that, in all cases studied in this paper, the MSEs of the MR-AMP can be predicted accurately using the state evolution above.

4.3.2 Noiseless Phase Transition of LR-AMP

In CS reconstruction without sampling noise, the phase transition curve (PTC) defines the minimum number of CS measurements required to perfectly recover $\mathbf{x}_o$, *i.e.*, $\theta^\infty(\mathbf{x}_o, \delta_1, 0, \tau) \to 0$ [28]. In this part, we investigate the noiseless phase transition of MR-AMP, where we assume both $\sigma_w^2 = 0$ and $\|\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\|_2^2 = 0$ in Eq. (4.5). The latter is possible for some special signals, and an example will be given in Sec. 4.5. We will show that by allowing LR reconstruction, the MR-AMP admits a family of PTCs, thereby enabling perfect reconstruction of an LR signal in the infeasible region of the original HR-AMP. This is an important generalization of the AMP theory.

The family $\mathbb{F}_{n,\varepsilon}$ is scale invariant [28], *i.e.*, $\eta_\sigma(\mathbf{y};\tau) = \sigma\eta_1(\mathbf{y}/\sigma;\tau)$. Therefore, we only need to consider $\sigma = 1$, and we can simplify the notation $\eta_\sigma(\mathbf{y};\tau)$ as $\eta(\mathbf{y};\tau)$. We then define the following asymptotic minimax MSE when a denoiser η with parameter τ is used to recover signals in the structured sparse family $\mathbb{F}_{n_1,\varepsilon_1}$ [28].

$$M(\varepsilon_1|\eta) \equiv \lim_{n_1\to\infty} \frac{1}{n_1} \inf_{\tau} \sup_{v_{n_1}\in\mathcal{F}_{n_1,\varepsilon_1}} \mathbb{E}_{v_{n_1}} \|\eta(\mathbf{x}_o + \mathbf{e};\tau) - \mathbf{x}_o\|_2^2 , \tag{4.12}$$

In words, $M(\varepsilon_1|\eta)$ is obtained by tuning the denoiser parameter to minimize the MSE per coordinate of the least favorable distribution in the family. The tuning rules of the parameters τ are provided in Sec. 4.5.1.

The minimax MSE has some basic properties [28, 32]. First, because the denoising can improve the reconstruction, we have $0 \le M(\varepsilon_1|\eta) \le 1$. Moreover, $M(\varepsilon_1|\eta) \to 0$ when $\varepsilon_1 \to 0$, and $M(\varepsilon_1|\eta) \to 1$ when $\varepsilon_1 \to 1$. Second, $M(\varepsilon_1|\eta)$ is monotonically increasing with respect to ε_1 [28] because the reconstruction difficulty increases with ε_1.

The detailed expression of $M(\varepsilon_1|\eta)$ for AMP with various denoisers is derived in [28, 31,32]. More importantly, it is shown in [28] that $M(\varepsilon_1|\eta)$ defines the minimum CS undersampling ratio δ_1 for perfect reconstruction, *i.e.*, it describes the phase transition curve of AMP as follows.

Theorem 4.3.1. *In the noiseless case, when using AMP with denoiser η to reconstruct signals in $\mathbb{F}_{n_1,\varepsilon_1}$, the AMP succeeds with high probability if*

$$\delta_1 > M(\varepsilon_1|\eta). \tag{4.13}$$

Vice versa, AMP fails with high probability for $\delta_1 < M(\varepsilon_1|\eta)$.

Combining Theorem 4.3.1 and the conditions in Sec. 4.1, we obtain the following generalized phase transition result for MR-AMP, which specifies the minimum sampling ratio to perfectly recover an LR signal. When $d = 1$, the result reduces to Theorem 4.3.1.

Corollary 4.3.2. *When Cond. 4.1.1, 4.1.2 and 4.1.3 are satisfied, if a signal $\mathbf{x} \in \mathbb{F}_{n_1,\varepsilon_1}$ is sampled according to Eq. (4.1) and if $\sigma_w^2 = 0$ and $\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\|_2^2 = 0$ in Eq. (4.5), then an LR signal $\mathbf{x}_d \in \mathbb{F}_{n_d,\varepsilon_d}$ with $\varepsilon_d \leqslant d\,\varepsilon_1$ can be reconstructed perfectly with high probability via the LR-AMP in Eq. (4.9) when the CS undersampling ratio satisfies*

$$\delta_1 > M(d\varepsilon_1|\eta)/d, \tag{4.14}$$

where $M(\varepsilon_1|\eta)$ is the minimax MSE of the original HR-AMP. On the other hand, the LR-AMP fails with high probability for $\delta_1 < M(d\varepsilon_1|\eta)/d$.

Proof. As mentioned above, $\delta_d = d\delta_1$. Because there is no approximation error in Eq. (4.5), Theorem 4.3.1 can be applied directly to the LR-AMP. Therefore, the LR-AMP succeeds with high probability if the CS sampling ratio satisfies

$$\delta_d = d\delta_1 > M(\varepsilon_d|\eta).$$

If Cond. 4.1.3 is satisfied, we have $\varepsilon_d \leqslant d\varepsilon_1$. Eq. (4.14) can thus be obtained using the property that $M(\varepsilon_d|\eta)$ is monotonically increasing with respect to ε_d. □

The next result shows that the LR reconstruction requires a lower sampling rate than does the HR-AMP. Specifically, the LR-AMP has a larger feasible operating region than the original HR-AMP under certain conditions.

Corollary 4.3.3. *If $M(\varepsilon_1|\eta)$ is a concave function of ε_1, then we have $M(d\varepsilon_1|\eta)/d \leq M(\varepsilon_1|\eta)$.*

Proof. It is known that if a function f is concave and $f(0) \geq 0$, then f is subadditive, *i.e.*, $f(x+y) \leq f(x)+f(y)$. From this, we can obtain $f(tx) \leq tf(x)$ for $t \geq 1$. It is clear from the definition that $M(\varepsilon_1|\eta) \geq 0$. Therefore, if $M(\varepsilon_1|\eta)$ is concave, then, by the subadditivity property, we can obtain $M(d\varepsilon_1|\eta) \leq dM(\varepsilon_1|\eta)$, *i.e.*, $M(d\varepsilon_1|\eta)/d \leq M(\varepsilon_1|\eta)$. □

The concavity condition of $M(\varepsilon_1|\eta)$ is satisfied for many families of structured signals. In particular, this is proved in [30] for simple sparse signals in Eq. (4.2) when the soft-thresholding denoiser is used. It is also confirmed in [28] for block-sparse signals with a

block soft-thresholding denoiser. In the Appendix, we prove that it is satisfied for piecewise constant signals. Finally, we also show in Sec. 4.4 that the concavity condition holds for 2D images in both the simple sparse and piecewise constant families.

Corollary 4.3.3 confirms the motivation discussed in the introduction of the paper, *i.e.*, if the CS sampling rate is too low, although the full-resolution reconstruction will fail, we can still reconstruct an LR version of the signal. Moreover, in the noiseless case, given δ_1, ε_1, we can precisely determine the critical downsampling factor d by solving the equation $\delta_1 = M(d\varepsilon_1|\eta)/d$.

4.3.3 Noise Sensitivity of MR-AMP

The noiseless case studied above is quite restrictive. In practice, we are more interested in the performance of the algorithm in the presence of noise. In this part, we study the noise sensitivity of LR-AMP when the noises $\mathbf{w}$ and $\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$ in Eq. (4.5) are not zero. As in [32, 70], the noise sensitivity of HR-AMP is defined as

$$NS(\sigma_w^2, \delta_1) = \inf_{\tau} \sup_{v_{n_1} \in F_{n_1,\varepsilon_1}} \mathbb{E}_{v_{n_1}}\{\theta^{\infty}(\mathbf{x}_o, \delta_1, \sigma_w^2, \tau)\},$$

which is the minimax MSE per coordinate of the HR-AMP output when the iteration number goes to ∞ in Eq. (4.10). It is shown in [32, 70] that, when the undersampling ratio meets the same phase transition condition as in Theorem 4.3.1, the structured sparse signal can be recovered with a bounded noise sensitivity.

When studying the noise sensitivity of the LR-AMP, we use $NS(\sigma_{d,w}^2, \delta_d)$ to represent the noise sensitivity of LR-AMP, where $\sigma_{d,w}^2$ is the variance of the LR-AMP noise. The next result shows that, when the undersampling ratio meets the same condition as in Corollary 4.3.2, we can also recover the LR signal $\mathbf{x}_d$ with a bounded noise sensitivity.

Corollary 4.3.4. *When Cond. 4.1.1, 4.1.2 and 4.1.3 are satisfied, if the undersampling ratio satisfies Eq. (4.14), i.e., $\delta_1 > M(d\varepsilon_1|\eta)/d$ in the compressed sensing of $\mathbf{x} \in \mathbb{F}_{n_1,\varepsilon_1}$ in Eq. (4.1) with noise variance σ_w^2, an LR version of the signal $\mathbf{x}_d \in \mathbb{F}_{n_d,\varepsilon_d}$ with $\varepsilon_d \leqslant d\varepsilon_1$ can be reconstructed via LR-AMP with the downsampling matrix $\mathbf{D}_d$ and the upsampling matrix $\mathbf{U}_d$. In addition, the noise sensitivity is bounded by*

$$\begin{aligned} &NS(\sigma_{d,w}^2, \delta_d) \\ &\leqslant \frac{M(d\varepsilon_1|\eta)}{1 - M(d\varepsilon_1|\eta)/(d\delta_1)}(\sigma_w^2 + \frac{1}{m}\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\|_2^2). \end{aligned} \tag{4.15}$$

Proof. According to Prop. 2 in [70], the noise sensitivity of AMP with various denoisers is bounded by

$$NS(\sigma_w^2, \delta_1) \leqslant \frac{M(\varepsilon_1|\eta)}{1 - M(\varepsilon_1|\eta)/\delta_1}\sigma_w^2.$$

Replacing δ_1, ε_1 and σ_w^2 by δ_d, ε_d and $\sigma_{d,w}^2$ in the formula above, respectively, we have

$$NS(\sigma_{d,w}^2, \delta_d) \leq \frac{M(\varepsilon_d|\eta)}{1 - M(\varepsilon_d|\eta)/\delta_d}(\sigma_w^2 + \sigma_{d,w}^2).$$

Because $M(\varepsilon_d|\eta)$ is monotonically increasing with ε_d, it is clear that $\frac{M(\varepsilon_d|\eta)}{1-M(\varepsilon_d|\eta)/\delta_d}$ is also monotonically increasing. Together with $\varepsilon_d \leq d\varepsilon_1$, we can obtain

$$NS(\sigma_{d,w}^2, \delta_d) \leq \frac{M(d\varepsilon_1|\eta)}{1 - M(d\varepsilon_1|\eta)/(d\delta_1)}(\sigma_w^2 + \sigma_{d,w}^2).$$

By the central limit theorem, if the entries of $\mathbf{A}$ follow a i.i.d. $\mathbb{N}(0, 1/m)$ distribution and if $\mathbf{D}_d$ and $\mathbf{U}_d$ are deterministic, then for a given $\mathbf{x}$, each entry of $\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$ converges to an i.i.d. Gaussian distribution with zero mean and variance $1/m \left\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\right\|_2^2$. Therefore, the equivalent noise variance $\sigma_{d,w}^2$ for the LR-AMP problem is $(\sigma_w^2 + 1/m \left\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\right\|_2^2)$, which proves the result. □

In contrast to the original AMP, the upper bound of the LR-AMP noise sensitivity $NS(\sigma_{d,w}^2, \delta_d)$ is conditional because it depends on the approximation error term $(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$, which varies for different input signals. Therefore, it is crucial to design good up-/down-sampling matrices to reduce the LR reconstruction error, which will be studied in Sec. 4.4. It should be noted that the upper bound is finite in many applications. Moreover, we can sometimes further derive a signal-independent upper bound. For example, in 8-bit images, the pixel value ranges from 0 to 255. Therefore, the worst value of each entry in $(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$ is 255, and the worst value of $\left\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\right\|_2^2$ is thus $255^2 n_1$. The upper bound in Eq. (4.15) can be further bounded by

$$\begin{aligned} NS(\sigma_{d,w}^2, \delta_d) &\leq \frac{M(d\varepsilon_1|\eta)}{1 - M(d\varepsilon_1|\eta)/(d\delta_1)}(\sigma_w^2 + \frac{255^2}{\delta_1}) \\ &\leq \frac{M(d\varepsilon_1|\eta)}{1 - M(d\varepsilon_1|\eta)/(d\delta_1)}(\sigma_w^2 + \frac{255^2 d}{M(d\varepsilon_1|\eta)}). \end{aligned} \tag{4.16}$$

The upper bound above is overly pessimistic because the LR approximation $\mathbf{U}_d\mathbf{D}_d\mathbf{x}$ usually has a much smaller approximation error than 255. The upper bound can be reduced if a more accurate estimate of $\left\|(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}\right\|_2^2$ is known.

Corollary 4.3.4 is more general than Corollary 4.3.2 because it allows for sampling noise and LR approximation noise. The corollary gives further affirmative answers to the questions raised in the introduction of the paper, *i.e.*, if the CS sampling rate is too low for the full-resolution signal recovery, we can reconstruct an LR version of the signal with bounded noise sensitivity. The noisy case shares the same PTC as the noiseless case, as in the original AMP, which serves as a guideline for determining the critical resolution under which the noise sensitivity of the LR signal recovery is bounded.

4.4 Design of Downsampling and Upsampling Matrices for MR-AMP

In this section, we give examples of the design of the up-/down-sampling matrices that satisfy the three conditions in Sec. 4.1 perfectly or approximately so that they can be used in MR-AMP-based image reconstruction. Three pairs of matrices will be designed. The first pair is in the DCT or wavelet transform domain and is designed for the simple sparse family. The other two pairs are in the spatial domain and are suitable for piecewise constant signals.

In [53], DCT-based and total-variation (TV)-based up-/down-sampling matrices are designed for videos such that the downsampling matrix $\mathbf{D}_d$ satisfies Cond. 4.1.4 and the upsampling matrix $\mathbf{U}_d$ satisfies Cond. 4.1.1. However, the proof in that reference mainly concerns TV-based up-/down-sampling matrices. Moreover, the impact of MR design on the quality of the measurement matrix is not considered, *i.e.*, it is not clear whether Cond. 4.1.2 holds.

4.4.1 Transform-Domain Downsampling and Upsampling

Natural images are approximately sparse in the DCT or wavelet domain. The sparse representation of a $n_1 \times n_1$ image $\mathbf{X}$ thus belongs to the simple sparse family in Eq. (4.2), and the soft-thresholding denoiser can be used in the transform domain. To apply CS sampling and reconstruction to images, we need to introduce the transform basis to Eq. (4.1) and Eq. (4.5).

For an $n_1 \times n_1$ image $\mathbf{X}$, an $n_d \times n_d$ LR image $\mathbf{X}_d$ can be obtained via transform-domain downsampling by first applying an HR 2D transform, extracting the $n_d \times n_d$ low-frequency coefficients, and then applying the LR 2D inverse transform [36, 84].

Let $\mathbf{\Psi}_{n_1}$ and $\mathbf{\Psi}_{n_d}$ represent the $n_1 \times n_1$ and $n_d \times n_d$ DCT or orthogonal multiple-level wavelet transform, respectively. We use the following 1D transform-domain downsampling operator [84]

$$\mathbf{D}_d = \sqrt{\frac{1}{d}}\, \mathbf{\Psi}_{n_d}^T \mathbf{I}_{n_d \times n_1} \mathbf{\Psi}_{n_1}. \tag{4.17}$$

where the fat identity matrix $\mathbf{I}_{n_d \times n_1}$ serves as a truncation operator because it only keeps the first n_d coefficients of the input after being transformed by $\mathbf{\Psi}_{n_1}$.

Given the downsampling matrix, one way to satisfy Cond. 4.1.1, *i.e.,* $\mathbf{D}_d\mathbf{U}_d = \mathbf{I}$, is to use transform-domain zero-padding. The corresponding upsampling matrix $\mathbf{U}_d$ is

$$\mathbf{U}_d = \sqrt{d}\, \mathbf{\Psi}_{n_1}^T \mathbf{I}_{n_1 \times n_d} \mathbf{\Psi}_{n_d}. \tag{4.18}$$

The 2D downsampling and upsampling can thus be represented as

$$\begin{aligned} \mathbf{X}_d &= \mathbf{D}_d \mathbf{X} \mathbf{D}_d^T, \\ \hat{\mathbf{X}} &= \mathbf{U}_d \mathbf{X}_d \mathbf{U}_d^T. \end{aligned} \tag{4.19}$$

It should be noted that, according to the definitions in [36], for the downsampling in the DCT domain, we can achieve a non-integer downsampling ratio because we simply take the top left $n_d \times n_d$ low-frequency coefficients and apply the LR 2D inverse DCT transform. However, for the downsampling in the wavelet domain, we can only obtain an integer downsampling ratio that is a power of 2 because the LR image is the appropriately scaled low-pass subband in the multi-level wavelet transform.

