Ph.D. DISSERTATION

A STUDY OF SPARSITY-AWARE WIRELESS COMMUNICATION USING COMPRESSED SENSING AND DEEP LEARNING

압축센싱과 딥러닝을 이용한 희소인지 무선통신

BY

Wonjun Kim

AUGUST 2021

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

A STUDY OF SPARSITY-AWARE WIRELESS COMMUNICATION USING COMPRESSED SENSING AND DEEP LEARNING

압축센싱과 딥러닝을 이용한 희소인지 무선통신

지도교수 심 병 효
이 논문을 공학박사 학위논문으로 제출함

2021년 8월

서울대학교 대학원

전기 정보 공학부

김원준

Wonjun Kim의 공학박사 학위 논문을 인준함

2021년 8월

위 원 장:	최 완
부위원장:	심 병 효
위 원:	이 경 한
위 원:	김 성 륜
위 원:	최 준 원

Abstract

The new wave of the technology revolution, named the fourth industrial revolution, is changing our daily life dramatically. These days, unprecedented services and applications such as driverless vehicles and drone-based deliveries, smart cities and factories, remote medical diagnosis and surgery, and artificial intelligence-based personalized assistants are emerging. Communication mechanisms associated with these new applications and services are way different from traditional communications in terms of latency, energy efficiency, reliability, flexibility, and connection density. Further, when the wireless environments and networks are becoming more and more complicated, it is very difficult to come up with simple yet accurate analytic expression. Since the current radio access mechanism cannot support these diverse services and applications, a new approach to deal with these relentless changes should be introduced.

In the first part of the dissertation, we study the sparse vector transmission (SVT) based on the compressed sensing (CS) technique for the low-latency short-packet transmission. The key idea of SVT is to transmit short pieces of information after the sparse vector transformation. One distinctive feature of SVT over the conventional transmission scheme is that positions as well as symbols can be used to convey the information. Using the principle of CS, we decode the packet using a small number of resources. From the performance analysis and numerical evaluations, we demonstrate that the proposed SVT scheme achieves a significant reduction in the physical-layer latency over the conventional systems. Based on the SVT mechanism, we introduce two SVT schemes for the time division duplex (TDD) and the vehicle-to-everything (V2X) systems. First, we propose an approach to support a low latency TDD access, called channel-aware sparse transmission (CAST). By encoding a grant signal in a form of sparse vector and then decoding it with a small number of early arrived samples, up-

link access latency can be reduced dramatically. Second, we present the SVT scheme, called partial sample transmission (PST), for the low-latency V2X sidelink transmission. By using the fact that only small number of samples are required in the decoding process, we can reduce the receiving vehicle's processing latency substantially. In particular, we employ an entirely new decoding technique based on a deep learning (DL) in the PST decoding.

In the second part of the dissertation, we turn our attention to the DL-based wireless communications, especially for the active user detection (AUD) in massive machine-type communications (mMTC) scenarios. Basically, DL-based wireless systems are distinct from the conventional systems in two main respects: data-driven training and end-to-end learning. Instead of following the analytical avenue, the DL model approximates the complicated, often highly nonlinear, relationship between the input dataset and the desired output without human intervention. In the proposed DL-based AUD (D-AUD), by feeding the training data to the properly designed deep neural network, the neural network learns the nonlinear mapping between the received non-orthogonal multiple access (NOMA) signal and indices of active devices. As long as we train the deeply stacked hidden layers using a proper loss function and the backpropagation mechanism, the trained deep neural network can handle the whole AUD process, achieving an accurate detection of the active users. From our simulations, we demonstrate that the D-AUD scheme is very effective in the highly-overloaded mMTC scenarios.

keywords: compressed sensing, deep learning, wireless communication, sparse vector transmission.

student number: 2016-20876

Contents

Abstract **i**

Contents **iii**

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Background 1

1.1.1 Basics of Compressed Sensing 2

1.1.2 Basics of Deep Learning 3

1.2 Contribution and Organization 5

1.3 Notation 9

2 Sparse Vector Transmission for Ultra Low-latency Communications **10**

2.1 Introduction 11

2.2 Principle of Sparse Vector Transmission 12

2.3 Sparse Vector Transmission 13

2.3.1 System Model 14

2.3.2 SVT Decoding 14

2.4 Numerical Performance Evaluation 17

2.5 Summary 19

3 Channel Aware Sparse Transmission for Ultra Low-latency Communications in TDD Systems **20**

3.1 Introduction 21

3.2 Uplink Access Latency in TDD systems 23

3.3 Channel-aware Sparse Transmission 25

3.3.1 System Description of CAST 25

3.3.2 Encoding Operation in CAST 28

3.3.3 Decoding Process in CAST 31

3.3.4 CAST Performance Analysis 34

3.4 Simulation Results 39

3.5 Summary 44

4 Partial Sample Transmission and Deep Neural Decoding for URLLC V2X System **46**

4.1 Introduction 47

4.2 Receiver Processing Latency in Sidelink Transmission 49

4.3 Partial Sample Transmission 51

4.3.1 System Description of PST 52

4.3.2 PST Decoding 56

4.3.3 D-PST Decoder Architecture 58

4.3.4 D-PST Training 61

4.4 Practical PST Implementation For Low-Latency V2X 63

4.4.1 Basic Principle of PST Decoding 63

4.4.2 Retransmission-less PST 64

4.4.3 Synchronization-free PST 66

4.5 Numerical Results 69

4.6 Summary 71

5 Deep Learning-based Wireless Communication Systems: Design Perspective **73**

5.1 Introduction 74

5.2 Artificial Intelligence-Based Wireless Communications 74

5.2.1 Design Principles of Conventional and AI-based Wireless Systems 74

5.2.2 Learning Techniques for DL-based Wireless Communication . 76

5.3 Issues To Be Considered For DL-based Wireless Communication Systems 79

5.3.1 Training Dataset Acquisition 79

5.3.2 DNN Architecture Design 82

5.4 Summary 88

6 Deep Neural Network Based Active User Detection for Grant-free NOMA Systems **90**

6.1 Introduction 91

6.2 AUD System Model 93

6.3 Deep Neural Network Based AUD 99

6.3.1 D-AUD Architecture 99

6.3.2 D-AUD Training 104

6.3.3 Comments on Complexity 107

6.4 Practical Issues for D-AUD Implementation 109

6.4.1 Training Data Collection 110

6.4.2 Sparsity Estimation 111

6.5 Simulations and Discussions 113

6.5.1 Simulation Setup 113

6.5.2 Simulation Results 115

6.6 Summary 119

7 Conclusion **120**

A Proof of (3.14) **123**

B Proof of (3.18) **128**

C Proof of the computational complexities in Table 6.1 **130**

Abstract (In Korean) **143**

Acknowlegement **145**

List of Tables

2.1 System setup for SVT simulations. 17

3.1 Average latency under two different TDD configuration 44

5.1 Summary of DL Techniques 78

6.1 Comparison of Computational Complexity $(N = 80, m = 40, \alpha = 500, L = 6)$ 107

List of Figures

2.1 System models for SVT scheme. 12

2.2 Illustration of the SVT-based short packet transmission in the TDD systems ($k = 3$). 15

2.3 BLER of URLLC packet transmission ($m = 256$). 18

3.1 Overall description of channel-aware sparse transmission (encoding and decoding) based on compressed sensing technique. The base station encodes the grant information (e.g., user ID, timing offset, and transmission band) into the small number of frequency-domain subcarriers (symbols). After receiving the early measurements $\tilde{\mathbf{y}}$, mobile device can decode the information using the sparse signal recovery algorithm. 21

3.2 An example of the scheduling-based uplink transmission in TDD systems. D and U denote the downlink subframe and uplink subframe, respectively. S is a special subframe required for switching the transmit direction. We assume that the uplink data is generated at the beginning of n-th radio frame. 25

3.3 Column correlation between $\mathbf{a}_{\omega_p}$ and $\mathbf{a}_{\omega_q}$ as a function of index difference $|\omega_p - \omega_q|$ ($N = 1024$). 28

3.4 Illustration of the CAST-based access in the TDD systems. 29

3.5 When $k = 2$, $\Omega = \{\omega_1, \omega_2\}$, $\hat{\Omega} = \{\hat{\omega}_1, \hat{\omega}_2\}$, and $\tau = 2$, success decisions for the exact support identification and τ-close support identification are described : (a) The support identification is failed since $\hat{\omega}_2 \neq \omega_2$. (b) The support identification is successful since $\hat{\omega}_1 \in \{\omega_1 - 1, \omega_1, \omega_1 + 1\}$ and $\hat{\omega}_2 \in \{\omega_2 - 1, \omega_2, \omega_2 + 1\}$. 33

3.6 Comparison between τ-close support identification and conventional (exact) support identification in the first iteration using $\tau = \frac{N}{2m}$ ($N = 1024$ and $k = 12$) 34

3.7 Empirical simulation results and upper bound of the error probability of support identification ($N = 1024$ and $\tau = 2$). 38

3.8 Average access latency for the uplink transmission as a function of SNR ($N = 1024$, $k = 9$, and $\tau = 2$) 39

3.9 Decoding success probability of the proposed CAST scheme as a funtion of m under three different SNRs ($N = 1024$, $k = 6$, and $\tau = 2$). . 40

3.10 CAST performances as a function of m ($N = 1024$, SNR = 3dB, and $\tau = 2$) : (a) Decoding success probability for different sparsity level ($k = 4, 8$ and 12). (b) Average latency for the CAST procedure. . . . 41

3.11 Symbol error rate for various number of received samples ($N = 1024$, $k = 10$, and $\tau = 2$). In these simulations, the quadrature phase shift keying (QPSK) modulation is used. 42

3.12 Block error rate of the CAST scheme and PDCCH using the perfect channel information and the estimated channel information. 43