Let $\mathbf{x}$, $\mathbf{x}_d$, and $\hat{\mathbf{x}}$ be the vectorized versions of $\mathbf{X}$, $\mathbf{X}_d$, and $\hat{\mathbf{X}}$, respectively, by concatenating the columns of each matrix together. Let $\otimes$ denote the Kronecker product. The 2D downsampling and upsampling can be converted to the following 1D formulas.

$$\begin{aligned} \mathbf{x}_d &= (\mathbf{D}_d \otimes \mathbf{D}_d)\mathbf{x}, \\ \hat{\mathbf{x}} &= (\mathbf{U}_d \otimes \mathbf{U}_d)\mathbf{x}_d. \end{aligned} \tag{4.20}$$

Similarly, let $\mathbf{S}_1 = \mathbf{\Psi}_{n_1} \mathbf{X} \mathbf{\Psi}_{n_1}^T$ and $\mathbf{S}_d = \mathbf{I}_{n_d \times n_1} \mathbf{S}_1 \mathbf{I}_{n_1 \times n_d}$ be the 2D transform of $\mathbf{X}$ and its low-frequency part, and let $\mathbf{s}_1$ and $\mathbf{s}_d$ be their vectorized versions. The 2D inverse transform can be represented by a 1D transform as follows:

$$\begin{aligned} \mathbf{x} &= (\mathbf{\Psi}_{n_1}^T \otimes \mathbf{\Psi}_{n_1}^T)\mathbf{s}_1, \\ \mathbf{x}_d &= \frac{1}{d}(\mathbf{\Psi}_{n_d}^T \otimes \mathbf{\Psi}_{n_d}^T)\mathbf{s}_d, \end{aligned} \tag{4.21}$$

where the two matrices remain orthogonal. Note that the corresponding 1D downsampling ratio is $n_1^2/n_d^2 = d^2$. Clearly, the concavity condition in Corollary 4.3.3 holds here because the 1D sparse representation of a 2D image is simply the vectorized version of its 2D representation.

We next show that the transform-domain up-/down-sampling operators defined above satisfy Cond. 4.1.2 and Cond. 4.1.3.

First, we assume $\mathbf{s}_1 \in \mathbb{F}^{SS}_{n_1^2,\varepsilon_1}$. Because the transform-domain downsampling operator simply extracts the low-frequency components of $\mathbf{s}_1$, the number of nonzero entries in $\mathbf{s}_d$ is certainly no more than that in $\mathbf{s}_1$; hence, $\varepsilon_d \leq n_1^2\varepsilon_1/n_d^2 = d^2\varepsilon_1$, and $\mathbf{s}_d \in \mathbb{F}^{SS}_{n_d^2,d^2\varepsilon_1}$. Cond. 4.1.3 is thus satisfied.

To check Cond. 4.1.2, note that the equivalent 1D measurement matrix for the HR signal is $\mathbf{\Phi}_1 = \mathbf{A}(\mathbf{\Psi}_{n_1}^T \otimes \mathbf{\Psi}_{n_1}^T)$, whereas the equivalent 1D measurement matrix for the

LR-CS problem in Eq. (4.5) is

$$\begin{aligned}\mathbf{\Phi}_d &= \frac{1}{d}\mathbf{A}(\mathbf{U}_d \otimes \mathbf{U}_d)(\mathbf{\Psi}_{n_d}^T \otimes \mathbf{\Psi}_{n_d}^T) \\ &= \mathbf{A}(\mathbf{\Psi}_{n_1}^T \mathbf{I}_{n_1 \times n_d}) \otimes (\mathbf{\Psi}_{n_1}^T \mathbf{I}_{n_1 \times n_d}).\end{aligned} \tag{4.22}$$

Clearly, $\mathbf{\Phi}_d$ is the first n_d^2 columns of $\mathbf{\Phi}_1$. Because our proposed algorithms are based on AMP, where each entry of the measurement matrix $\mathbf{A}$ follows an i.i.d. $\mathbb{N}(0, 1/m)$ distribution, it can be shown that, given $\mathbf{\Psi}_{n_1}$, each entry of $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_d$ also follows an i.i.d. $\mathbb{N}(0, 1/m)$ distribution. Therefore, with the proposed transform-domain up-/down-sampling method, the quality of the measurement matrix for the LR-AMP is the same as that of the HR-AMP.

4.4.2 Spatial-Domain Downsampling and Upsampling

We next develop two pairs of spatial-domain up-/down-sampling matrices for MR-AMP. In this part, we assume that images are piecewise constant and belong to the family $\mathbb{F}_{n_1,\varepsilon_1}^{PC}$ in Eq. (4.3), which has a small number of change points.

Solution 1

We first design the operators for 1D signals and then extend them to 2D images. For 1D piecewise constant signals, to satisfy Cond. 4.1.3, the first downsampling matrix $\mathbf{D}_d$ we use is the row-decimated identity matrix, *i.e.*, a matrix whose (i, di)-th entries are 1 for all i, and all other entries are zero. The downsampled signal can be written as

$$\mathbf{x}_d = \mathbf{D}_d\mathbf{x} = \begin{bmatrix} x[d] & x[2d] & \dots & x[n_d d] \end{bmatrix}^T, \tag{4.23}$$

where $x[i]$ represents the i-th entry of $\mathbf{x}$.

The corresponding upsampling matrix $\mathbf{U}_d$ used in this part is the repetition operator, which duplicates each input sample d times.

$$\mathbf{U}_d = \begin{bmatrix} \mathbf{1}_{d\times 1} & & \\ & \ddots & \\ & & \mathbf{1}_{d\times 1} \end{bmatrix}, \tag{4.24}$$

where $\mathbf{1}_{d\times 1}$ is an all-one vector. Clearly, $\mathbf{D}_d$ and $\mathbf{U}_d$ satisfy $\mathbf{D}_d\mathbf{U}_d = \mathbf{I}$ in Cond. 4.1.1.

Next, we show that the spatial-domain up-/down-sampling matrices also satisfy Cond. 4.1.3.

Lemma 4.4.1. *If $\mathbf{x}$ is a piecewise constant signal generated from the family $\mathbb{F}_{n_1,\varepsilon_1}^{PC}$ in Eq. (4.3), then the downsampled signal $\mathbf{x}_d$ in Eq. (4.23) belongs to $\mathbb{F}_{n_d,\varepsilon_d}^{PC}$ with $\varepsilon_d \leqslant d\varepsilon_1$.*

Proof. An $n_1 \times 1$ piecewise constant signal $\mathbf{x}$ is sparse in the differential domain.

$$\mathbf{s}_1 = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \mathbf{x} \equiv \mathbf{\Psi}_{n_1}\mathbf{x} = \begin{bmatrix} x[2]-x[1] \\ x[3]-x[2] \\ \vdots \\ x[n_1]-x[n_1-1] \end{bmatrix}. \tag{4.25}$$

Similarly, the representation of the downsampling signal in the differential domain can be written as

$$\begin{aligned} \mathbf{s}_d &= \mathbf{\Psi}_{n_d}\mathbf{x}_d \\ &= \begin{bmatrix} x[2d]-x[d], & \ldots, & x[n_d d]-x[(n_d-1)d] \end{bmatrix}^T. \end{aligned} \tag{4.26}$$

Therefore, calculating the number of change points in $\mathbf{x}_d$ is equivalent to counting the number of nonzero entries in $\mathbf{s}_d$.

To facilitate the proof, we construct two new vectors $\mathbf{s}_1^* = \begin{bmatrix} x[1] & \mathbf{s}_1^T \end{bmatrix}^T$ and $\mathbf{s}_d^* = \begin{bmatrix} x[d] & \mathbf{s}_d^T \end{bmatrix}^T$, *i.e.*, adding the first entry of $\mathbf{x}$ and $\mathbf{x}_d$ to $\mathbf{s}_1$ and $\mathbf{s}_d$, respectively. If we add d consecutive entries of $\mathbf{s}_1^*$, we can obtain one entry of $\mathbf{s}_d^*$. For example, $(x[d+1]-x[d])+(x[d+2]-x[d+1])+\ldots+(x[2d]-x[2d-1]) = x[2d]-x[d]$. In matrix form, this means that $\mathbf{s}_d^* = \mathbf{U}_d^T\mathbf{s}_1^*$. If $\mathbf{x}$ is generated from $\mathbb{F}_{n_1,\varepsilon_1}^{PC}$, the maximum expected number of nonzero entries in $\mathbf{s}_1^*$ will be $n_1\varepsilon_1+1$ due to the extra $x[1]$ in it. According to $\mathbf{s}_d^* = \mathbf{U}_d^T\mathbf{s}_1^*$, the maximum expected number of nonzero entries in $\mathbf{s}_d^*$ is still $n_1\varepsilon_1+1$. This occurs when there is at most one nonzero entry in every d entries in $\mathbf{s}_1^*$; hence, $\varepsilon_d \le (n_1\varepsilon_1+1)/n_d = d\varepsilon_1 + 1/n_d \to d\varepsilon_1$ when $n_1 \to \infty$.

□

We next extend the above results to 2D images. The 2D $n_d \times n_d$ LR image $\mathbf{X}_d$ can be written as Eq. (4.19) with $\mathbf{D}_d$ in Eq. (4.23). If an image is piecewise constant, its 2D gradient is sparse, where the 2D gradient at each pixel is given by

$$(\nabla\mathbf{X})_{i,j} = [X_{i+1,j} - X_{i,j}, X_{i,j+1} - X_{i,j}]. \tag{4.27}$$

The number of change points in a 2D piecewise constant signal $\mathbf{X}$ equals the number of nonzero entries in $\nabla\mathbf{X}$, where $(\nabla\mathbf{X})_{i,j}$ is counted as one nonzero entry if one or two of its components are nonzero. Therefore, we can also vectorize the 2D $\nabla\mathbf{X}$ into a 1D vector and apply the method in the Appendix to prove the concavity in Corollary 4.3.3 for 2D piecewise constant signals. Additionally, the vertical differences and the horizontal differences are disjoint. By Lemma 4.4.1, the number of horizontal or vertical change points of $\mathbf{X}_d$ is no larger than that of $\mathbf{X}$; thus, Cond. 4.1.3 is true for 2D images.

The remaining problem is to choose the appropriate denoiser for 2D piecewise constant signals. In this paper, instead of using the denoisers discussed in [70,90], such as NLM (non-local means) and BM3D (3D block matching), we use a 2D-TV-based denoiser in $\eta_{\sigma_d^t}(\mathbf{z}_d^t)$ of Eq. (4.9). Our method is denoted as AMP-TV-2D.

The TV norm of 2D piecewise constant signals is defined as

$$\|\mathbf{X}\|_{\mathrm{TV}} = \sum_{i,j} \sqrt{|X_{i+1,j} - X_{i,j}|^2 + |X_{i,j+1} - X_{i,j}|^2}, \tag{4.28}$$

which is isotropic and un-differentiable. This norm will be used by the 2D-TV-based denoiser. Additional details are given in Sec. 4.5.1. This is different from the 1D TV denoiser in [28], where the TV norm for 1D piecewise constant signals is written as $\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{i=1}^{n_1-1} |x_{i+1} - x_i|$.

In Sec. 4.5.4, we compare the performance of our AMP-TV-2D with the state-of-the-art algorithm TVAL3 (TV minimization by Augmented Lagrangian and ALternating direction ALgorithms) in [53], [64]. Note that TVAL3 depends on two slack parameters, which have to be manually tuned for each image and each measurement rate. In contrast, the thresholding parameters in our AMP-TV-2D are automatically tuned in each iteration, as will be discussed in Sec. 4.5.1. Recently, an algorithm similar to our AMP-TV-2D, the dual-constraints AMP (DC-AMP) (Sec. 8.1 of [8]), was developed for 2D piecewise smooth signals and can achieve similar performance to TVAL3. However, it also includes a smoothness parameter that needs to be manually tuned. Moreover, no theoretical analysis of DC-AMP has been performed.

Given the spatial-domain up-/down-sampling matrices, to satisfy Cond. 4.1.2, *i.e.*, the quality of the measurement matrix for LR-AMP is no worse than that of HR-AMP, we need to normalize the measurement matrix for LR-AMP, *i.e.*,

$$\mathbf{\Phi}_d = \frac{1}{d}\mathbf{A}(\mathbf{U}_d \otimes \mathbf{U}_d), \tag{4.29}$$

such that each entry of $\mathbf{\Phi}_d$ follows an i.i.d. $\mathbb{N}(0, 1/m)$ distribution.

Solution 2

In addition to the simple up-/down-sampling matrices in Eq. (4.23) and Eq. (4.24), we also develop a pair of bicubic up/-downsampling matrices and evaluate them in Sec. 4.5.4. In bicubic downsampling, each pixel in the LR image is the weighted average of sixteen pixels in the HR image, which has been known to produce smoother LR images than Eq. (4.23), *i.e.*, with fewer change points in $\mathbf{X}_d$. Therefore, Cond. 4.1.3 holds for bicubic downsampling. On the other hand, the upsampling first inserts $d-1$ zeros between neighboring samples of the LR image and then performs bicubic interpolation. However, it can be verified that the corresponding product $\mathbf{D}_d\mathbf{U}_d$ is not an identity matrix, although it is very close. Therefore, strictly speaking, Cond. 4.1.2 does not hold for bicubic matrices, and the simple scaling matrix $\mathbf{\Lambda}$ cannot make each entry of $\mathbf{\Phi}_d$ exactly follow an i.i.d. $\mathbb{N}(0, 1/m)$ distribution. Nevertheless, this remains approximately true, and the efficiency of this scheme will be verified empirically in Sec. 4.5.4. Moreover, according to Corollary 4.3.4, the condi-

tional upper bound of the noise sensitivity is proportional to the LR approximation error $\|(\mathbf{I}-\mathbf{U}_d\mathbf{D}_d)\mathbf{x}\|_2^2$. Therefore, for images, in terms of LR approximation error, the bicubic up-/down-sampling matrices remain better than the simple matrices in Eq. (4.23) and Eq. (4.24).

Finally, we note the differences with our methods compared to the methods in [53, 107]. In [53], a similar spatial-domain up-/down-sampling framework was proposed; however, the proof in the reference was implicit. In addition, TVAL3 was chosen as the reconstruction algorithm, which requires manual tuning of two parameters. Moreover, the reconstruction performance cannot be predicted. Our AMP-TV-2D does not include a manually tuned parameter, and its performance can be accurately predicted via state evolution. In [107], the same piecewise constancy model and up-/down-sampling matrices as in Eq. (4.23) and (4.24) are used. The algorithm first reconstructs the original HR image and uses this estimated HR image to reduce the approximation error $\mathbf{A}(\mathbf{I}-\mathbf{U}_d\mathbf{D}_d)\mathbf{x}$. However, there is no theoretical guarantee that such an operation can reduce the approximation error, and the algorithm only works when the undersampling rate δ_1 is sufficiently large, at least larger than 10%. Moreover, the complexity of this approach is higher than reconstructing the LR image directly.

4.5 Experimental Results

In this section, we demonstrate the performance of the proposed MR-AMP with both transform- and spatial-domain up-/down-sampling, denoted by AMP-ST (soft thresholding) and AMP-TV (total variation), respectively. The empirical results will also be shown to verify some theoretical results. In each method, to facilitate comparison with the conventional approach, we use LR-AMP-ST and LR-AMP-TV to denote the proposed LR reconstruction schemes and HR-AMP-ST and HR-AMP-TV to denote the original AMP with HR reconstruction. In addition, H2L-AMP-ST and H2L-AMP-TV represent the naive solutions that are first used to reconstruct the HR signal and then downsample to the LR signal.

All tests in this paper use a column-normalized i.i.d. Gaussian measurement matrix $\mathbf{A}$. All simulations are conducted on a PC with a 3.4 GHz Intel Core i7 quad-core processor and 64 GB of memory. The utilized testing images include the popular images Lena, Barbara, Boat, House, and Peppers as well as some land remote sensing images, including Memorial Stadium at the University of Nebraska-Lincoln (Cornhuskers) and Sea World in San Diego. We follow the setup in [70] to rescale all images to 128×128. This enables the entire measurement matrix $\mathbf{A}$ to be stored in memory. We also include some experiments of larger 256×256 images to demonstrate the visual comparison, following the same setup in [70]. We have posted our code online[1].

[1] https://github.com/xingwangsfu/MR-AMP

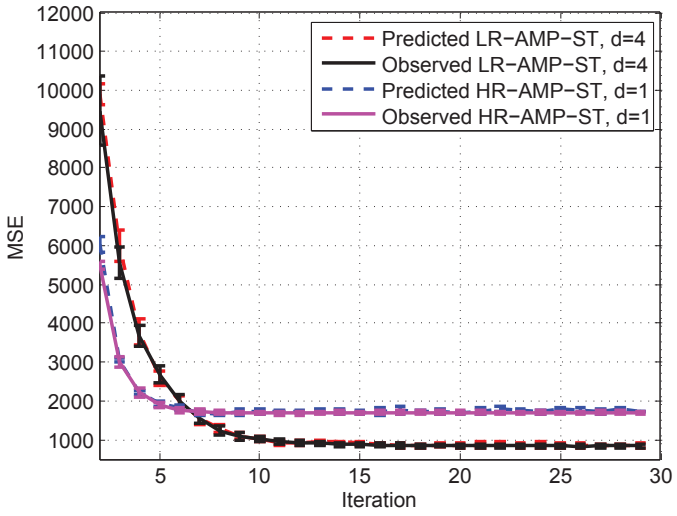


Figure 4.1: Empirical intermediate MSE and predicted state evolution of HR-AMP-ST and LR-AMP-ST for the Barbara image with $d = 4$.

4.5.1 Parameter Tuning

One of the main challenges in implementing different MR-AMP algorithms is the tuning of each algorithm's free parameters. Many techniques exist to estimate the noise variance in an image. In this paper, we use the following convenient feature of AMP algorithms: $\|\mathbf{r}_d^t\|_2^2/m \approx (\sigma_d^t)^2$ [68].

For MR-AMP-ST, we set its threshold using three methods. For the 1D synthetic examples in Sec. 4.5.3, we assume that the sparsity rate is known and set the thresholding parameter according to the minimax rule in [31]. For the 2D imaging examples in Sec. 4.5.4, because images are not exactly sparse in the transform domain, we have to estimate the sparsity rate. For a sufficiently large CS undersampling rate δ_1, such as 10% and 20%, we use the SURE (Stein's unbiased risk estimate)-based method in [73] to decide on the thresholding parameter in each iteration. For very small δ_1, such as 3% and 4%, SURE does not perform well because it is based on a large system limit. We choose the max-min optimal threshold as determined by [69].