4.1 Description of the V2X systems supported by the vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) communications. 48

4.2 An illustration of the receiver processing latency T_{Rx} in (a) the RB-based sidelink transmission and (b) partial slot-based sidelink transmission. In both scenarios, the buffering latency T_{buff} accounts for a significant portion of T_{Rx}. 50
4.3 The proposed PST scheme: (a) the block diagram of the PST and (b) the reception latency T_{Rx} of PST. 53
4.4 System model for the proposed PST scheme. 55
4.5 Mutual coherence $\mu(\mathbf{\Phi})$ of the IDFT submatrix as a function of the number of measurements m ($N = 256$). 57
4.6 Detailed architecture of the D-PST decoding scheme. 58
4.7 Description of the sidelink retransmission; (a) An illustration of the sidelink retransmission mechanism. After the decoding failure, the BS allocates the retransmission resources. Then, the transmit UE retransmits the packets and the receiving UE initiates the decoding process. (b) One simple option of the retransmission-less PST. 65
4.8 BLER performance of the proposed PST scheme as a function of SNR. 67
4.9 PST performances for various number of measurements ($m = 24, 32, 48$): (a) BLER performance of PST as a function of SNR. (b) The signaling latency for the PST. 68
4.10 Probability of receiver processing latency to complete the packet transmission (SNR=5 dB). 70
4.11 Support identification performance of the D-PST scheme for various number of hidden layers ($L = 3, 6, 9$). 71

5.1 Design principles of traditional communication system and AI-based communication system. 75
5.2 Illustration of data acquisition strategies: (a) synthetic data generation; (b) GAN-based data generation; (c) MSE performances of the DL-based channel estimator using three distinct strategies. 81

5.3 (a) Exemplary DNN architecture designed for the AoA detection. (b) AoA detection performance of various DNNs as a function of SNR. P_{succ} denotes the success probability which corresponds to the percentage of detected AoAs among all angles. 86

6.1 System model of the mMTC uplink scenario where only a few MTC devices are active. 93
6.2 Block diagram of the proposed D-AUD scheme. 95
6.3 Detailed architecture of the proposed D-AUD. 98
6.4 Description of the ReLU layer. 101
6.5 Dropout neural network model: (a) A standard neural network consists of three hidden layers. All hidden units in hidden layers are activated. (b) After applying the dropout, the activated hidden units are dropped out randomly. 102
6.6 Examples of activation patterns corresponding to the strongly correlated supports Ω_1 and Ω_2. Using the dropout layer, the randomly chosen hidden units are dropped out and the activation patterns for Ω_1 and Ω_2 can be better resolved. 103
6.7 Description of the ensemble network: (a) training phase for independent D-AUD scheme with different training set and (b) ensembling test phase using the independently trained D-AUD schemes. 106
6.8 Validation loss $J_v(\Theta)$ for various number of training samples N_{train} ($k = 4$ and $m = 70$). 110
6.9 P_{succ} as a function of SNR ($N = 100, k = 4, N_d = 7, m = 70$). . . . 114
6.10 P_{succ} as a function of SNR with various overloading factor ($N = 100, k = 4$). 115
6.11 P_{succ} as a function of k with 2 different SNR ($N = 100, N_d = 7, m = 70$). 116

6.12 An example of hyperparameter tuning process: (a) depth of hidden layers, (b) width of hidden layers, (c) batch size, (d) dropout probability, (e) optimizer, and (f) activation function 117

6.13 P_{succ} as a function of SNR in the multi-antenna scenario ($N = 100, k = 4$). 118

A.1 If $|\omega^* - \omega_p| \geq \frac{N}{m}$, there exists a local maximum of $f(|\omega^* - \omega_p|)$ such that $f(|\omega^* - \omega_p|) \leq \frac{1}{m\left|\sin \frac{\pi(2i_{\omega_p}+1)}{2m}\right|}$. For example, if $\frac{N}{m} \leq |\omega^* - \omega_1| \leq \frac{2N}{m}$, then $f(|\omega^* - \omega_1|) \leq \frac{1}{m\left|\sin \frac{\pi(2i_{\omega_1}+1)}{2m}\right|}$. In a similar way, if $\frac{3N}{m} \leq |\omega^* - \omega_2| \leq \frac{4N}{m}$, then $f(|\omega^* - \omega_2|) \leq \frac{1}{m\left|\sin \frac{\pi(2i_{\omega_2}+1)}{2m}\right|}$. 125

Chapter 1

Introduction

1.1 Background

The new wave of the technology revolution, named the fourth industrial revolution, is changing the way we live, work, and communicate with each other. We are now witnessing the emergence of unprecedented services and applications such as driverless vehicles and drone-based deliveries, smart cities and factories, remote medical diagnosis and surgery, and artificial intelligence-based personalized assistants. Communication mechanisms associated with these new applications and services are way different from traditional human-centric communications in terms of latency, energy efficiency, reliability, flexibility, and connection density. Therefore, coexistence of human-centric and machine-type services as well as hybrids of these will render emerging wireless environments more diverse and complex. To address diversified services and applications, International Telecommunication Union (ITU) has classified fifth generation (5G) services into three categories: ultra-reliable and low latency communication (URLLC), massive machine-type communication (mMTC), and enhanced mobile broadband (eMBB) [1]. To cope with these new service categories, various performance requirements such as lower latency, higher reliability, massive connectivity, and better energy efficiency have been newly introduced. Since the current radio ac-

cess mechanism and conventional approaches, based on Shannon's channel coding theorem, cannot support these stringent requirements, a new type of transmission approach is required. Before we proceed, we provide the fundamentals of the compressed sensing technique and the deep learning technique.

1.1.1 Basics of Compressed Sensing

Compressed sensing (CS) is a new paradigm to process or recover the sparse signals. This new approach is very competitive option for information processing operations including sampling, sensing, compression, estimation, and detection. Traditional way to acquire and reconstruct analog signals from sampled signals is based on the Nyquist-Shannon's sampling theorem which states that the sampling rate should be at least twice the bandwidth. While these fundamental principles works well, they might be bottleneck of resource overhead and also complexity in a situation where signals are sparse, meaning that the signals can be represented using a relatively small number of nonzero coefficients. At the heart of the CS lies the fact that a sparse signal vector can be recovered from the underdetermined linear system in a computationally efficient way. In other words, a small number of linear measurements (projections) of the signal contain enough information for its reconstruction.

Consider a system model with m-dimensional measurement vector $\mathbf{y}$ and n-dimensional input vector $\mathbf{s}$ given by

$$\mathbf{y} = \mathbf{A}\mathbf{s} \tag{1.1}$$

where $\mathbf{A}$ is the system (sensing) matrix relating the input vector $\mathbf{s}$ and the measurement vector $\mathbf{y}$. If $\mathbf{A}$ is a tall or square matrix, meaning that the dimension of $\mathbf{y}$ is larger than or equal to the dimension of $\mathbf{s}$ $(m \geq n)$, one can recover $\mathbf{s}$ using the conventional techniques (e.g., Gaussian elimination) as long as the sensing matrix is a full rank. However, if the matrix $\mathbf{A}$ is fat $(m < n)$, meaning that the number of unknowns is larger than the dimension of observation vector, it is in general not possible to find out the unique solution since there exist infinitely many possible solutions.

In this pathological scenario where the inverse problem is ill-posed, sparsity of an input vector comes to the rescue. A vector $\mathbf{s}$ is called sparse if the number of nonzero entries is sufficiently smaller than the dimension of the vector. If a vector $\mathbf{s}$ is k-sparse, meaning that there are k nonzero elements in $\mathbf{s}$, the measurement vector $\mathbf{y}$ is expressed as a linear combination of k columns of $\mathbf{A}$ associated with the nonzero entries of $\mathbf{s}$. If the support $\Omega_{\mathbf{s}}$ (set of nonzero indices in $\mathbf{s}$) is known a priori by any chance, then by removing columns corresponding to the zero entries in $\mathbf{s}$, we can convert the underdetermined system into over-determined one and thus can find out the solution using the standard technique. CS theory asserts that as long as the sensing matrix is generated at random, k-sparse vector can be recovered with $m \approx ck\log(n/k)$ measurements (c is a constant). In performing the recovery task, ℓ_1-norm minimization technique and greedy sparse recovery algorithm (e.g., orthogonal matching pursuit (OMP)) have been popularly used (see, e.g., [2]).

It is worth mentioning that underlying assumption in many CS-based studies is that the signal is sparse in nature or can be sparsely represented in a properly chosen basis. Indeed, applications of CS in wireless communications have been mainly on the recovery of naturally sparse signals such as sparse millimeter wave channel estimation in angle and delay domains, angle (DoA/AoD) estimation, and spectrum sensing [2, 3, 4]. Intriguing feature of the works in this dissertation is to purposely transmit the sparse vector to achieve the gain in terms of performance, latency, and energy efficiency.

1.1.2 Basics of Deep Learning

Deep learning is a set of learning methods attempting to model data with complex architectures combining different nonlinear transformations. The elementary bricks of deep learning are the neural networks, that are combined to form the deep neural networks. The neural network can model the nonlinear function f between an input $\mathbf{x}$ and an output $\mathbf{y}$ with respect to the neural network parameters $\boldsymbol{\Theta}$ (i.e., $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\Theta})$). Specifically, a neuron in the neural network is used to approximate the function f_j of

the input $\mathbf{x} = [x_1, \cdots, x_d]$ weighted by a weight vector $\mathbf{w}_j = [w_{j,1}, \cdots, w_{j,d}]$, added by a bias vector $\mathbf{b}_j = [b_{j,1}, \cdots, b_{j,d}]$, and activated by an activation function ϕ. That is, an output element y_j can be expressed

$$y_j = f_j(\mathbf{x}) = \phi(\langle \mathbf{w}_j, \mathbf{x} \rangle + \mathbf{b}_j). \tag{1.2}$$

Some commonly used activation functions are sigmoid function $\phi_{sig}(x) = \frac{1}{1+e^{-x}}$ and rectified linear unit (ReLU) function $\phi_{ReLU}(x) = \max(0, x)$.