For AMP-TV, we use different tuning methods for 1D and 2D signals. For 1D signals, we use the source code from [55] directly. For 2D images, there are many methods for adaptively choosing the regularization parameter in TV-based image denoising, *e.g.*, [48] and [38]. In this paper, we use Algorithm 6 in [38] due to its simplicity and efficiency. In each iteration of AMP-TV-2D, a Lagrangian optimization problem whose constraint is the TV of the solution is solved, and the Lagrangian parameter can be adaptively determined analytically.

In AMP-ST, the Onsager term is obtained by Eq. (4.1) in [32]. For AMP-TV-1D, the Onsager term is calculated by Eq. (5.11) in [28]. For AMP-TV-2D, it is difficult to obtain

an exact expression of the divergence. We thus apply the Monte Carlo method in [70] to find a good approximation of the divergence.

4.5.2 State Evolution in MR-AMP

In this part, we compare the predicted and observed performances of MR-AMP with different denoisers. Recall that the state evolution of MR-AMP is given in Eq. (4.11). To compute this value, at every iteration, we add white Gaussian noise with standard deviation σ_d^t to $\mathbf{x}_{d,o}$, denoise the signal with denoiser $\eta_{\sigma_d^t}(:,\tau)$, and then compute the MSE.

Fig. 4.1 compares the empirical MSE and predicted state evolution of MR-AMP-ST for the test image Barbara with a size 128×128, with DCT being the sparsifying basis. It can be observed that the state evolution is quite accurate. Moreover, the converged MSE per entry of the LR image is approximately 50% smaller than that of the HR image, which verifies the motivation of this paper, *i.e.*, we can recover an LR signal with smaller MSE when the MSE of the HR signal is too high. Note that the LR reference image is obtained via the DCT-domain downsampling in Sec. 4.4.1, and the corresponding MSE is the MSE between the LR image reconstructed by LR-AMP-ST and the LR reference image.

Fig. 4.2 shows the state evolution performance of MR-AMP-TV. Two different upsampling matrices are compared: the repetition interpolator in Eq. (4.24) (MR-AMP-TV-2D-R) and the bicubic interpolator (MR-AMP-TV-2D-B). The reference LR image is obtained using Matlab's imresize($\mathbf{x}$,1/d) command with a bicubic interpolator. There is a near-perfect correspondence between the predicted and true MSEs for the repetition interpolation. For the bicubic interpolator, a slight mismatch exists because the entries of the new measurement matrix are not exactly independent. The figures also show that a lower resolution provides a smaller MSE, and the bicubic interpolator outperforms the repetition operator.

Note that the denoiser in the AMP-TV-2D is essentially a non-scalar denoiser, similar to [8, 70, 90]. Although the state evolution for AMP with non-scalar denoisers has not been proved rigorously, the results in Fig. 4.2 suggest that the state evolution derived in our paper is quite accurate.

4.5.3 Performance with Synthetic 1D Signals

In this part, we demonstrate the performance of the proposed scheme for synthetic 1D signals, which can verify the theoretical noiseless phase transition curve (PTC) and noise sensitivity.

Transform Domain Approach

To obtain the empirical noiseless PTC of HR-AMP-ST, we fix $n_1 = 2000$ and take 30 equally spaced values of $\delta_1 = m/n_1$ in the range of $[0.05, 0.95]$ and 30 equally spaced values of $\rho_1 = k_1/m$ in $[0.05, 0.95]$. For each combination of (δ_1, ρ_1), a 1D Bernoulli-Gaussian

γ	HR-AMP-ST Bound	HR-AMP-ST Empirical	LR-AMP-ST Bound	LR-AMP-ST Empirical
0.95	3.80	3.06	5.23	2.78
0.98	9.80	7.47	5.23	4.12
0.99	19.80	14.62	5.23	4.15
0.998	39.80	28.89	5.23	4.79

Table 4.1: Noise sensitivity of MR-AMP-ST with $\delta_1 = 0.2$ and $\rho_1 = 0.3$.

signal and its CS samples are generated before applying the HR-AMP-ST. The empirical PTCs are obtained by connecting operating points with a 50% signal recovery success rate, where the recovery is considered successful when the normalized MSE (NMSE) satisfies $\|\mathbf{x}_o - \hat{\mathbf{x}}\|_2^2 / \|\mathbf{x}_o\|_2^2 \leqslant 10^{-6}$.

To study the empirical noiseless PTC of LR-AMP-ST, we generate a special $n_1 \times 1$ sparse signal, whose first $n_d = n_1/d$ entries are Bernoulli-Gaussian distributed, and all other entries are 0. According to Eq. (4.5), the truncation operator does not introduce any approximation error $\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$. We then run the HR-AMP-ST and LR-AMP-ST algorithms to recover the target HR and LR signals respectively. Note that we are interested in the case $m/n_1 < 1/d$; otherwise, the setup is no longer a CS problem. Although the procedure for generating the HR signal here is different from that in the above simulation of empirical HR-AMP-ST, both signals belong to the same class of probability distributions if the numbers of nonzero coefficients are the same, and the experimental results show that these two empirical PTCs for HR-AMP-ST coincide with each other.

The theoretical noiseless PTC in Eq. (4.14) and the empirical noiseless PTC of LR-AMP-ST are shown in Fig. 4.3 for simple sparse signals with different d. The two sets of curves agree perfectly. It can be shown that, as d increases, the PTC curve shifts to the left, which means that the LR-AMP can recover the signal even when the HR-AMP fails.

The example above does not have an approximation error. Next, we construct a special case to show that the noise sensitivity of HR-AMP-ST is unbounded above the PTC, whereas the noise sensitivity of the LR-AMP-ST remains bounded. The setup is similar to that in [32], where a special 3-point distribution of $\mathbf{x}$ is constructed in Lemma 4.4, whose MSE above the phase transition boundary is given by $\delta_1\gamma/(1-\gamma)$. Therefore, the MSE can go to infinity when γ is close to 1. We present the noise sensitivity of MR-AMP-ST in Table 4.1 with $n_1 = 2000$, $\delta_1 = 0.2$, $\rho_1 = 0.3$ and $\sigma_w^2 = 1$. As shown in Fig. 4.3, this setup is above the PTC of $d = 1$ but below the PTC of $d = 2$. The non-zero locations of $\mathbf{x}$ are chosen with probability $1.8\varepsilon_1$ from the first n_2 entries to generate the 3-point distribution and with probability $0.2\varepsilon_1$ to generate Bernoulli-Gaussian signals for the second n_2 entries to fix the approximation error in Eq. (4.9) for different γs. We then apply HR-AMP-ST and LR-AMP-ST to reconstruct $\mathbf{x}$ and $\mathbf{x}_d$.

It can be observed from Table 4.1 that, as γ approaches 1, the noise sensitivity bound of HR-AMP-ST continues increasing; however, the noise sensitivity bound of LR-AMP-ST is stable because all parts in Eq. (4.15) are fixed. This verifies the advantage of our LR-AMP. The empirical results of both methods are also below their noise sensitivity bounds.

Spatial Domain Approach

It is difficult to reproduce the theoretical noiseless PTC of HR-AMP-TV-1D in [28] because it relies on complicated numerical optimization, and no open source code is available. Instead, we study the empirical noiseless PTC of HR-AMP-TV-1D by replicating an experiment from [55] using their source code. We fix $n_1 = 628$ and consider a 30×30 uniform grid in the range of $\delta_1 = m/n_1 \in [0.05, 0.95]$ and $\rho_1 = k_1/m \in [0.05, 0.95]$. The corresponding HR Bernoulli-Gaussian 1D finite-difference signal is then generated. The empirical noiseless PTC of HR-AMP-TV-1D is shown in Fig. 4.4 (a) (with $d = 1$).

To obtain the empirical noiseless PTCs of LR-AMP-TV-1D, we first generate the LR signal $\mathbf{x}_d$ that yields a 1D Bernoulli-Gaussian finite-difference sequence with sparsity rate $d\varepsilon_1$. We then duplicate each entry d times to obtain the HR piecewise constant signal with sparsity rate ε_1 according to $\mathbf{D}_d$ and $\mathbf{U}_d$ in Eq. (4.23) and (4.24). From the analysis in Sec. 4.4.2, the approximation error is zero. Successful recovery is declared when the NMSE is less than 10^{-4}. The results with $d = 2$ and $d = 4$ are also shown in Fig. 4.4 (a).

To study the noise sensitivity of MR-AMP-TV-1D, we recover the target HR and LR piecewise constant signals after introducing additional white Gaussian noise (AWGN) with $\text{SNR} \triangleq \|\mathbf{Ax}\|_2^2 / \|\mathbf{w}\|_2^2 = 60dB$ in the measurement. Fig. 4.4 (b) shows the median NSNR defined as $\text{NSNR} \triangleq \|\mathbf{x}_o\|_2^2 / \|\mathbf{x}_o - \hat{\mathbf{x}}\|_2^2$ versus the sampling ratio $\delta_1 = m/n_1$ at the fixed sparsity rate $\varepsilon_1 = 0.05$, as in [14]. This shows that the LR-AMP-TV-1D obtains a lower NMSE than does the HR-AMP-TV-1D. This verifies Corollary 4.3.4, *i.e.*, the LR reconstruction obtains a better performance than the HR reconstruction.

4.5.4 Performance with 2D Images

In this part, we apply the MR-AMP theory to MR 2D image reconstruction. All reported experimental results are the averages of 20 Monte Carlo simulations.

Target LR image

The target LR images are different when different downsampling matrices are used. For the transform-domain approach, the target LR image $\mathbf{X}_d$ is represented by Eq. (4.19). Both DCT and the Daubechies-8 (D8) wavelet are tested. For the spatial-domain approach, although the simple matrix in Eq. (4.23) can be applied, we choose to use the bicubic downsampling matrix because it leads to a better LR image. As discussed before, Cond. 4.1.3 still holds in this case. Given the bicubic downsampling matrix, we test the repetition

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	5%	HR-AMP-ST	16.75	15.96	17.60	18.21	15.53	15.86	14.39
		H2L-AMP-ST	17.40	16.48	18.30	18.58	15.93	16.49	15.10
		LR-AMP-ST	**18.02**	**17.11**	**18.77**	**19.13**	**16.68**	**16.89**	**15.33**
	10%	HR-AMP-ST	18.50	17.79	18.94	19.71	17.35	16.95	15.17
		H2L-AMP-ST	19.43	18.56	19.97	20.34	18.19	18.05	16.10
		LR-AMP-ST	**20.82**	**19.94**	**21.07**	**21.72**	**19.43**	**18.71**	**16.79**
	20%	HR-AMP-ST	21.28	20.36	21.08	22.31	19.93	18.69	16.60
		H2L-AMP-ST	22.58	21.69	22.61	23.46	21.27	20.13	18.06
		LR-AMP-ST	**24.90**	**24.25**	**24.46**	**26.34**	**23.76**	**21.89**	**19.72**
4	3%	HR-AMP-ST	15.37	14.76	16.54	17.09	14.33	14.96	13.51
		H2L-AMP-ST	16.98	16.33	18.55	18.86	15.91	**17.24**	**15.97**
		LR-AMP-ST	**18.22**	**17.66**	**18.92**	**19.58**	**17.02**	17.13	15.59
	4%	HR-AMP-ST	16.03	15.24	16.95	17.56	14.87	15.27	13.75
		H2L-AMP-ST	17.81	16.91	19.09	19.46	16.65	**17.71**	**16.39**
		LR-AMP-ST	**19.21**	**18.42**	**19.63**	**20.31**	**17.78**	17.66	15.76
	5%	HR-AMP-ST	16.52	15.74	17.24	17.97	15.91	15.52	13.95
		H2L-AMP-ST	18.44	17.54	19.57	20.04	17.29	**18.13**	**16.72**
		LR-AMP-ST	**19.66**	**18.90**	**19.67**	**20.60**	**18.45**	17.48	15.72

Table 4.2: PSNRs (dB) of 128×128 image reconstructions with DCT-domain MR-AMP-ST.

upsampling matrix in Eq. (4.24) as well as the bicubic upsampling matrix. It can be verified that Cond. 4.1.1 $\mathbf{D}_d\mathbf{U}_d = \mathbf{I}$ holds approximately between these two upsampling matrices and the bicubic downsampling matrix.

In this paper, we use the Peak SNR (PSNR) to measure the objective quality of a reconstructed image, which is defined as $10\log_{10}(255^2/\text{MSE}(\mathbf{X} - \hat{\mathbf{X}}))$, where $\mathbf{X}$ is the reference image and $\hat{\mathbf{X}}$ is the test image.

Scaling Matrix $\mathbf{\Lambda}$

During the reconstruction of the LR image, to ensure that Cond. 4.1.2 in Sec. 4.3 is satisfied, we need to scale its corresponding measurement matrix $\mathbf{A}\mathbf{U}_d$ into $\mathbf{A}_d = \mathbf{A}\mathbf{U}_d\mathbf{\Lambda}$ to obtain normalized columns, as shown in Eq. (4.22) and Eq. (4.29). Because no specific entries in the target LR image are preferred, the scaling matrix $\mathbf{\Lambda}$ should be a diagonal matrix with equal diagonal entries. For LR-AMP-ST in the DCT and wavelet domain, the diagonal entry is the inverse of the downsampling factor d, according to Eq. (4.22). For LR-AMP-TV-2D in the TV domain, things are slightly different. For the repetition operator that replaces each pixel in the LR image with a $d \times d$ block of pixels in the HR image, the diagonal entry in the scaling matrix remains $1/d$. For bicubic interpolation, we empirically set the diagonal entry in the scaling matrix to be $1/2.68$ for $d = 2$ and $1/5$ for $d = 4$. Although this approach cannot exactly normalize the columns and although there remain some correlations between entries in the new measurement matrix, the approach works quite well in practice.

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	5%	HR-AMP-ST	16.58	15.84	17.81	17.66	15.31	15.94	14.33
		H2L-AMP-ST	16.85	16.46	18.55	18.30	15.83	16.86	15.04
		LR-AMP-ST	**17.35**	**17.01**	**19.13**	**18.95**	**16.67**	**17.28**	**15.38**
	10%	HR-AMP-ST	18.20	17.47	19.29	19.56	17.14	16.99	15.05
		H2L-AMP-ST	19.13	18.25	20.53	20.43	18.05	18.18	15.98
		LR-AMP-ST	**20.62**	**19.72**	**21.46**	**21.97**	**19.46**	**19.02**	**16.79**
	20%	HR-AMP-ST	21.27	20.15	21.62	22.68	20.04	18.86	16.60
		H2L-AMP-ST	22.98	21.59	23.53	24.47	21.61	20.64	18.11
		LR-AMP-ST	**24.98**	**23.89**	**25.02**	**26.44**	**23.51**	**22.05**	**19.80**
4	3%	HR-AMP-ST	15.14	14.67	16.66	16.41	14.26	15.01	13.54
		H2L-AMP-ST	16.83	16.40	18.90	18.31	15.95	**17.45**	**16.19**
		LR-AMP-ST	**17.83**	**17.24**	**19.31**	**19.28**	**16.93**	17.33	15.52
	4%	HR-AMP-ST	15.62	15.16	17.09	17.08	14.70	15.40	13.69
		H2L-AMP-ST	17.47	17.04	19.53	19.21	16.63	**18.07**	**16.50**
		LR-AMP-ST	**18.73**	**18.09**	**19.70**	**19.78**	**17.60**	17.74	15.78
	5%	HR-AMP-ST	15.99	15.58	17.50	17.52	15.19	15.66	13.90
		H2L-AMP-ST	17.99	17.63	**20.18**	19.81	17.25	**18.51**	**16.87**
		LR-AMP-ST	**19.00**	**18.47**	19.65	**20.14**	**17.84**	17.55	15.60

Table 4.3: PSNRs (dB) of 128×128 image reconstructions with wavelet-domain MR-AMP-ST.

Noiseless image recovery

Tables 4.2, 4.3 and 4.4 compare the performances of DCT-domain MR-AMP-ST, wavelet-domain MR-AMP-ST, and spatial-domain MR-AMP-TV when there is no measurement noise. In each case, we compare our proposed LR-AMP, which recovers the LR image directly; the conventional HR-AMP, which reconstructs the HR image; and the naive H2L-AMP, which recovers the HR image first and then downsamples it to obtain the LR image with the corresponding downsampling matrix. The highest PSNR in each case is highlighted.

From Tables 4.2 and 4.3, we can see that LR-AMP-ST almost always outperforms the other two algorithms, except when $d = 4$ for HuskerStadium and SeaWorld. This is partially due to two reasons. First, land remote sensing images contain more details compared to natural images. Second, the suboptimal thresholding rule in [69] is used for $d = 4$, whereas the optimal SURE-based thresholding method in [73] is used for $d = 2$.

In the spatial-domain approach, LR-AMP-TV-2D-B and H2L-AMP-TV-2D are the top two algorithms. Their reconstruction performances are comparable, and the PSNR difference between them is less than 1 dB. However, H2L-AMP-TV-2D is much slower than the proposed LR-AMP-TV-2D, as will be detailed in the computational complexity part. Because the reference HR image is the same for the three approaches listed in Tables 4.2, 4.3 and 4.4, it can be observed that the TV-based approach yields higher PSNR than do the transform-domain approaches.

Fig. 4.5 and Fig. 4.6 illustrate the visual quality of the recovered 256×256 Barbara and Stadium images under different methods. It can be observed that transform-domain and

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	5%	HR-AMP-TV-2D	20.88	19.66	20.67	22.85	19.60	18.36	16.13
		H2L-AMP-TV-2D	22.55	21.09	22.51	24.53	21.07	20.27	**17.93**
		LR-AMP-TV-2D-R	21.79	20.25	22.17	23.80	20.13	19.95	17.75
		LR-AMP-TV-2D-B	**22.68**	**21.17**	**22.67**	**25.07**	**21.13**	**20.33**	17.87
	10%	HR-AMP-TV-2D	23.62	22.18	22.80	26.55	22.51	20.20	17.64
		H2L-AMP-TV-2D	**25.83**	24.08	**25.35**	**28.91**	**24.63**	**22.70**	**19.97**
		LR-AMP-TV-2D-R	23.83	22.32	24.07	26.42	22.31	21.64	19.16
		LR-AMP-TV-2D-B	25.66	**24.18**	25.24	28.82	24.33	22.63	19.95
	20%	HR-AMP-TV-2D	26.51	25.05	25.02	30.91	25.70	22.15	19.31
		H2L-AMP-TV-2D	**29.49**	**27.65**	**28.39**	**34.11**	**28.49**	**25.38**	22.26
		LR-AMP-TV-2D-R	26.24	24.83	26.19	29.23	24.60	23.60	20.85
		LR-AMP-TV-2D-B	28.92	27.63	27.99	32.44	27.51	25.28	**22.34**
4	3%	HR-AMP-TV-2D	18.69	17.90	18.89	20.32	17.71	16.77	15.08
		H2L-AMP-TV-2D	21.75	**20.80**	22.29	23.55	**20.70**	20.26	18.71
		LR-AMP-TV-2D-R	20.73	19.63	22.10	23.59	19.33	20.22	18.80
		LR-AMP-TV-2D-B	**21.95**	20.49	**23.02**	**24.47**	20.46	**20.92**	**19.22**
	4%	HR-AMP-TV-2D	19.89	18.89	19.84	21.64	18.83	17.66	15.65
		H2L-AMP-TV-2D	**23.43**	**22.21**	23.77	25.39	**22.26**	21.62	19.75
		LR-AMP-TV-2D-R	21.53	20.28	22.75	24.19	20.31	20.81	19.35
		LR-AMP-TV-2D-B	22.99	21.54	**23.93**	**25.55**	21.66	**21.64**	**19.93**
	5%	HR-AMP-TV-2D	20.88	19.66	20.67	22.85	19.60	18.36	16.13
		H2L-AMP-TV-2D	**24.86**	**23.24**	**25.07**	**27.05**	**23.40**	**22.74**	**20.49**
		LR-AMP-TV-2D-R	22.24	20.86	23.41	25.09	20.75	21.51	19.66
		LR-AMP-TV-2D-B	23.97	22.32	24.80	26.58	22.52	22.54	20.30

Table 4.4: PSNRs (dB) of 128×128 image reconstructions with spatial-domain MR-AMP-TV-2D.

spatial-domain approaches produce different types of reconstruction artifacts. The former approach preserves more details but also contains more high-frequency noises, whereas the latter approach is blockier, despite the higher PSNRs.

Comparison between AMP-TV-2D-B and optimal TVAL3 [53, 64]

In Table 4.5, we compare the results of TVAL3 with optimized slack parameters [53, 64] and our parameter-free AMP-TV-2D-B for the MR-CS problem in Eq. (4.5). For the original HR image reconstruction, the performance of HR-AMP-TV-2D-B is comparable to the optimized HR-TVAL3. However, for the LR image reconstruction, our LR-AMP-TV-2D outperforms the optimized LR-TVAL3 in almost all cases by up to 1 dB. More importantly, the theoretical analyses developed in Sec. 4.3 and 4.4 are applicable for MR-AMP-TV, whereas there are only some qualitative analyses in [53].

Comparison between LR-AMP-TV-2D-B and [107]

The authors in [107] study a similar problem as ours and modify the sampled data to reduce the approximation error level $\mathbf{A}(\mathbf{I} - \mathbf{U}_d\mathbf{D}_d)\mathbf{x}$ in Eq. (4.5). For a fair comparison, we change the bicubic downsampling matrix in the previous parts to the decimation downsampling matrix in Eq. (4.23) used in [107] and choose AMP-TV-2D as the reconstruction algorithm

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	10%	HR-AMP-TV-2D	**23.62**	**22.18**	22.80	**26.55**	**22.51**	20.20	17.64
		HR-TVAL3	23.36	21.80	**22.94**	26.21	21.89	**20.32**	**17.71**
		LR-AMP-TV-2D-B	**25.66**	**24.18**	**25.24**	**28.82**	**24.33**	**22.63**	**19.95**
		LR-TVAL3	25.44	24.05	25.09	28.60	23.98	21.94	19.24
	20%	HR-AMP-TV-2D	26.51	25.05	25.02	30.91	**25.70**	22.15	19.31
		HR-TVAL3	**26.80**	**25.22**	**25.49**	**31.79**	25.65	**22.42**	**19.62**
		LR-AMP-TV-2D-B	**28.92**	**27.63**	**27.99**	**32.44**	**27.51**	**25.28**	**22.34**
		LR-TVAL3	28.58	27.45	27.59	31.67	27.02	24.19	21.38
4	4%	HR-AMP-TV-2D	**19.89**	**18.89**	19.84	**21.64**	**18.83**	17.66	15.65
		HR-TVAL3	19.69	18.77	**20.21**	21.43	18.48	**18.21**	**16.02**
		LR-AMP-TV-2D-B	**22.99**	**21.54**	**23.93**	**25.55**	**21.66**	**21.64**	**19.93**
		LR-TVAL3	22.77	21.31	23.85	25.19	21.56	20.95	19.80
	5%	HR-AMP-TV-2D	**20.88**	**19.66**	20.67	**22.85**	**19.60**	18.36	16.13
		HR-TVAL3	20.58	19.29	**20.94**	22.57	19.15	**18.70**	**16.38**
		LR-AMP-TV-2D-B	**23.97**	**22.32**	**24.80**	**26.58**	**22.52**	**22.54**	20.30
		LR-TVAL3	23.80	21.85	24.40	26.15	22.46	21.45	**20.45**

Table 4.5: Comparison of the final reconstruction results in PSNR between TVAL3 and AMP-TV-2D.

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	5%	LR-AMP-TV-2D-B	**20.88**	**19.83**	**20.92**	**23.34**	**19.56**	**18.55**	**16.26**
		[107]	19.59	18.53	20.17	21.77	18.25	18.00	16.03
	10%	LR-AMP-TV-2D-B	22.79	**21.77**	22.34	25.64	**23.31**	**19.96**	**17.48**
		[107]	**22.81**	21.25	**22.47**	**26.26**	21.29	19.95	17.31
4	3%	LR-AMP-TV-2D-B	**18.34**	**17.48**	**19.01**	**20.98**	**16.84**	**17.42**	**15.27**
		[107]	16.90	16.39	18.31	19.54	15.76	16.50	14.79
	4%	LR-AMP-TV-2D-B	**18.67**	**17.77**	**19.28**	**21.48**	**17.17**	**17.78**	**15.47**
		[107]	17.86	17.14	18.96	20.51	16.64	17.09	15.27

Table 4.6: Comparison of the final reconstruction results in PSNR between LR-AMP-TV-2D-B and [107].

for [107]. For the upsampling matrix, the duplication upsampling matrix in Eq. (4.24) is used in [107], whereas we still use the bicubic upsampling matrix here. In Table 4.6, we compare the results of AMP-TV-2D-B with the algorithm proposed in [107]. The approach in [107] only works when δ_1 is sufficiently large, e.g., δ_1 should be at least greater than 10% for $d = 2$. Moreover, there is no theoretical guarantee involved in [107].

Imaging in the presence of measurement noise

Table 4.7 shows the performance of MR-AMP in different domains when various amounts of measurement noise are added. The proposed LR-AMP still outperforms the HR-AMP and H2L-AMP in almost all cases.

AWGN with a standard deviation of 20										
DCT	d = 2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-ST	16.00	17.70	19.92		HR-AMP-ST	14.74	15.22	15.73
		H2L-AMP-ST	16.45	18.43	21.08		H2L-AMP-ST	16.32	16.90	17.53
		LR-AMP-ST	**17.12**	**19.65**	**22.88**		LR-AMP-ST	**17.56**	**18.37**	**18.71**
Wavelet	d=2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-ST	15.80	17.47	19.64		HR-AMP-ST	14.65	15.14	15.56
		H2L-AMP-ST	16.44	18.33	21.02		H2L-AMP-ST	16.38	17.04	17.61
		LR-AMP-ST	**16.85**	**19.56**	**22.79**		LR-AMP-ST	**17.20**	**17.98**	**18.29**
TV	d=2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-TV-2D	19.59	21.84	23.93		HR-AMP-TV-2D	17.82	18.85	19.59
		H2L-AMP-TV-2D	21.00	23.68	26.24		H2L-AMP-TV-2D	**20.72**	**22.13**	**23.09**
		LR-AMP-TV-2D-R	20.20	22.08	24.17		LR-AMP-TV-2D-R	19.60	20.25	20.80
		LR-AMP-TV-2D-B	**21.05**	**23.70**	**26.31**		LR-AMP-TV-2D-B	20.45	21.48	22.22
AWGN with a standard deviation of 40										
DCT	d = 2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-ST	15.89	17.37	19.06		HR-AMP-ST	14.67	15.16	15.59
		H2L-AMP-ST	16.34	18.03	19.94		H2L-AMP-ST	16.26	16.96	17.43
		LR-AMP-ST	**17.00**	**19.11**	**21.21**		LR-AMP-ST	**17.41**	**18.09**	**18.24**
Wavelet	d=2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-ST	15.68	17.19	18.72		HR-AMP-ST	14.61	15.08	15.42
		H2L-AMP-ST	16.21	17.87	19.75		H2L-AMP-ST	16.37	17.00	17.49
		LR-AMP-ST	**16.81**	**18.95**	**21.10**		LR-AMP-ST	**17.00**	**17.76**	**17.93**
TV	d=2	δ_1	5%	10%	15%	d=4	δ_1	3%	4%	5%
		HR-AMP-TV-2D	19.36	21.13	22.52		HR-AMP-TV-2D	17.75	18.70	19.36
		H2L-AMP-TV-2D	20.70	22.80	**24.51**		H2L-AMP-TV-2D	**20.62**	**21.94**	**22.75**
		LR-AMP-TV-2D-R	20.02	21.58	23.05		LR-AMP-TV-2D-R	19.55	20.16	20.61
		LR-AMP-TV-2D-B	**20.73**	**22.81**	24.44		LR-AMP-TV-2D-B	20.35	21.31	21.91

Table 4.7: PSNRs (dB) of the reconstruction of the 128×128 Barbara image with varying amounts of additive Gaussian measurement noise.

LR approximation

Another important problem in MR-CS is how to use an LR image recovered by LR-AMP to facilitate the reconstruction of a higher resolution image. As an initial attempt, we show in Table 4.8 some results obtained by simply upsampling the recovered LR image with the upsampling matrix to obtain an HR image, named L2H-AMP. As shown by the table, even this simple method can sometimes provide better HR images than HR-AMP. For example, L2H-AMP-ST can outperform HR-MP-ST in almost all cases. However, HR-AMP-TV-2D-B outperforms L2H-AMP-TV-2D-B when $d = 4$ and $\delta_1 = 0.05$, which implies that L2H-AMP is far from optimal. This is because high-frequency information can be captured in CS measurements $\mathbf{y}$; however, L2H-AMP is based on LR-AMP. It thus treats the high-frequency information as approximation errors, and the upsampling matrix cannot estimate such information from LR images.

Computational complexity

The computational complexities of various methods are reported in Table 4.9, which shows that when $d = 2$, the proposed LR-AMP is approximately 2 times faster than the HR-AMP

d	δ_1	Algorithm	Lena	Barbara	Boat	House	Peppers	HuskerStadium	SeaWorld
2	10%	HR-AMP-ST	18.50	17.79	18.94	19.71	17.35	16.95	15.17
		L2H-AMP-ST	**20.14**	**19.46**	**20.18**	**21.25**	**19.02**	**18.03**	**16.07**
		HR-AMP-TV-2D-B	23.62	22.18	22.80	**26.55**	**22.51**	20.20	17.64
		L2H-AMP-TV-2D-B	**23.65**	**22.52**	**22.86**	26.21	22.30	**20.25**	**17.79**
4	5%	HR-AMP-ST	16.52	15.74	17.24	17.97	15.91	15.52	**13.95**
		L2H-AMP-ST	**18.76**	**17.85**	**19.05**	**20.00**	**17.65**	**16.38**	11.93
		HR-AMP-TV-2D-B	**20.88**	**19.66**	**20.67**	**22.85**	**19.60**	**18.36**	**16.13**
		L2H-AMP-TV-2D-B	20.47	19.39	20.56	22.25	19.10	18.23	16.11

Table 4.8: PSNRs (dB) of the 128×128 image reconstructions with HR-AMP and L2H-AMP. The transform domain in AMP-ST is DCT.

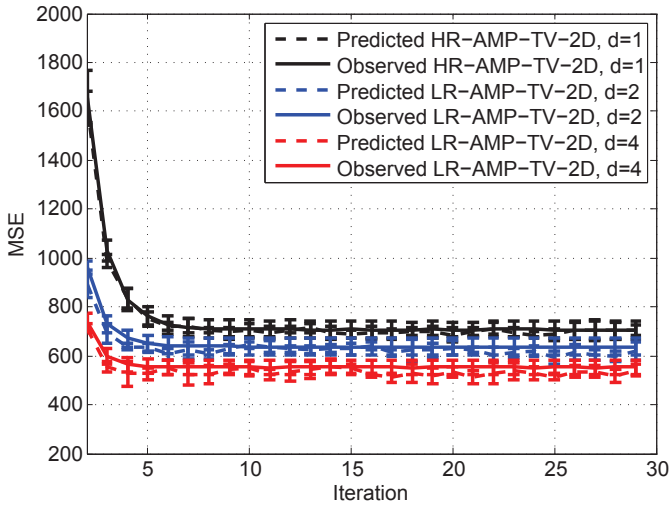
d=2 for LR-AMP-ST			d=2 for LR-AMP-TV-2D			
δ_1%	HR-AMP-ST	LR-AMP-ST	δ_1%	HR-AMP-TV-2D	LR-AMP-TV-2D-R	LR-AMP-TV-2D-B
5	10.8969	3.7318	5	9.9753	2.4964	2.3032
10	11.5907	3.9249	10	6.9049	2.2856	2.0812
20	12.5869	4.1898	20	5.7327	2.6401	2.4869
d=4 for LR-AMP-ST			d=4 for LR-AMP-TV-2D			
δ_1%	HR-AMP-ST	LR-AMP-ST	δ_1%	HR-AMP-TV-2D	LR-AMP-TV-2D-R	LR-AMP-TV-2D-B
3	0.2489	0.0075	3	14.4794	0.8791	0.8486
4	0.3205	0.0080	4	11.8068	0.8831	0.8594
5	0.3937	0.0107	5	9.9753	0.9104	0.9005

Table 4.9: CPU running time in seconds for different methods for the 128×128 Barbara image.

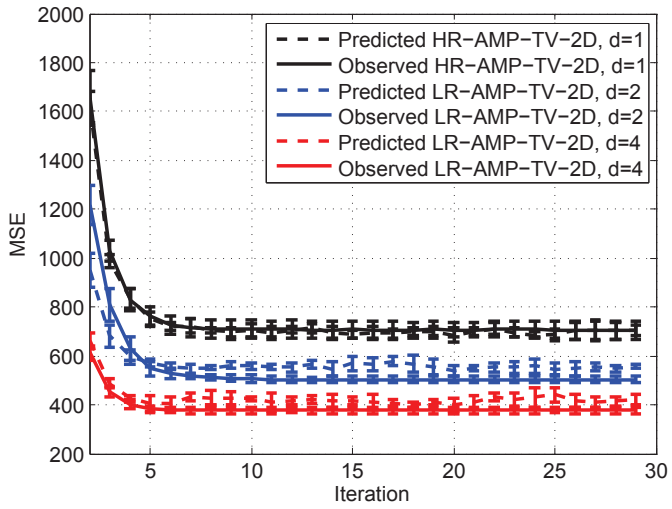
(the H2L-AMP is even slower than HR-AMP due to the additional downsampling), and the spatial-domain method is faster than the transform-domain method. However, when $d = 4$ (the size of the LR image is 1/16 that of the HR image), the thresholding rule in the soft-thresholding denoiser is changed from the time-consuming optimal SURE method in [73] for $d = 2$ to the fast suboptimal max-min method in [69]. Thus, the LR-AMP-ST is approximately 36 times faster than HR-AMP-ST, the latter being approximately 25 times faster than the HR-AMP-TV, and LR-AMP-ST is approximately 100 times faster than LR-AMP-TV. Moreover, LR-AMP-TV is approximately 13 times faster than HR-AMP-TV. This provides some guidelines on how to choose the appropriate method according to the value of d when the complexity is a primary concern.

4.6 Summary

In this chapter, we systematically study the multi-resolution compressed sensing reconstruction problem, which can stably recover a low-resolution signal when the sampling rate is too low to recover the full resolution signal. We develop an AMP-based solution and study its theoretical performance. We also develop the appropriate up-/down-sampling operators in both the transform and spatial domains. The performance of the proposed scheme is demonstrated via simulation results.



(a)



(b)

Figure 4.2: State evolutions of MR-AMP-TV with a CS sampling rate of 5% and no measurement noise for the 128×128 Barbara image. (a) Repetition interpolator. (b) Bicubic interpolator.