The neural network consists of one input layer, one (shallow network) or more (deep network) hidden layers, and one output layer. Based on the connection shape between the neighboring layers, the neural network can be divided into three types: feedforward neural network (FNN), convolutional neural network (CNN), and recurrent neural network (RNN). In the FCN, each neuron (hidden unit) is connected to all neurons of the previous layer so that this network can be applied universally. In the CNN, using the convolution between the convolution filter and the layer input, the hidden unit is locally connected with a part of the previous layer. In doing so, the spatial correlation feature within the convolution filter can be delivered to the next hidden layer. In the RNN, by using the current input together with the output of the previous hidden layer, the timely-correlated feature can be readily extracted.

Once the architecture of the neural network has been chosen, the parameters Θ (e.g., the weights $\mathbf{w}_j$ and biases $\mathbf{b}_j$) have to be estimated (or updated) through the training process. The primary goal of the training phase is to find out the network parameters Θ minimizing the loss function $J(\Theta)$. When $J(\Theta)$ is differentiable, network parameters can be updated by the gradient descent method in each training iteration. Commonly used loss functions include the mean squared error (MSE) and cross entropy (CE). Given output $\mathbf{y}$ and its estimate $\hat{\mathbf{y}}$, MSE can be expressed as

$$J_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \tag{1.3}$$

and CE is defined as

$$J_{\mathrm{CE}}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k, \tag{1.4}$$

where K is the number of class in the classification problem. Using this type of losses, parameters are updated by the stochastic gradient descent (SGD) algorithm and the backpropagation mechanism.

1.2 Contribution and Organization

In this dissertation, we introduce a new approach employing the compressed sensing and deep learning techniques in the wireless communications.

In Chapter 2, we propose new type of short packet transmission scheme referred to as sparse vector transmission (SVT). Key idea of SVT is to transmit the short-sized information after the sparse vector transformation. Using the principle of compressed sensing (CS), we decode the packet using a small number of resources. SVT has a number of advantages over the conventional transmission strategies; it is simple to implement, reduces the transmission latency as well as the encoding/decoding complexity. When the position of a sparse vector is used to encode the information exclusively, decoding can be done without the channel knowledge, saving the pilot transmission overhead and the channel estimation effort. Further, SVT can inherently improve the user identification quality and security. In a nutshell, SVT is a viable solution for massive machine-type communication (mMTC) and URLLC scenarios having many advantages over the conventional packet transmission mechanism.

In Chapter 3, we propose a low latency uplink access scheme suitable for TDD-based URLLC systems. Key feature of the proposed scheme is to transmit the latency sensitive information without waiting for the transmit direction change. To be specific, the base station switches the transmit direction to UL right after sending the URLLC grant signal and hence a mobile device having the latency sensitive information can access the UL resources quickly. To support the fast uplink access, we introduce a new

grant signaling scheme, referred to as *channel-aware sparse transmission* (CAST). Key idea of CAST is to encode the URLLC grant information into a small number of subcarriers in the OFDM symbol. In doing so, we make the frequency-domain OFDM symbol vector *sparse*. This together with the fact that the sensing matrix is a submatrix of the inverse discrete Fourier transform (IDFT) matrix allows us to use the compressed sensing (CS) principle in the decoding of the grant signal. It is now well-known from the theory of CS that an accurate recovery of a sparse vector is guaranteed with a relatively small number of measurements as long as the sensing (measurement) process preserves the energy of an input sparse vector [14]. In our context, this means that a mobile device can accurately decode the grant information with a small number of *early arrived* received samples, which in turn means that UL access latency (latency of transmission and processing of the grant signal) can be reduced dramatically. From the performance analysis in terms of the decoding success probability and also numerical evaluations on the latency sensitive data transmission, we demonstrate that the proposed CAST scheme is very effective and achieves fast uplink access. In particular, in a realistic simulation setup, we observe that CAST achieves more than 80% reduction in the uplink access latency over the 4G LTE and LTE-Advanced TDD systems.