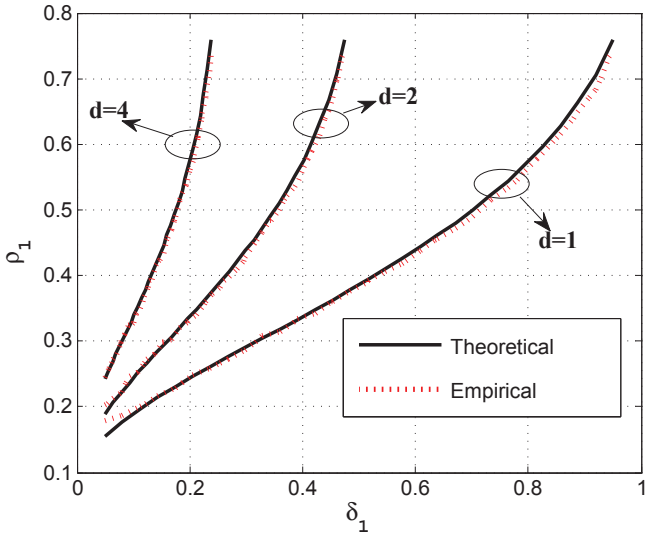
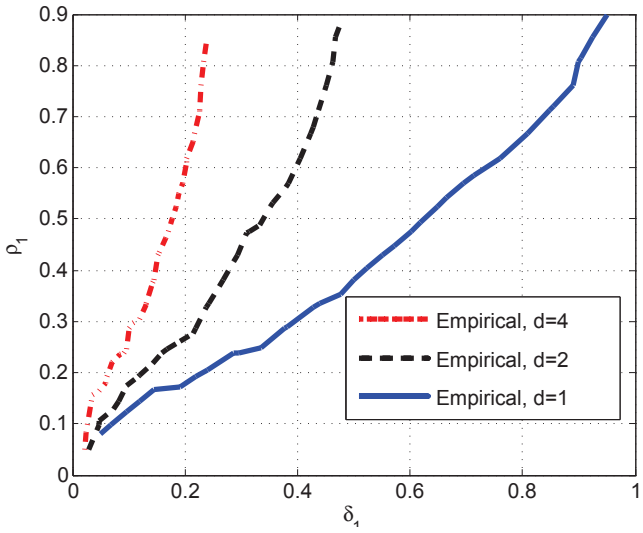
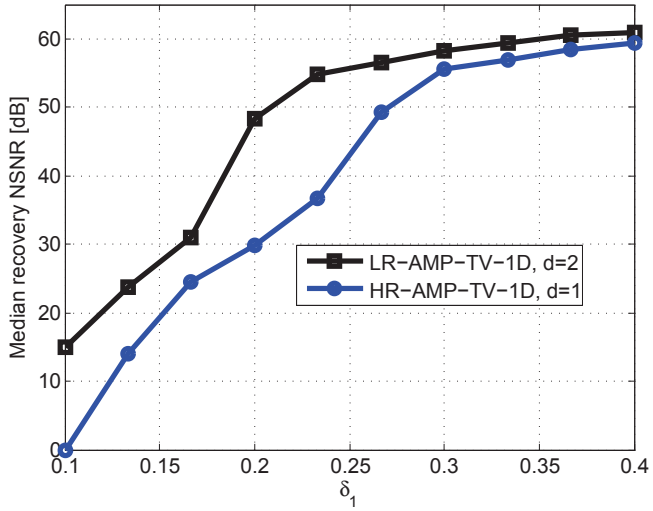


Figure 4.3: The theoretical and empirical PTCs of MR-AMP-ST.

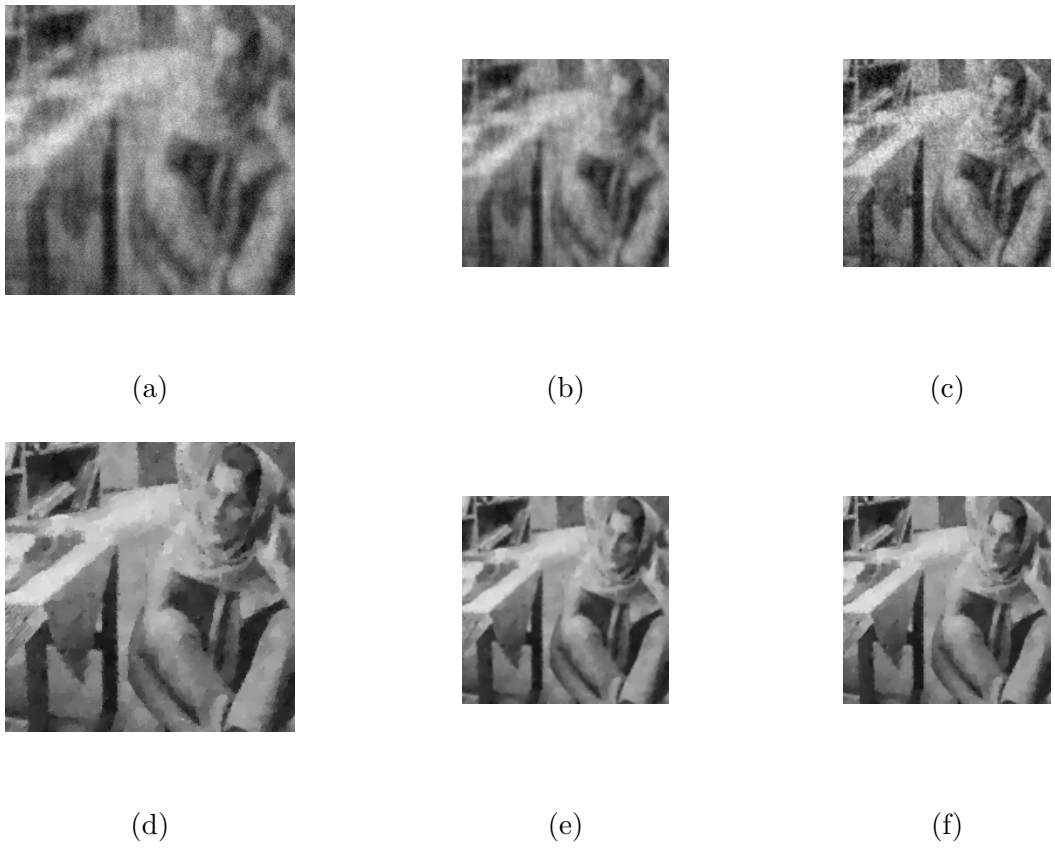


(a)



(b)

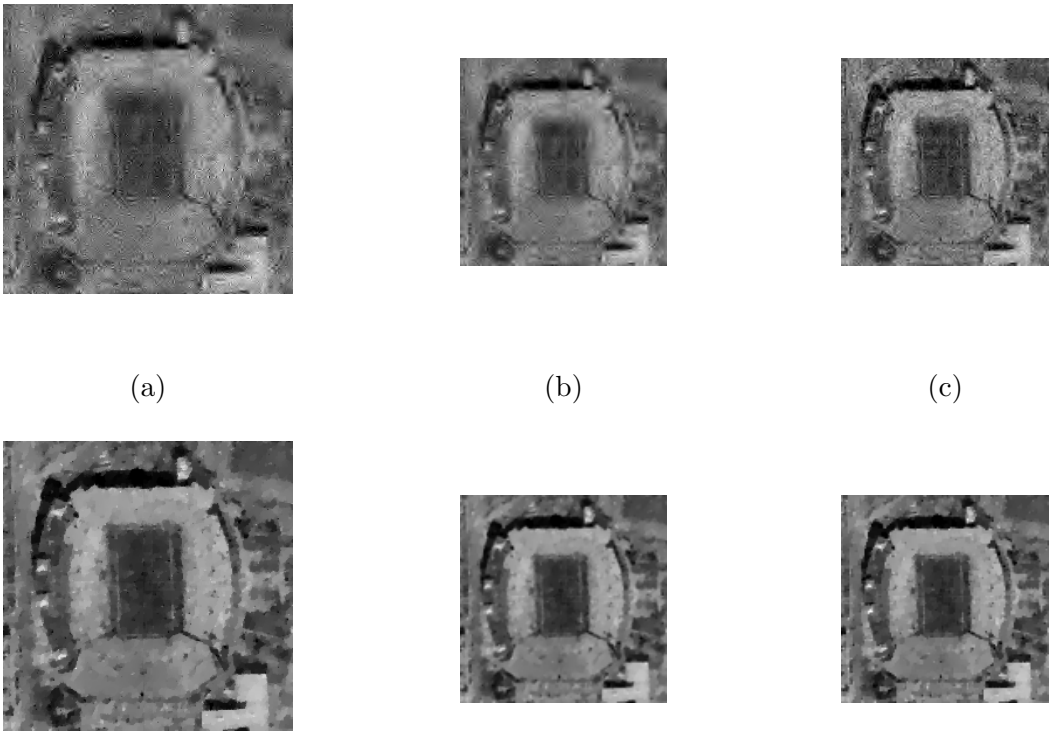
Figure 4.4: (a) The empirical PTCs of MR-AMP-TV-1D for Bernoulli-Gaussian finite-difference signals. (b) MR recovery of Bernoulli-Gaussian finite-difference signals with sparsity rate $\varepsilon_1 = 0.05$ and SNR of 60 dB in the measurement.



(a) (b) (c)

(d) (e) (f)

Figure 4.5: Reconstruction of 10% sampled 256×256 Barbara image with downsampling factor $d = 2$ and DCT as the sparsifying basis for MR-AMP-ST. (a) HR-AMP-ST (20.32 dB). (b) H2L-AMP-ST (21.31 dB). (c) LR-AMP-ST (22.72 dB). (d) HR-AMP-TV-2D (25.06 dB). (e) H2L-AMP-TV-2D (27.75 dB). (f) LR-AMP-TV-2D-B (27.54 dB).



(a) (b) (c)

(d) (e) (f)

Figure 4.6: Reconstructions of 20% sampled 256×256 HuskerStadium image with downsampling factor $d = 2$ and D8 wavelet as the sparsifying basis for MR-AMP-ST. (a) HR-AMP-ST (18.65 dB). (b) H2L-AMP-ST (20.22 dB). (c) LR-AMP-ST (21.05 dB). (d) HR-AMP-TV-2D (21.66 dB). (e) H2L-AMP-TV-2D (24.52 dB). (f) LR-AMP-TV-2D-B (24.38 dB).

Chapter 5

Scalable Compression of Deep Neural Networks

Deep neural networks (DNNs) have shown great success in versatile high level vision problems such as image classification [49, 60], face recognition [6], object detection [40], and image captioning [57]. However, DNNs generally involve multiple layers with millions of parameters, making them difficult to be deployed and updated on devices with limited resources such as mobile phones and other smart embedded systems. For image classification task, the winner of ILSVRC2012, AlexNet, has 60M parameters and needs 240 MB of storage space. The second place of ILSVRC2014, VGG-16, has 138M parameters and needs 530 MB of storage space. To make things worse, various high level vision tasks need different DNN models, e.g., the model used for face recognition [6] is different from the model used for image classification. We need to store both models in embedded systems if we plan to perform image classification and face recognition.

In this chapter, motivated by the successful applications of scalable coding in various image and video coding standards such as JPEG 2000, H.264, and H.265/HEVC [85,88,92], we propose a scalable compression framework for DNNs, which has not been addressed before. Our goal is to represent the DNN parameters in a scalable fashion such that we can easily truncate the representation of the network according to the storage constraint and still get near-optimal performance at each rate. Moreover, if the network needs to be upgraded with higher rate and better performance, the existing low-rate network can be reused, and only some incremental data are needed. This is better than recompressing and re-transmitting the network as in [45, 58].

To achieve this goal, we propose a three-stage pipeline. First, a hierarchical representation of weights in DNNs is developed. Second, we propose a backward greedy search algorithm to adaptively select the bits assigned to each layer given the total bit budget. Finally, we fine-tune the compressed model.

The rest of the chapter is organized as follows. Sec. 5.1 is devoted to hierarchical quantization of the DNN parameters. In Sec. 5.2 we formulate the bit allocation as an optimization problem and propose a backward search solution. A fine-tuning method is presented in Sec. 5.3. Experimental results on MNIST, CIFAR-10 and ImageNet datasets are reported in Sec. 5.4, followed by conclusions in Sec. 5.5.

5.1 Hierarchical Quantization

The K-means clustering-based quantization is a popular technique in the compression of DNN [42], [45]. Therefore, in this chapter, we also choose K-means clustering with linear initialization [45] to compress the weights in DNN. However, the framework developed in this chapter is quite general and can also be applied to other quantization techniques, e.g., the fixed-point quantization in [65] and other similar tasks besides classification, e.g., regression problems.

In [45], the authors quantize the weights to enforce weight sharing with K-means clustering, e.g., they assign 8 bits (256 shared weights) to each CONV layer and 5-bits (32 shared weights) to each FC layer. However, every time a CONV layer is assigned a different bit, the K-means clustering has to be performed again, rendering scalable compression infeasible. On the other hand, some DNN layers have a large number of weights, e.g., the number of weights in the fc6 layer of AlexNet is 38M. Therefore the K-means clustering can be quite slow, even with the help of GPU.

To address this problem, we adopt the scalable coding concept in image/video coding [85, 88, 92], and represent the weights hierarchically, i.e., each weight is represented by a base-layer component and several enhancement-layer components; hence, we only need to perform the quantization step once during the entire scalable compression process, which also benefits the adaptive bit allocation in Sec. 5.2. Note that there are two different kinds of layers in this chapter: the network layers in DNN, and the hierarchical quantization layers in the scalable representation of the weights.

Suppose we want to allocate n bits to each weight in a pre-trained DNN layer. We first perform K-means clustering of all weights with $K = 2$ (1-bit quantization), and record the corresponding cluster indices and centroids. We also record the corresponding quantization error. This yields the 1-bit base-layer approximations of all weights. Next, we perform another K-means clustering with $K = 2$ on all quantization errors, and record the corresponding cluster indices, centroids, and quantization errors. The gives us the 1-bit first-enhancement-layer representations of all weights. By repeating this procedure, we can obtain a n-layer hierarchical representation of a weight, i.e.,

$$w \approx b_1 + e_1 + ... + e_{n-1}, \tag{5.1}$$

where w is a uncompressed weight, b_1 and e_i are the centroid of the base layer and the i-th enhancement layer respectively.

This hierarchical quantization only needs to be performed once, which facilitates future network updating, as we only need to add or delete certain quantization layers to meet the new bit rate constraint. For the tradition K-means clustering used in [42] and [45], we have to perform K-means clustering every time a new bit budget is required.

After the hierarchical quantization, we can build a codebook that stores the centroid and cluster index information of all quantization layers. For a network layer of DNN with N weights, there are $2n$ centroids, and the number of cluster indices is Nn. If each uncompressed weight or centroid is represented by b bits (b=32 for single-precision floating-point number), the compression rate of the n-bit hierarchical quantization scheme is

$$r = \frac{Nb}{Nn + 2nb}. \tag{5.2}$$

In contrast, in the conventional K-means method [45], given the same n-bit quantization, the compression rate is

$$r = \frac{Nb}{Nn + 2^n b}. \tag{5.3}$$

Note that the storage cost is dominated by Nn, compared to $2nb$ or $2^n b$, because the number of connections N in a DNN is usually very large.

5.2 Adaptive Bit Allocation

In DNN, the redundancies in different network layers are different [45, 65]. Therefore it is necessary to design an optimal bit allocation algorithm, i.e., given a bit budget, how to allocate the bits to different network layers in order to get the best performance. In this part, we formulate the following optimization problem.

$$\begin{aligned} &\arg \min_{\{\mathbf{n},\mathbf{C},\mathbf{G}\}} f(\mathbf{n}, \mathbf{C}, \mathbf{G}) \\ &\text{s.t.} \sum_{i=1}^{L} N_i n_i + 2n_i b \leq \mu. \end{aligned} \tag{5.4}$$

where $\mathbf{n} = \begin{bmatrix} n_1 & ... & n_L \end{bmatrix}$ is a vector containing the bits allocated to L network layers, $\mathbf{C}$ is the centroid vector, $\mathbf{G}$ is the cluster-by-index matrix for the network layers, N_i is the number of weights in the i-th network layer, and μ is the bit budget. We use the cross entropy between the pdf of the predicted labels and true labels as the cost function $f(\cdot)$, which is frequently used in classification tasks.

It is hard to solve the combinatorial optimization in Eq. (5.4), since the number of bits assigned to each network layer n_i has to be integer and the number of entries in the cluster-by-index matrix $\mathbf{G}$ is $\sum_{i=1}^{L} N_i n_i$, even larger than the number of weights $\sum_{i=1}^{L} N_i$ in

the pre-trained DNN. Therefore, we use a similar method to [45] to first approximate the original uncompressed weights with high-rate quantized weights. More specifically, we first use the hierarchical method in Sec. 5.1 to assign M bits to each CONV layer weight and P bits to each FC layer weight. This is used as the initialization step. The centroid vector $\mathbf{C}$ and the cluster-by-index matrix $\mathbf{G}$ are then determined and fixed. We use E to denote the number of bits to store this initial network.

Next, we adaptively allocate bits to network layers such that $u < \mathrm{E}$. The problem in Eq. (5.4) is simplified to

$$\begin{aligned} &\arg\min_{\{\mathbf{n}\}} f(\mathbf{n}) \\ \text{s.t. } &B = \sum_{i=1}^{L} (N_i + 2b) n_i \leq \mu. \end{aligned} \tag{5.5}$$

For small-scale problems, the optimization above can be solved by exhaustive grid search, where configurations that violate the bit constraint are skipped, and the others are evaluated to find the best solution. The process can be accelerated by parallel computing, since different configurations are independent. However, for large-scale problems, exhaustive search becomes infeasible, as the number of configurations grows exponentially with the number of bits. For example, in AlexNet, there are 5 CONV layers and 3 FC layers. If 10 bits are assigned to each CONV layer and 5 bits are assigned to each FC layer, the total number of configurations would be $10^5 \times 5^3 = 12.5$M.

One way to speed up the process is to use random search [13], since the number of bits assigned to each network layer can be treated as a hyper-parameter for the DNN. Theoretical analysis in [13] shows that randomly selecting 60 configurations can ensure that the top 5% result can be achieved with a probability of 0.95. For the bit allocation problem here, we should randomly select a number of configurations that satisfy the bit constraint. In Sec. 5.4, random search is used as a baseline algorithm for comparison.