In Chapter 4, we propose a ultra low-latency packet transmission scheme suitable for the mission-critical V2X scenarios. In the proposed scheme, dubbed *partial sample transmission* (PST), a receiving vehicle can decode a packet without waiting for the arrival of a whole packet. Key distinctive feature of PST over the conventional transmission scheme is that the transmit information is converted into a sparse symbol vector and then decoded using a small number of received (time-domain) samples. In the transmit vehicle, information is mapped to the position of the subcarrier vector in the frequency domain. This together with the fact that the submatrix of inverse discrete Fourier transform (IDFT) matrix serves as a system matrix (a.k.a. sensing matrix) allows us to use the compressed sensing (CS) principle in the decoding of sparse trans-

mit vector. Main premise of CS is that an input sparse vector can be recovered with a small number of measurements under the proper sensing mechanism [2]. Beauty of the proposed approach is that while we maintain the OFDM-based mechanism, meaning that with a minimal change in the encoder, a receiving vehicle can accurately decode the PST packet using a small portion of time-domain samples. In particular, by picking *firstly arrived* samples in an OFDM symbol (say only 25% of whole samples), we can significantly reduce the latency associated with the transmission, buffering, and decoding. In the decoding process, instead of using the conventional sparse recovery algorithms, we employ a novel approach based on the deep learning (DL). In the receiving vehicle's perspective, when the number of received samples decreases, the PST system model becomes more underdetermined, resulting in a highly correlated columns in the sensing matrix. To mitigate the decoding error caused by this, we exploit the DL, a learning-based approach to approximate the complicated and nonlinear function [30, 62, 32]. In our decoding scheme, called *deep PST* (D-PST), a deep neural network (DNN) learns the nonlinear mapping between the received signal vector and nonzero position of transmit sparse vector (a.k.a. support). In the test phase (i.e., real decoding phase), by using the learned correlation structure as a prior information, an ambiguity among correlated supports can be better resolved and thus the D-PST scheme identifies the support accurately. Since the learning process is performed offline, time and effort in the training phase does not affect the real operation.

In Chapter 5, we share the essential knowledge and provide useful tips for the design of AI-based wireless communication systems. In specific, we briefly compare the design principles between the conventional wireless systems and the AI-based systems and also discuss how specific communication function is mapped to the deep learning technique. Then, we discuss the three major challenges occurring in the introduction of DL into wireless systems, mainly related to the dataset collection, neural network architecture, and training strategy. For each item, we have provided the learning-based solutions which can be easily implemented in practice. First, in order to collect the

sufficient training data, we provide three options: collection from the actual received signals, synthetic data generation using the analytic system model, and real-like training set generation using generative adversarial network (GAN). Second, in order to design the proper DL model, we provide the useful DNN architecture based on the input characteristics, wireless environments, and system configurations. Third, in order to train the DNN efficiently, we introduce the knowledge distillation and the federated learning techniques.

In Chapter 6, we propose the DL-based active user detection (AUD) for the grant-free NOMA scenario. For an efficient and accurate AUD, we exploit the deep neural network (DNN), a learning-based tool to approximate the complicated and nonlinear function. Over the years, DNN has been successfully applied in numerous applications such as image classification [48], machine translation [63], automatic speech recognition [49], and Go game [47]. Recently, DNN has been also applied to various wireless systems such as multiple-input and multiple-output (MIMO) detection, wireless scheduling, direction-of-arrival (DoA) estimation, and multi-user detection [64, 65, 66]. In these works, DNN is used to learn a desired nonlinear function (e.g., classification and decision) through the training process. In [64], for instance, the DNN structure to learn the mapping between the interference pattern and the optimized scheduling has been proposed. In [65], a DNN architecture for the symbol generation, encoding, and decoding in grant-free NOMA systems has been proposed. In [66], the long short-term memory (LSTM) network performing the channel estimation and data detection in grant-based NOMA systems has been presented. In our framework, DNN learns the complicated mapping between the received NOMA signal and the indices of active users in the transmit signal. To be specific, the proposed AUD scheme, henceforth referred to as deep AUD (D-AUD), learns the sparse structure of device activity using a deliberately designed training dataset. It is now well-known from the *universal approximation theorem* that DNN processed by the deeply stacked hidden layers can well approximate the desired function [67]. In our context, this means that the trained

DNN with multiple hidden layers can handle the whole AUD process, resulting in an accurate detection of the active users.

Chapter 7 summarizes the contributions of the dissertation and discuss the future research directions based on studies of this dissertation.

1.3 Notation

This dissertation uses the following notation: We employ uppercase boldface letters for matrices and lowercase boldface letters for vectors. The operations $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose, respectively. The operators $\odot$ and $\oslash$ denote the Hadamard product and the Hadamard division, respectively. $\mathbb{C}$ and $\mathbb{R}$ denote the field of complex numbers and real numbers, respectively. Also, $\mathbb{N}$ denotes the field of natural numbers. $\|\cdot\|_p$ indicates the p-norm. $\langle \mathbf{a}, \mathbf{b} \rangle$ is an inner product between $\mathbf{a}$ and $\mathbf{b}$. $\Re(c)$ and $\Im(c)$ are the real and imaginary part of c, respectively. $\mathbf{x}_i$ denotes the i-th column of the matrix $\mathbf{X}$ and x_i is the i-th element of the vector $\mathbf{x}$. $\mathbf{X}_\Omega$ is the submatrix of $\mathbf{X}$ that contains the columns specified in the set Ω and $\mathbf{x}_\Omega$ is the vector constructed by picking the elements specified in the set Ω. $\mathbf{A}^\dagger$ is the Penrose-Moore inverse of the matrix $\mathbf{A}$.

Chapter 2

Sparse Vector Transmission for Ultra Low-latency Communications

In this chapter, we introduce a short packet transmission scheme referred to as sparse vector transmission (SVT). Key idea of SVT is to transmit the short-sized information after the sparse vector transformation. Using the principle of compressed sensing (CS), we decode the packet using a small number of resources. SVT has a number of advantages over the conventional transmission strategies; it is simple to implement, reduces the transmission latency as well as the encoding/decoding complexity. When the position of a sparse vector is used to encode the information exclusively, decoding can be done without the channel knowledge, saving the pilot transmission overhead and the channel estimation effort. Further, SVT can inherently improve the user identification quality and security. In a nutshell, SVT is a viable solution for massive machine-type communication (mMTC) and URLLC scenarios having many advantages over the conventional packet transmission mechanism.

The work of Chapter 2 has been published in part in [5].

2.1 Introduction

These days, automated things such as vehicles, drones, sensors, machines, and robots, combined with artificial intelligence (AI) technologies, have found their way into almost every industry. Remarkable growth of business models using autonomous machines is accelerating the need for communication between machines as well as machine to human communications [6]. One important feature of machine-centric communications over the long-standing human-oriented communications is that the amount of information to be transmitted is tiny. For example, information to be exchanged in the autonomous vehicles, robots used for the smart factory, and home appliances is in a form of control and command-type information such as start/stop, turn on/off, move left/right, speed up/slow down, shift, and rotate. Typically, required information bit in these applications is in the range of $10 \sim 100$ bits. Information acquired from the sensors (e.g., temperature, pressure, speed, gas density) is in the order of 10 bits. Also, similar sized packets are used in many feedback or control channels (e.g., ACK/NACK feedback in 4G LTE/5G NR PUCCH [7, 8]).

Crucial observation in these applications is that conventional transmission mechanism is unduly complicated and inefficient, resulting in a waste of resources, transmit power, and processing latency. Shannon's channel coding theorem, governing principle of today's packet transmission, is based on the law of large numbers so that it works properly only when the packet size is sufficiently large. In fact, when the packet length is short, noise introduced by the channel cannot be averaged out properly, degrading the packet reception quality substantially (see, e.g., information theoretic analysis in [9]). Further, in the ultra short-packet transmission regime, size of the non-payload (pilot signals and control data) easily exceeds the payload size so that the cost caused by the non-payload outweighs the cost of payload. In particular, in some applications requiring high reliability (e.g., ultra-reliable and low latency communications (URLLC) in 5G [10]), cost caused by the pilot signaling increases sharply, further degrading the resource utilization efficiency. Without doubt, relying on today's

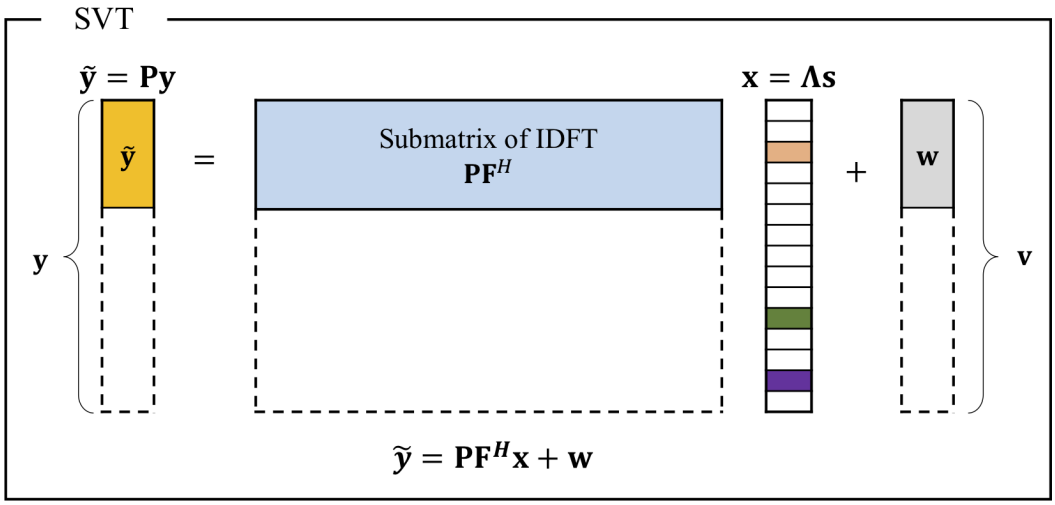


Figure 2.1: System models for SVT scheme.

transmission mechanism would not be efficient due to the waste of resources, large decoding latency, and also expensive operational cost.