In this part, we propose a backward greedy search algorithm to address the bit constraint explicitly and solve the problem in Eq. (5.5). We start from the initial high-rate quantized network as discussed above. Denote the bit allocation in the t-th iteration as $\mathbf{n}^t = [n_1^t, \ldots, n_L^t]$, whose corresponding total bit cost is B^t. To find $\mathbf{n}^{t+1}$ at iteration $t+1$, we follow the spirit of the gradient descent method by assigning one less bit to each network layer respectively, calculating the corresponding gradient of the total bit cost, and choosing the configuration that has the maximum gradient. In other words, let $\mathbf{n}^{t,j} = [n_1^t, \ldots, n_{j-1}^t, n_j^t - 1, n_{j+1}^t, \ldots, n_L^t]$, the bit allocation in the $(t+1)$-th iteration is obtained by

$$\begin{aligned} &\arg\max_{\mathbf{n}^{t,j}} \tfrac{f(\mathbf{n}^{t,j}) - f(\mathbf{n}^t)}{B^{t,j} - B^t} \\ \text{s.t.} \quad &\mathbf{n}^{t,j} \subset \{\mathbf{n}^{t,1}, \mathbf{n}^{t,2}, ..., \mathbf{n}^{t,L}\}, \\ &B^{t,j} = \sum_{i=1}^{L} (N_i + 2b) n_i^{t,j}. \end{aligned} \tag{5.6}$$

The iteration terminates until the bit constraint is satisfied. The entire backward greedy search algorithm is summarized in Alg. 1. The intuition behind the gradient defined above is twofold. First, if two bit allocations have the same cost function value, the one with smaller total bit cost should be chosen. Second, if two bit allocations have the same total bit cost, we should choose the one with lower cost function value and use the maximum function in Eq. (5.6) due to $B^{t,j} < B^t$.

Algorithm 1 Backward Greedy Search Algorithm

Initialization: Quantize the network with M bits for each CONV layer and P bits for each FC layer. Let $t = 0$.
while $B^t > \mu$ **do**
  for each network layer $j \leq L$ **do**
    $n_j^{t,j} \leftarrow n_j^t - 1, n_p^{t,j} \leftarrow n_p^t \, \text{for} \, p \neq j$.
    Update the weights of DNN based on the hierarchical framework in Sec. 5.1
    Test with the validation data and record $B^{t,j}$ and $f(\mathbf{n}^{t,j})$
  end for
  Select $\mathbf{n}^{t+1}$ based on Eq. (5.6)
  $t \leftarrow t + 1$
end while

5.3 Fine Tuning

It is shown in [45, 65] that fine-tuning (FT) of the centroids after the quantization of DNN can significantly improve the classification performance. In this chapter, we also perform fine-tuning after the adaptive bit allocation to update the centroids similar to Eq. (3) in [45].

Denote the loss by f, the weight in i-th column and j-th row by W_{ij}, the cluster index of weight W_{ij} by G_{ij}, and the k-th centroid in the m-th quantization layer by C_{mk}. With the indicator function $\mathbb{I}(\cdot)$, the gradient of the centroids is calculated as

$$\frac{\partial f}{\partial C_{mk}} = \sum_{i,j} \frac{\partial f}{\partial W_{ij}} \frac{\partial W_{ij}}{\partial C_{mk}} = \sum_{i,j} \frac{\partial f}{\partial W_{ij}} \mathbb{I}(G_{ij} = m^k) \tag{5.7}$$

where m^k is the index k in the m-th quantization layer.

The advantage of the proposed scalable compression of the DNN is that for each target bit rate, we can find a near-optimal bit allocation. If later on the DNN bit rate on a device needs to be updated, instead of re-transmitting a new set of the DNN parameters, we only need to transmit some incremental data, including the centroid vector $\mathbf{C}$ and cluster-by-index matrix $\mathbf{G}$. The required bit rate is thus much lower than replacing the entire network.

During the update, some additional bits caused by the fine-tuning are needed to update the centroids of the previous compressed model. However, according to the analysis in Sec.

5.1, the centroid update will cost $2b\sum_{i=1}^{L} n_i$ at most, while the minimal bits needed to update the cluster-by-index matrix are $\min\{N_1, N_2, ..., N_L\}$. The storage cost is dominated by the cluster indices instead of centroids; hence the overhead introduced by the fine-tuning is negligible. Take AlexNet as an example, if we use 10 bits to quantize CONV layers and 5 bits for FC layers, at most 0.52KB are needed to update these centroids, while we may use at least 5KB to update the cluster-by-index matrix every time a different bit budget is given.

5.4 Experimental Results

We test the proposed scalable compression on 3 networks designed for the MNIST [63], CIFAR-10 [59] and ImageNet [81] datasets respectively. We implement the network training based on the CNN toolbox MatConvNet [98] with our own modifications. The training is done on a desktop with a NVIDIA TIAN X GPU with 12GB memory.

5.4.1 Implementation Details

For the fine-tuning part, the initial learning rate for LeNet-5, CIFAR-10-quick and AlexNet is 1e-6, 1e-5 and 1e-8, respectively. The reason why the initial learning rate is so small is the gradient of a certain centroid is the sum of the gradients of weights that share the same centroid, which would be rather large in each iteration. We drop the learning rate by 10 when the loss begins to reach an apparent plateau, repeating this several times.

5.4.2 LeNet-5 for MNIST

We use the cnn_mnist_experiment.m function in MatConvNet to train LeNet-5 for MNIST dataset. There are 2 CONV layers and 2 FC layers. The pre-trained model can achieve 0.88% Top-1 error and needs a storage of 1720KB. We use 8 bits to hierarchically quantize each CONV layer and 5 bits for each FC layer. The initial quantized model can achieve 0.97% Top-1 error, and the corresponding storage cost is 279KB. In Fig. 5.1(a), we compare the proposed backward greedy search method (BS) with the exhaustive grid search method (GS). We also present the number of configurations tested on the validation set in Table 5.1 to compare the computational complexity. We can see that our proposed backward search algorithm can achieve comparable compression performance to the grid search with much smaller computational complexity. The only exception happens when the compression rate is extremely large, e.g., 28.67 in Fig. 5.1(a). However, after fine tuning, the performance is still very close to the original one.

LeNet-5 for MNIST				
Bit Budget (KB)	200	150	80	60
Compression Rate	8.60	11.47	21.50	28.67
BS Number	26	51	101	115
GS Number	960	640	320	79
CIFAR-10-quick for CIFAR-10				
Bit Budget (KB)	120	100	50	30
Compression Rate	4.85	5.82	11.64	19.40
BS Number	26	51	101	115
RS Number	120			

Table 5.1: Number of configurations tested on MNIST and CIFAR-10 validation set v.s. compression rate.

5.4.3 CIFAR-10-quick for CIFAR-10

We use the provided cnn_cifar.m in MatConvNet to train CIFAR-10-quick for CIFAR-10 dataset. There are 3 CONV layers and 2 FC layers in the network. The reference model can achieve 19.97% Top-1 error and needs a storage space of 582KB. We use 10 bits to quantize each CONV layer and 5 bits for each FC layer. The initial quantized model can achieve 22.70% Top-1 error and needs 141KB storage space. Since there are at most 25K configurations which takes too much time to evaluate, instead of using grid search as a comparison, we use the random search method (RS) [13]. In each trial, we randomly choose 120 configurations that satisfy the bit constraint from the configuration pool.

The compression performance is shown in Fig. 5.1(b) and the computational complexity is presented in Table 5.1. It can be seen that the proposed backward search algorithm can achieve similar or even better performance than random search with much smaller computational complexity, especially when the bit rate is close to that of the initial quantized network. The only exception happens when the compression rate is extremely large, e.g., 20 in Fig. 5.1(b). For the fine-tuning in the random search method, we fine-tune the result that achieves the median classification accuracy in the 10 trials.

5.4.4 AlexNet for ILSVRC12

We use the provided cnn_imagenet.m to train AlexNet for ILSVRC12. The reference model is slightly different from that of the original AlexNet in [60], where the order of pooling layer and norm layer are swapped. It contains 5 CONV layers and 3 FC layers. This reference model can achieve 41.39% Top-1 error, 18.85% Top-5 error, and needs 240 MB to store. We use 10 bits to quantize each CONV layer and 5 bits for each FC layer. This initial quantized model can achieve 56.09% Top-1 error, 31.63% Top-5 error, and needs 39.5 MB to store. The number of configurations in each trial of RS is 150. The number of trials is 5

Bit Budget (MB)	35	20	15	10
Compression Rate	6.86	12	16	24
BS Number	25	81	89	126
RS Number	150			

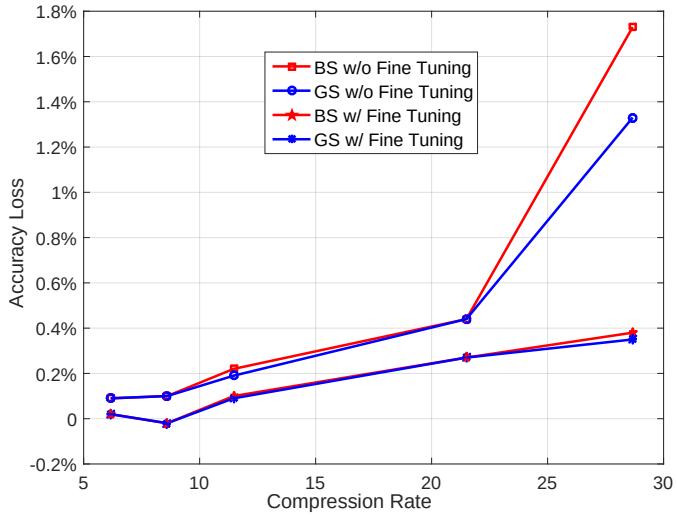
Table 5.2: Number of configurations tested on ILSVRC12 validation set v.s. compression rate.

in order to get 0.95 confidence interval. The result that achieves the median classification accuracy in the 5 trials is fine-tuned.

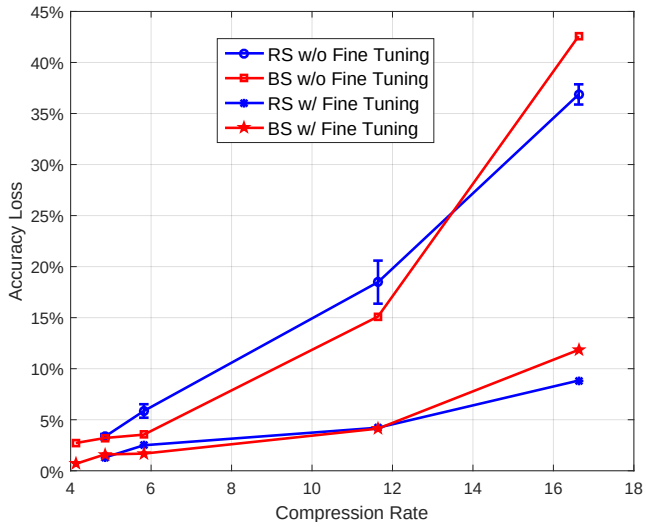
The compression performance is shown in Fig. 5.2, and the computational complexity is shown in Table 5.2. We can see that with much smaller computational complexity, the proposed backward search can achieve better compression performance than random search. Moreover, the classification performance of proposed scalable compression framework drops little when the compression rate is within 10.

5.5 Summary

In this chapter, we discuss the scalable compression of deep neural networks, and propose a three-stage pipeline: hierarchical quantization of weights, backward greedy search for bit allocation, and fine-tuning. Its efficacy is tested on three different DNNs.

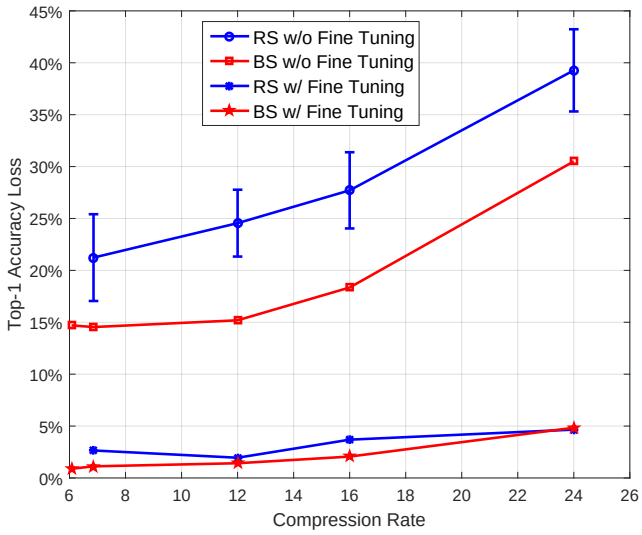


(a)

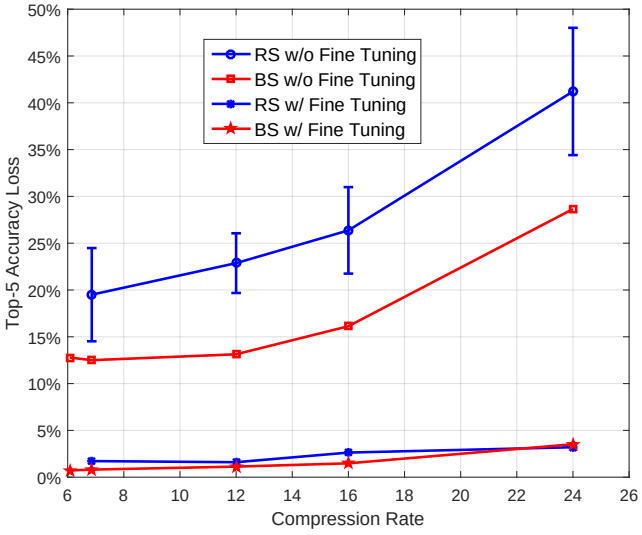


(b)

Figure 5.1: Top-1 accuracy loss of compressed DNNs under different bit allocation methods. (a) LeNet-5 and (b) CIFAR-10-quick.



(a)



(b)

Figure 5.2: Accuracy loss of compressed AlexNet v.s. compression rate.

Chapter 6

Conclusions

6.1 Conclusions

In this thesis, we investigate some novel topics on two different compression concepts in "Big Data", compressed sensing in data compression and deep learning model compression.

Although compressed sensing and deep learning are relatively new, compared to image/video coding, in this thesis, we show that some old techniques and ideas in image/video coding which has been studied extensively, can inspire and solve new problems in CS and DL. In the first topic of this thesis, side information-aided compressed sensing is modelled as a distributed source coding problem. The next two topics, scalable compressed sensing and scalable compression of deep neural networks are both motivated by scalable image/video coding.

More specifically, in the first topic of this thesis, we study side information-aided compressed sensing, where an additional noisy version of the original signal is available for CS reconstruction. We model this problem from the setup of distributed source coding, incorporate the side information into the decoding process and formulate the corresponding optimization problem. Next, we develop a GENP-aided approximate message passing algorithm (GENP-AMP), and study its parameter selection, state evolution, and noise sensitivity. The contribution of the GENP is also examined. We also develop a parameterless GENP-AMP that does not need to know the sparsity of the unknown signal and the variance of the GENP. Simulation results with 1-D synthetic data and multiview images demonstrate the performances of the proposed methods.

Motivated by the quality scalability in scalable image/video coding, the second topic of this thesis, multi-resolution compressed sensing, tries to answer the following question: if the number of CS samples is not sufficient to reconstruct the target high-resolution image, is it possible to stably recover the corresponding low-resolution preview ? We systematically study the multi-resolution compressed sensing reconstruction problem, develop an AMP-based solution and study the theoretical performance. Moreover, we also develop

the appropriate up-/down-sampling operators in both transform and spatial domains. The performance of proposed scheme is demonstrate on both synthetic 1-D data and 2-D images.

A key benefit of deep learning is the analysis and learning of massive amounts of data, making it a valuable tool for Big Data Analytics. However, DNNs generally involve multiple layers with millions of parameters, making them difficult to be deployed and updated on devices with limited resources such as mobile phones and other smart embedded systems. Motivated by scalable image/video coding, where scalability is achieved by the hierarchical representation of bitstream, we hierarchically quantize the weights in DNNs and adaptively select the subsets of output from hierarchical quantization based on user specified bit constraint. Finally, we fine tune the centroids from hierarchical quantization step to improve the final performance. Experimental results on several famous deep learning models are also presented.

6.2 Future Work

In this thesis, we show that some old ideas and techniques in image/video coding can solve and inspire new problems in new areas, i.e., compressed sensing and deep learning. We hope this thesis can inspire the readers with image/video coding background to contribute more in this direction. Some interesting future work include, but not limited to :

6.2.1 Side Information-aided Multiview Video CS Reconstruction

For the future work of GENP-AMP as presented in Chapter 3, a parameterless GENP-AMP algorithm that can accurately work in the whole plane need to be developed. According to the noise sensitivity analysis in Sec. 3.4, there is no phase transition boundary, and the MSE is bounded in the whole plane. However, the parameterless GENP-AMP proposed in Sec. 3.5 only works well below the phase transition boundary of the standard AMP, due to the unbounded MSE above the phase transition boundary of the standard AMP and the approximation accuracy of SURE.

The original AMP is based on the simple soft thresholding in each iteration. Recently, it is found in [71, 89] that other denoising methods can be employed in AMP to further improve the reconstruction. For example, using the BM3D denoising algorithm [22], state-of-the-art CS reconstructions can be achieved in imaging applications. This approach can also be adopted into the GENP-AMP framework in this paper.

Applying the proposed schemes to multiview videos instead of multiview images is another attractive topic, where the approaches in [108, 109] could be useful. Since there is a third temporal dimension in videos, compared to images, it will be quite interesting if we borrow some ideas from video coding and fuse them into the proposed schemes to improve the video CS reconstruction performance.

6.2.2 Multi-Resolution Video CS Reconstruction

Scalable image CS reconstruction in Chapter 4 is motivated by the quality scalability in scalable image/video coding. Actually, there are three scalability concepts in scalable video coding: temporal (frame rate) scalability, spatial (picture size) scalability and quality scalability [85]. It remains unknown how we can achieve temporal and spatial scalability under CS setup. It is pretty exciting if we can extend current multi-resolution image CS reconstruction method to multi-resolution video CS reconstruction by taking these three scalability concepts into consideration.

6.2.3 Compression of Deep Neural Networks

We are currently considering the following for future work. In [45], the authors can compress AlexNet from 240 MB to 6.9 MB without loss of accuracy, which is much smaller than what is achieved in this paper. The reason is that network pruning is used [46], which removes many small-weight connections from the network. This not only compresses the network, but also reduces the complexity of the implementation. In addition, entropy coding is used in [45]. However, the quantization in [45] is fixed and not scalable. This paper focuses on the scalable quantization and adaptive bit allocation. It is also shown from Fig. 7 in [45] that pruning does not hurt quantization. Therefore the pruning and entropy coding can also be used in our scheme to further improve the performance. Moreover, Huffmann coding is the entropy coding method used in [45]. There are many more advanced entropy coding methods than Huffmann coding in image/video coding, e.g., Golomb-Rice coding, and arithmetic coding. Higher compression rate can be expected if we include pruning and more advanced coding methods into our scalable compression scheme. Moreover, It is of great importance to provide theoretical analysis on the compression performance of DNNs. There is plenty of theoretical work on image/video coding performance. If the readers are interested, they can try to extend the rate-distortion theory in image/video coding, e.g., [21] to the compression of DNNs and get some theoretical results.

6.2.4 Sparsity-Constrained Deep Learning

The number of neurons in the human brain is close to 2e10. Each neuron is only connected to about 1e4 other neurons on average though [4]. In deep learning, we see this in CNN. Each neuron receives input only from a very small patch in the layer below. Unfortunately, so far, learning weights that are sparse has not really paid off. Although there are some work on sparsity-constrained deep learning [66, 105], they are built on pre-trained CNN models. Developing other deep learning models besides CNN, whose weights are sparse, is still of great interest. In the future, we will focus on building sparse deep models from scratch instead of pre-trained CNN models. We also note that sparsity plays an important role in the transcoding of image/video coding that images are sparse in transform domain.

There have been some preliminary results on training deep learning models from scratch in frequency domain [80]. More effort is needed in this area to build a connection between deep learning in transform domain and sparsity.

Bibliography

[1] Eyevision. http://www.wikipedia.org/wiki/Bullet-time.

[2] Fujii lab's multi-view sequences download lists. http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/.

[3] The matrix. http://www.wikipedia.org/wiki/Bullet-time.

[4] Neuron. https://en.wikipedia.org/wiki/Neuron.

[5] A. Mohamed A. Graves and G. Hinton. Speech recognition with deep recurrent neural networks. In Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., pages 6645–6649, 2013.

[6] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[7] G. Andrew and J. Gao. Scalable training of ℓ_1-regularized log-linear models. In Proc. of International Conference on Machine Learning, pages 33–40, 2007.

[8] J. Barbier. Statistical physics and approximate message passing algorithms for sparse linear estimation problems in signal processing and coding theory. PhD thesis, Université Paris Diderot, 2015.

[9] R. Baron, S. Sarvoham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. IEEE Trans. Signal Proc., 58(1):269–280, Jan. 2010.

[10] M. Bayati and A. Montanari. The dynamics of message passing on dense graphs, with applications to compressed sening. IEEE Trans. Inf. Theory, 57(2):1462–1474, 2011.

[11] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal in Imgace Sciences, 2(1):183–202, 2009.

[12] P. Beigi, X. Xiu, and J. Liang. Compressive sensing based multiview image coding with belief propagation. In Proc. Asilomar Conference on Signals, Systems, and Computers, pages 430–433, 2010.

[13] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13:281–305, February 2012.

[14] M. Borgerding and P. Schniter. Generalized approximate message passing for the cosparse analysis model. In IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 3756–3760, Apr. 2015.

[15] E. J. Candes and T. Tao. Decoding by linear programming. IEEE Transaction on Information Theory, 51(12):4203–4215, 2005.

[16] T. Canh, K. Quoc, and B. Jeon. Multi-resolution kronecker compressive sensing. IEIE Transactions on Smart Processing and Computing, 3(1):19–27, Feb. 2014.

[17] A. S. Charles, M. S. Asif, J. Romberg, and C. J. Rozell. Sparsity penalties in dynamic system estimation. In Conference on Information Science and Systems, pages 1–6, 2011.

[18] S. S. Chen, D. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20(1):33–61, 1998.

[19] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 2285–2294. JMLR Workshop and Conference Proceedings, 2015.

[20] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12:2493–2537, Feb. 2011.

[21] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience, 2006.

[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Proc., 16(8):2080–2095, 2007.

[23] G.E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on, 20(1):30 –42, jan. 2012.

[24] M. Denil, B. Shakibi, L. Dinh, M. A. Ranzato, and N. Freitas. Predicting parameters in deep learning. In Advances in Neural Information Processing Systems, pages 2148–2156. Curran Associates, Inc., 2013.

[25] E. L. Denton, W. Zaremba, J. Bruna, Y. Lecun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In Advances in Neural Information Processing Systems, pages 1269–1277. Curran Associates, Inc., 2014.

[26] T. Do, Y. Chen, D. T. Nguyen, L. Gan, and T. D. Tran. Distributed compressed video sensing. In IEEE International Conference on Image Processing, pages 1393–1396, 2009.

[27] D. Donoho. Compressed sening. IEEE Transaction on Information Theory, 52(4):1289–1306, Apr. 2006.

[28] D. Donoho, I. Johnstone, and A. Montanari. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising. IEEE Trans. Inf. Theory, 59(6):3396–3433, Jun. 2013.

[29] D. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. Biomefrika, 81(3):425–455, 1994.

[30] D. Donoho and I. M. Johnstone. Minimax risk over l_p balls. Prob. Theory and Rel. Fields, 99:277–303, 1994.

[31] D. Donoho, A. Maleki, and A. Montanari. Message passing algorithms for compressed sensing. Proceedings of the National Academy of Sciences, 106(45):18914–18919, 2009.

[32] D. Donoho, A. Maleki, and A. Montanari. The noise-sensitivity phase transition in compressed sensing. IEEE Transaction on Information Theory, 57(10):6920–6941, Oct. 2011.

[33] M. F. Duarte and R. G. Baraniuk. Kronecker compressive sensing. IEEE Trans. Image Proc., 21(2):494–504, Feb. 2012.

[34] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. IEEE Signal Processing Magazine, 25(2):83–91, Mar. 2008.

[35] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction. IEEE Journal of Selected Topics in Signal Processing, 1(4):586–597, Dec. 2007.

[36] T. Frajka and K. Zeger. Downsampling dependent upsampling of images. Signa Processing: Image Communication, 19:257–265, 2004.

[37] L. Gan, T. T. Do, and T. D. Tran. Fast compressive imaging using scrambled block hadamard ensemble. In European Signal Processing Conf. (EUSIPCO), 2008.

[38] P. Getreuer. Rudin-Osher-Fatemi total variation denoising using split bregman. Image Processing On Line, pages 74–95, May 2012.

[39] R. Girshick. Fast r-cnn. In International Conference on Computer Vision (ICCV), 2015.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[41] T. Goldstein, L. Xu, K. Kelly, and R. Baraniuk. The STOne transform: Multi-resolution image enhancement and real-time compressive video. IEEE Trans. Image Proc., 24(12):5581–5593, Dec. 2015.

[42] Y. Gong, L. Liu, M. Yang, and L. Bourdev. Compressing deep convolutional networks using vector quantization. arXiv preprint, arXiv:1412.6115, Dec. 2014.

[43] I. Goodfellow, Y. Bengio, and A. Courville. MIT Press, 2016.

[44] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, Sep. 2013.

[45] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In International Conference on Learning Representations, 2016.

[46] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In Advances in Neural Information Processing Systems 28, pages 1135–1143. 2015.

[47] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge Univ. Press, 2003.

[48] C. He, C. Hu, W. Zhang, and B. Shi. A fast adaptive parameter estimation for total variation image restoration. IEEE Trans. Image Proc., 23(12):4954–4967, Sep. 2014.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint, arXiv: 1512.03385, 2015.

[50] G. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. Neural Computation, 18:2006, 2006.

[51] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, November 1997.

[52] S. Hong. Multi-resolution bayesian compressive sensing for cognitive radio primary user detection. In Proc. IEEE Global Telecommun. Conf., pages 1–6, 2010.

[53] H. Jiang, C. Li, R. Cohen, P. Wilfod, and Y. Zhang. Scalable video coding using compressive sensing. Bell Labs Technical Journal, 16(4):149–170, Mar. 2012.

[54] U. Kamilovr, M. Unser, A. Flectche, and S. Rangan. Approximate message passing with consistent parameter estimation and applications to sparse learning. arXiv preprint, arXiv:1207.3859, Dec. 2012.

[55] J. Kang, H. Jung, H. Lee, and K. Kim. Spike-and-slab approximate message passing recovery for 1-D piecewise-constant signal. arXiv preprint, arXiv: 1408.3930, 2014.

[56] L. W. Kang and C. S. Lu. Distributed compressive video sensing. In IEEE International Conference on Acoustics, Speech and Signal Processing, pages 1169–1172, 2009.

[57] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3128–3137, 2015.

[58] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. In International Conference on Learning Representations, 2016.

[59] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[60] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pages 1097–1105, 2012.

[61] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. IEEE Transaction on Information Theory, 47(2):498–519, Feb. 2001.

[62] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang. Multi-view imaging and 3dtv (special issue overview and introduction).

[63] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[64] C. Li. An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing. Master's thesis, Rice University, 2009.

[65] D. D. Lin, S. S. Talathi, and V. S. Annapureddy. Fixed point quantization of deep convolutional networks. arXiv preprint, arXiv:1511.06393, Nov. 2015.

[66] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy. Sparse convolutional neural networks. In IEEE Conference on Computer Vision and Pattern Recognition, pages 806–814. 2015.

[67] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, November 2015.

[68] A. Maleki. Approximate message passing algorithms for compressed sensing. PhD thesis, Stanford University, 2010.

[69] A. Maleki and A. Montanari. Analysis of approximate message passing algorithm. In Annual Conference in Information Sciences and Systems, pages 1–7, Mar. 2010.

[70] C. Metzler, A. Maleki, and R. Baraniuk. From denoising to compressed sensing. arXiv preprint, arXiv:1406.4175, 2014.

[71] C. A. Metzler, A. Maleki, and R. G. Baraniuk. From denoising to compressed sensing. arXiv preprint, arXiv: 1406.4175, Jun. 2014.

[72] A. Montanari. Graphical models concepts in compressed sensing. In Compressed Sensing Theory and Applications, pages 394–438. Cambrige University Press, 2012.

[73] A. Mousav, A. Maleki, and R. G. Baraniuk. Parameterlss optimal approximate message passing. arXiv preprint, arXiv: 1311.0035, Oct. 2013.

[74] J. Y. Park and M. B. Wakin. A multiscale framework for compressive sensing of video. In Picture Coding Symposium, pages 1–4, 2009.

[75] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In British Machine Vision Conference, 2015.

[76] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. arXiv preprint, arXiv: 1010.5141, Aug. 2012.

[77] S. Rangan, A. Fletcher, V. Goyal, U. Kamilov, J. Parker, P. Schniter, J. Vila, J. Ziniel, and M. Borgerding. gampmatlab: Generalized approximate message passing. `http://sourceforge.net/projects/gampmatlab/files/`, May. 2014.

[78] S. Rangan, A. K. Fletcher, and V. K. Goyal. Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing. IEEE Transaction on Information Theory, 58(3):1902–1923, Mar. 2012.

[79] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), 2015.

[80] O. Rippel, J. Snoek, and R. P. Adams. Spectral representations for convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), 2015.

[81] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. arpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[82] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[83] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In Proceedings of the International Conference on Artificial Intelligence and Statistics, volume 5, pages 448–455, 2009.

[84] C. Salazar and T. D. Tran. A complexity scalable universal dct domain image resizing algorithm. IEEE Trans. Circ. Syst. Video Tech., 17(4):495–499, 2007.

[85] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circ. Syst. Video Tech., 17(9):1103–1120, Sep. 2007.

[86] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint, arXiv:1409.1556, 2014.

[87] S. Som, L. C. Potter, and P. Schniter. On approximate message passing for reconstruction of non-uniformly sparse signals. In IEEE National Aerospace and Electronics Conference, pages 223–229, Jul. 2010.

[88] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro. Standardized extensions of high efficiency video coding. IEEE Journal on Selected Topics in Signal Processing, 7(6):1001–1016, Dec. 2013.

[89] J. Tan, Y. Ma, and D. Baron. Compressive imaging via approximate message passing with image denoising. arXiv preprint, arXiv:1405.4429, May 2014.

[90] J. Tan, Y. Ma, and D. Baron. Compressive imaging via approximate message passing with image denoising. IEEE Trans. Signal Proc., 63(8):2085–2092, Apr. 2015.

[91] M. Tanimoto, T. Fujii, and K. Suzuki. View synthesis algorithm in view synthesis reference software 3.5 document m16090. ISO/IEC JTC1/SC29/WG11 (MPEG), May 2009.

[92] D. Taubman and M. Marcellin. JPEG2000: image compression fundamentals, standards, and practice. Kluwer Academic Publishers, Boston, 2002.

[93] R. Tibshirani. Regression shrinkage and selection with the lasso. J. Royal. Statist. Soc., B 58:267–288, 1996.

[94] M. Trocan, T. Maugey, J. E. Fowler, and B. Pesquet-Popescu. Disparity-compensation compressed-sensing reconstruction for multiview images. In IEEE International Conference on Multimedia and Expo, pages 1225–1228, 2010.

[95] J. A. Troop and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transaction on Information Theory, 53(12):4655–4666, Dec. 2007.

[96] D. Valsesia and E. Magli. Spatially scalable compressed image sensing with hybrid transform and inter-layer prediction model. In IEEE International Workshop on Multimedia Signal Processing (MMSP), pages 373–378, 2013.

[97] N. Vaswani and W. Lu. Modified-CS: Modifying compressive sensing for problems with partially known support. IEEE Transaction on Signal Processing, 58(9):4595–4607, Sep. 2010.

[98] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In Proceeding of the ACM Int. Conf. on Multimedia, 2015.

[99] J. P. Vila and P. Schniter. An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals. arXiv preprint, arXiv: 1310.2806, Oct. 2013.

[100] J. P. Vila and P. Schniter. Expectation-Maximization Gaussian-mixture approximate message passing. IEEE Transaction on Signal Processing, 61(19):4658–4672, Oct. 2013.

[101] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11:3371–3408, Dec. 2010.

[102] M. Wall. Big data: Are you ready for blast-off?, Mar. 2014.

[103] X. Wang and J. Liang. View interpolation confidence-aided compressed sensing of multiview images. In IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 1651–1655, 2013.

[104] Z. Wang, L. Zhang, Y. Yang, J. Zhou, G. B. Giannakis, and T. S. Huang. Deep double sparsity encoder: Learning to sparsity not only features but also parameters. arXiv preprint, arXiv: 1608.06374, Aug. 2016.

[105] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. arXiv preprint, arXiv: 1608.03665, Aug. 2016.

[106] X. Xiu, D. Pang, and J. Liang. Rectification-based view interpolation and extrapolation for multiview video coding. IEEE Transcation on Circuits and Systems for Video Technology, 21(6):693–707, Jun. 2011.

[107] S. Zhu, B. Zeng, L. Fang, and M. Gabbouj. Downward spatially-scalable image reconstruction based on compressed sensing. In Proc. IEEE Conf. on Image Proc., pages 1352–1356, 2014.

[108] J. Ziniel and P. Schniter. Dynamic compressive sensing of time-varying signals via approximate message passing. IEEE Transactions on Signal Processing, 61(21):5270–5284, Nov. 2013.

[109] J. Ziniel and P. Schniter. Efficient high-dimensional inference in the multiple measurement vector problem. IEEE Transactions on Signal Processing, 61(2):340–354, Jan. 2013.

[110] H. Zou and T. Hastie. Regularization and variable selection via elastric net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320, Apr. 2005.

Appendix A

Proofs in Chapter 3

A.1 A heuristic derivation of the state evolution of GENP-AMP

In this section, we derive the state evolution of GENP-AMP in Eq. (3.28) of Sec. 3.3.3. The derivation is generalized from that in [72] for AMP. We start from the GENP-AMP iteration in (3.17) and (3.20), but introduce the following three modifications: (i) The random matrix $\mathbf{A}$ is replaced by a new i.i.d. $\mathbf{A}(t)$ at each iteration t, where $A_{ij}(t) \sim N(0, 1/m)$; (ii) The corresponding observation becomes $\mathbf{y}^t = \mathbf{A}(t)\mathbf{x} + \mathbf{w}$; (iii) The last term in the update equation for $\mathbf{r}^t$ is eliminated. We thus get the following dynamics:

$$\mathbf{x}^{t+1} = \eta(\frac{u_t}{1+u_t}\tilde{\mathbf{x}} + \frac{1}{1+u_t}(\mathbf{x}^t + \mathbf{A}(t)^T\mathbf{r}^t); \theta_t), \tag{A.1}$$

$$\mathbf{r}^t = \mathbf{y}^t - \mathbf{A}(t)\mathbf{x}^t. \tag{A.2}$$

Eliminating $\mathbf{r}^t$, the first equation becomes:

$$\begin{aligned}\mathbf{x}^{t+1} &= \eta(\frac{u_t}{1+u_t}\tilde{\mathbf{x}} + \frac{1}{1+u_t}(\mathbf{A}(t)^T\mathbf{y}^t + (\mathbf{I} - \mathbf{A}(t)^T\mathbf{A}(t))\mathbf{x}^t; \theta_t) \\ &= \eta(\mathbf{x} + \frac{u_t}{1+u_t}(\tilde{\mathbf{x}} - \mathbf{x}) + \frac{1}{1+u_t}(\mathbf{A}(t)^T\mathbf{w} + \mathbf{B}(t)(\mathbf{x}^t - \mathbf{x})); \theta_t),\end{aligned} \tag{A.3}$$

where $\mathbf{B}(t) = \mathbf{I} - \mathbf{A}(t)^T\mathbf{A}(t)$.

Since the large system limit is assumed here, similar to [32], q_t^2 in Sec. 3.3.3 can be approximated by $\lim_{n\to\infty} \|\mathbf{x}^t - \mathbf{x}\|_2^2 / n$. It can be shown using the central limit theorem that $\mathbf{B}(t)(\mathbf{x}^t - \mathbf{x})$ converges to a vector with i.i.d. normal entries, and each entry has zero mean and variance q_t^2/δ. In addition, the entries of $\mathbf{A}(t)^T\mathbf{w}$ have zero mean and variance of σ^2, and they are independent of $\mathbf{B}(t)(\mathbf{x}^t - \mathbf{x})$. Therefore, each entry of the vectors in the argument of η in Eq. (A.3) converges to $X_0 + \xi_t Z$ with $Z \sim \mathrm{N}(0, 1)$ independent of X_0, and

$$\xi_t^2 = \left(\frac{u_t}{1+u_t}\right)^2 \sigma_s^2 + \left(\frac{1}{1+u_t}\right)^2 (\sigma^2 + \frac{1}{\delta}q_t^2). \tag{A.4}$$

On the other hand, by Eq. (A.3), each entry of $\mathbf{x}^{t+1} - \mathbf{x}$ converges to $\eta(X_0 + \xi_t Z; \theta_t) - X_0$. Therefore

$$q_{t+1}^2 = \lim_{n\to\infty} \frac{1}{n} \left\|\mathbf{x}^{t+1} - \mathbf{x}\right\|_2^2 = \mathbb{E}\{[\eta(X_0 + \xi_t Z; \theta_t) - X_0]^2\}. \tag{A.5}$$

From Eq. (A.4) and Eq. (A.5), we can obtain the state evolution in Eq. (3.28).

This is a heuristic proof, more rigorous proof can be achieved following the proof in [10].

A.2 Proof of Proposition 3.4.1

In this part, we prove Prop. 3.4.1, which studies the bound of the MSE of the GENP-AMP in the (ρ, δ) plane.

Proof. Consider $p_0 \in \mathcal{F}_{\delta\rho}$, $\sigma^2 = 1$ and let $\alpha^*(\delta, \rho) = \alpha^{\pm}(\delta\rho)$ minimax the MSE. To simplify the notation, we define

$$\begin{aligned} \Psi(q^2, u; p) &= \Psi(q^2, u, \delta, \sigma = 1, \sigma_s, \alpha^*, p) \\ &= mse(npi(q^2, u, 1, \sigma_s, \delta); p, \alpha^*). \end{aligned} \tag{A.6}$$

Then, by the definition of fixed point, we get

$$\begin{aligned} q_*^2 &= \Psi(q_*^2, u^*; p), \\ u^* &= \frac{1 + \frac{q_*^2}{\delta}}{\gamma_s^2}. \end{aligned}$$

Using the scale invariance, we have $mse(\sigma^2; p, \alpha^*) = \sigma^2 mse(1; \tilde{p}, \alpha^*)$, where $\tilde{p}$ is a rescaled probability measure, $\tilde{p}\{x \cdot \sigma \in B\} = p\{x \in B\}$. For $p \in F_{\delta\rho}$, we have $\tilde{p} \in F_{\delta\rho}$ as well. Therefore,

$$\begin{aligned} q_*^2 &= mse(npi(q_*^2, u^*, 1, \sigma_s, \delta); p, \alpha^*) \\ &= mse(1; \tilde{p}, \alpha^*) \cdot npi(q_*^2, u^*, 1, \sigma_s, \delta) \\ &\leqslant M^{\pm}(\delta\rho) \cdot npi(q_*^2, u^*, 1, \sigma_s, \delta) \end{aligned}$$

Hence,

$$\frac{q_*^2}{npi(q_*^2, u^*; 1, \sigma_s, \delta)} \leqslant M^{\pm}(\delta\rho),$$

where we use the fact that $\sigma = 1$ and $\gamma_s = \sigma_s$.

By the definition of npi in Eq. (3.25), we have

$$\frac{q_*^2}{\left(\frac{u^*}{1+u^*}\right)^2 \gamma_s^2 + \left(\frac{1}{1+u^*}\right)^2 (1 + \frac{q_*^2}{\delta})} \leqslant M^{\pm}(\delta\rho).$$

Replacing u^* by (3.29), we get

$$q_*^2 \leqslant \frac{-G(\delta,\rho,\gamma_s^2) + \sqrt{G(\delta,\rho,\gamma_s^2)^2 + 4\delta\gamma_s^2 M^{\pm}(\delta\rho)}}{2} \tag{A.7}$$

where $G(\delta,\rho,\gamma_s^2) = \delta\gamma_s^2 + \delta - \gamma_s^2 M^{\pm}(\delta\rho)$.

It is easy to verify that the phase transition boundary only exists when $\gamma_s^2 = \infty$ from the inequality above. If we let $(\gamma_s^2+1)\delta < \gamma_s^2 M^{\pm}(\delta\rho)$, $G(\delta,\rho,\gamma_s^2)$ in the right hand side of Eq. (A.7) is positive. In such case, if γ_s^2 goes to ∞, then $\delta < M^{\pm}(\delta\rho)$, we can get $q_*^2 \leqslant \infty$, i.e., the mean square error is unbounded, corresponding to the classical AMP phase transition boundary.

To prove the second part of Prop. 3.4.1, we make a specific choice $\bar{p}$ of p, and fix a small constant $c > 0$.

Now for $\varepsilon = \delta\rho$, define $h = h^{\pm}(\varepsilon, c) \cdot \sqrt{\mathrm{NPI}^*}$. Let $\bar{p} = (1-\varepsilon)\delta_0 + (\varepsilon/2)\delta_{-h} + (\varepsilon/2)\delta_h$, similar to (3.8). Denote $q_*^2 = q_*^2(\bar{p})$ the highest fixed point corresponding to the signal distribution. Again, by the scale invariance, we have

$$\begin{aligned} q_*^2 &= mse(npi(q_*^2, u^*, 1, \gamma_s, \delta); \bar{p}, \alpha^*) \\ &= mse(1; \tilde{p}, \alpha^*) \cdot npi(q_*^2, 1, \gamma_s, \delta), \end{aligned}$$

where $\tilde{p}$ is a scaled probability measure, and $\tilde{p}\{x \cdot \sqrt{npi(q_*^2,1,\gamma_s,\delta)} \in B\} = \bar{p}\{x \in B\}$. Since $q_*^2 \leqslant M^*$, we have $\mathrm{npi}(q_*^2, 1, \gamma_s, \delta) \leqslant \mathrm{NPI}^*$ and hence

$$\frac{h}{\sqrt{\mathrm{npi}(q_*^2,1,\gamma_s,\delta)}} = h^{\pm}(\varepsilon,c) \cdot \sqrt{\frac{\mathrm{NPI}^*}{\mathrm{npi}(q_*^2,1,\gamma_s,\delta)}} > h^{\pm}(\varepsilon,c).$$

Note that $\mathrm{mse}(q; (1-\varepsilon)\delta_0 + (\varepsilon/2)\delta_{-x} + (\varepsilon/2)\delta_x, \alpha)$ increases monotonically in $|x|$. Recall that $p_{\varepsilon,c} = (1-\varepsilon)\delta_0 + (\varepsilon/2)\delta_{-h^{\pm}(\varepsilon,c)} + (\varepsilon/2)\delta_{h^{\pm}(\varepsilon,c)}$ is nearly-least-favorable for the minimax problem. Consequently,

$$\mathrm{mse}(1; \tilde{p}, \alpha^*) \geqslant \mathrm{mse}(1; p_{\delta\rho,c}, \alpha^*) = (1-c) \cdot M^{\pm}(\delta,\rho).$$

By the scale-invariant property, we conclude that

$$\frac{q_*^2}{\mathrm{npi}(q_*^2,1,\gamma_s,\delta)} \geqslant (1-c) \cdot M^{\pm}(\delta\rho).$$

Then, we can get the inequality

$$\begin{aligned} (q_*^2)^2 + [\delta(\gamma_s^2+1) - (1-c)M^{\pm}(\delta,\rho)\gamma_s^2]q_*^2 \\ - (1-c)M^{\pm}(\delta\rho)\gamma_s^2\delta \geqslant 0. \end{aligned}$$

Therefore,

$$\text{fMSE}(\alpha^*;\delta,\rho,1,\gamma_s^2,\bar{p}) \geqslant \frac{-[\delta(\gamma_s^2+1)-(1-c)M^{\pm}(\delta,\rho)\gamma_s^2]}{2} + \frac{\sqrt{[\delta(\gamma_s^2+1)-(1-c)M^{\pm}(\delta,\rho)\gamma_s^2]^2+4(1-c)M^{\pm}(\delta\rho)\gamma_s^2\delta}}{2},$$

where $\text{fMSE}(\alpha;\delta,\rho,\sigma,\gamma_s^2,p)$ is the equilibrium formal MSE for GENP-AMP (λ, τ_s) for the large system framework [32].

As $c>0$ is arbitrary, we conclude

$$\sup_{p\in F_{\delta\rho}} \text{fMSE}(\alpha^*;\delta,\rho,1,\gamma_s^2,p) \geqslant \frac{-[\delta(\gamma_s^2+1)-M^{\pm}(\delta,\rho)\gamma_s^2]}{2} + \frac{\sqrt{[\delta(\gamma_s^2+1)-M^{\pm}(\delta,\rho)\gamma_s^2]^2+4M^{\pm}(\delta\rho)\gamma_s^2\delta}}{2}.$$

Also, following the same procedure as Prop. 4.2 in [32], it can be shown that $M^* = \inf_{\alpha} \sup_{p\in F_{\delta\rho}} \text{fMSE}(\alpha;\delta,\rho,\sigma=1,\gamma_s^2,p)$.

The last part of Prop. 3.4.1 can be proven by simply substituting the fixed point results in the second part of Prop. 3.4.1 for the ones in Eq. (3.27). □

A.3 Proof of Proposition 3.5.1

In this part, we prove Prop. 3.5.1, which provides an accurate estimation of the variance of the prior $\tilde{\mathbf{x}}$, *i.e.*, σ_s^2. This is an important step of the parameterless GENP-AMP.

Proof. From the definition of the GENP $\tilde{\mathbf{x}}$, we get

$$\begin{aligned}
\sigma_s^2 &= \mathbb{E}[(\tilde{X}-X_0)^2] \\
&= E[(\tilde{X}-X_{\text{pos}}-X_0+X_{\text{pos}})^2] \\
&= \underbrace{E[(\tilde{X}-X_{\text{pos}})^2]}_{(a)} + \underbrace{E[(X_0-X_{\text{pos}})^2]}_{(b)} \\
&\quad - 2\underbrace{E[(\tilde{X}-X_{\text{pos}})(X_0-X_{\text{pos}})]}_{(c)}
\end{aligned} \tag{A.8}$$

where X_{pos} is the estimated sparse signal by GENP-AMP based on an postulated variance $\sigma^2_{\text{s-pos}}$. $\tilde{X}$ and X_{pos} can be explicitly expressed as follows.

$$\begin{aligned}
&\tilde{X} = X_0 + e,\ e\sim N(0,\sigma_s^2) \\
&X_{\text{pos}} = \eta(X_0+\sigma_* Z;\theta),\ Z\sim N(0,1),
\end{aligned} \tag{A.9}$$

where σ_*^2 is the variance of the unthresholded estimator in the last iteration of GENP-AMP.

Next, we look at each part of Eq. (A.8). Part (c) can be rewritten as

$$E[(\tilde{X} - X_{\text{pos}})(X_0 - X_{\text{pos}})] = E[(X_0 - X_{\text{pos}})^2] + E[e(X_0 - X_{\text{pos}})]. \tag{A.10}$$

Thus Eq. (A.8) becomes

$$\sigma_s^2 = E[(\tilde{X} - X_{\text{pos}})^2] - E[(X_0 - X_{\text{pos}})^2] - 2E[e(X_0 - X_{\text{pos}})]. \tag{A.11}$$

If $\sigma^2_{\text{s-pos}}$ is set to ∞, GENP-AMP degrades to AMP, which does not use $\tilde{\mathbf{x}}$. This implies that a perfect candidate of X_{pos} is the signal recovered by AMP, X_{AMP}. Therefore, the two Gaussian noises $\sigma_* Z$ and e are uncorrelated. As a result, $E[e(X_0 - X_{\text{AMP}})] = 0$, and σ_s^2 can be further represented as

$$\sigma_s^2 = E[(\tilde{X} - X_{\text{AMP}})^2] - E[(X_0 - X_{\text{AMP}})^2]. \tag{A.12}$$

Part (a) can be rewritten as $E[(\widehat{X} - \eta(\widehat{X} + \sigma_* Z - e; \theta))^2]$ This term can exactly be seen as a denoising operator. According to the large system limit [32], when n is sufficiently large,

$$E[(\tilde{X} - X_{\text{AMP}})^2] \approx \frac{\|\tilde{\mathbf{x}} - \mathbf{x}_{\text{AMP}}\|_2^2}{n}. \tag{A.13}$$

Next, $E[(X_0 - X_{\text{AMP}})^2]$ can be estimated by the method proposed in [73], inspired by the SURE theory. According to Theorem 4.3 and Theorem 4.7 in [73], it can be predicted by $\lim_{N\to\infty} \frac{\widehat{r}^t(\tau^t)}{N}$ when $t \to \infty$, where t is the inner iteration index of AMP. Usually it will converge in a few iterations.

Summarizing the analyses above, we can prove Prop. 3.5.1. □

Appendix B

Proofs in Chapter 4

In this appendix, we prove that the condition in Corollary 4.3.3, *i.e.*, $M(\varepsilon_1|\eta)$ is a concave function of ε_1, holds for the piecewise constant family in Eq. (4.3).

We start by defining a special family of distributions for simple sparse signals:

$$\mathcal{F}^{SS*}_{n_1,\varepsilon_1} \equiv \left\{ \upsilon_{n_1} : \mathbb{E}_{\upsilon_{n_1}} \left\{ \|\mathbf{s}[2:n_1]\|_0 \right\} \leqslant n_1\varepsilon_1 \right\}, \tag{B.1}$$

where $\mathbf{s}[2:n_1]$ refers to the subvector of a signal $\mathbf{s}$ from the second entry to the last entry.

Consider a signal $\mathbf{x}$ in the piecewise constant signal family $\mathcal{F}^{PC}_{n_1,\varepsilon_1}$ in Eq. (4.3), and define $\mathbf{s}$ as follows.

$$\mathbf{s} = [x[1], x[2] - x[1], \ldots, x[n_1] - x[n_1 - 1]]^T.$$

It is clear that $\mathbf{s} \sim \upsilon_{n_1}$, where $\upsilon_{n_1} \in F^{SS*}_{n_1,\varepsilon_1}$. Therefore, a bijection relationship holds between $\mathcal{F}^{PC}_{n_1,\varepsilon_1}$ and $\mathcal{F}^{SS*}_{n_1,\varepsilon_1}$ because every signal generated from a distribution in $\mathcal{F}^{PC}_{n_1,\varepsilon_1}$ is paired with exactly one signal from $\mathcal{F}^{SS*}_{n_1,\varepsilon_1}$, and every signal from $\mathcal{F}^{SS*}_{n_1,\varepsilon_1}$ is paired with exactly one signal from $\mathcal{F}^{PC}_{n_1,\varepsilon_1}$. As a result, the proof in [28] for the concavity of $M(\varepsilon_1|\eta)$ for block-sparse signals is applicable to the piecewise constant family. However, the proof in [28] (at the end of Page 3406) was very brief. Therefore, we include the following details for completeness.

The goal of the concavity proof is to show that

$$M(q\varepsilon_1 + (1-q)\varepsilon_2|\eta) \geq qM(\varepsilon_1|\eta) + (1-q)M(\varepsilon_2|\eta). \tag{B.2}$$

First, from Eq. (4.2) and (4.3), if a distribution $\upsilon_1 \in \mathcal{F}_{n_1,q\varepsilon_1+(1-q)\varepsilon_2}$, then we have $\upsilon_1 = q\upsilon_2 + (1-q)\upsilon_3$, where $\upsilon_2 \in \mathcal{F}_{n_1,\varepsilon_1}$ and $\upsilon_3 \in \mathcal{F}_{n_1,\varepsilon_2}$ because any measure in $\mathcal{F}_{n_1,q\varepsilon_1+(1-q)\varepsilon_2}$ can be written as a convex combination of measures in $\mathcal{F}_{n_1,\varepsilon_1}$ and measures in $\mathcal{F}_{n_1,\varepsilon_2}$ [28]. Next, note that $M(\varepsilon_1|\eta)$ in Eq. (4.12) is obtained by tuning the denoising parameters to minimize the MSE of the least favorable distribution in the family. Eq. (B.2) can be proved by combining the two facts because each term on the right-hand side can be tuned independently to minimize its own least favorable MSE, whereas there is only one set of tuning parameters on the left-hand side, leading to larger minimax MSE.