2.2 Principle of Sparse Vector Transmission

In this section, we discuss the basic principle of SVT. Basically, there are two options in SVT. To ease our exposition, we discuss the orthogonal frequency division multiplexing (OFDM) system, standard systems for 4G, 5G cellular and WiFi systems, as a baseline. Nevertheless, main principle can be readily extended to different transmission schemes. In the SVT, information is embedded into a small number of subcarriers and then transmitted (see Fig. 2.1). In this case, composite of the channel matrix $\mathbf{H}$ and the IDFT matrix $\mathbf{F}^H$ becomes the sensing matrix so that the time-domain sample vector $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{v} = \mathbf{H}\mathbf{F}^H\mathbf{s} + \mathbf{v}$ becomes the measurement vector. While the symbol decoding in the conventional OFDM systems is initiated after receiving all time-domain samples, sparse vector $\mathbf{s}$ in SVT can be recovered with a small number of time-domain samples using the CS technique. Let $\mathbf{P} = [\mathbf{I}_m \mathbf{0}_(m(n-m))]$ be the

matrix taking early m samples of $\mathbf{y}$, then the vector of the first m measurements is expressed as $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{y}$ (see Fig. 2.1). For instance, if $k = 16, n = 1024,$ and $c = 4$, then only 11% of samples $(m \approx 115)$ is needed to decode $\mathbf{s}$.

Distinctive feature of SVT over the conventional transmission scheme is that positions as well as symbols can be used to convey the information. By way of analogy, one can imagine a process to generate the sparse vector as placing a few balls into the empty boxes. When we try to put k balls in n boxes $(k \leq n)$, we have $\binom{n}{k}$ choices, so that we can encode $\lfloor \log_2 \binom{n}{k} \rfloor$ bits of information into the position of the sparse vector $\mathbf{s}$. Suppose the modulation order is the same for all nonzero positions (say b_s bit per symbol), then kb_s bits can be encoded to the active symbols (symbols in the nonzero positions) so that one SVT block conveys $kb_s + \lfloor \log_2 \binom{n}{k} \rfloor$ bits in total.

There are various options to encode the information in SVT. One simple option is to use both positions and active symbols in the information transmission. Alternatively, one can map the user ID (UID) to the positions and the rest information to the active symbols to elegantly divide the user identification process and information decoding. Yet another option is to embed the message to the positions and the parity bits to the active symbols for the error detection and correction.

2.3 Sparse Vector Transmission

In this section, we discuss the SVT scheme in detail. In contrast to the conventional OFDM systems, SVT transmits the information in a form of a sparse vector and then uses the CS technique to decode the input sparse vector. We first discuss the system model and then explain the SVT decoding and environment-aware user identification, an approach to simplify the user identification process using environmental information.

2.3.1 System Model

As discussed, the system model for SVT is $\mathbf{y} = \mathbf{H}\mathbf{F}^H\mathbf{s} + \mathbf{v}$. Due to the addition of the cyclic prefix, $\mathbf{H}$ is a circulant matrix and thus can be eigen-decomposed by DFT basis. That is, $\mathbf{H} = \mathbf{F}^H\mathbf{\Lambda}\mathbf{F}$ where $\mathbf{F}$ is the DFT matrix and $\mathbf{\Lambda}$ is the diagonal matrix (λ_{ii} corresponds to the frequency channel of i-th subcarrier). The corresponding system model is expressed as $\mathbf{y} = \mathbf{F}^H\mathbf{\Lambda}\mathbf{s} + \mathbf{v}$. Since the supports of $\mathbf{s}$ and $\mathbf{s}$ are the same, by letting $\mathbf{x} = \mathbf{\Lambda}\mathbf{s}$, the system model is converted to $\mathbf{y} = \mathbf{F}^H\mathbf{x} + \mathbf{v}$. Recalling that CS operates with far fewer measurements than the conventional techniques require, a small part of $\mathbf{y}$ is enough to recover $\mathbf{x}$. As a metric to evaluate the sensing matrix, mutual coherence, the largest magnitude of normalized inner product between two distinct columns of sensing matrix (i.e., $\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}$), is widely used [6]. Since the mutual coherence of IDFT submatrix ($\mathbf{P}\mathbf{F}^H$) remains constant as long as we choose consecutive samples, the recovery performance would not be affected by the choice of samples in $\mathbf{y}$. Thus, it is beneficial to use early arrived samples to achieve a reduction in transmission and decoding latencies (see Fig. 2.2). Since the system model $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{F}^H\mathbf{x} + \mathbf{w}(\mathbf{w} = \mathbf{P}\mathbf{v})$ is a standard setting for CS, any sparse recovery algorithm can be employed to decode $\mathbf{x}$ from $\tilde{\mathbf{y}}$.

2.3.2 SVT Decoding

Basically, decoding of SVT consists of two steps; in the first step, support of $\mathbf{x}$ is identified by the sparse recovery algorithm. For example, greedy sparse recovery algorithm identifies one column (position of a vector) of the sensing matrix in each iteration. In our case, a column of $\mathbf{P}\mathbf{F}^H$ that is maximally correlated with the measurement vector is chosen. Once the support $\Omega_{\mathbf{x}}$ of $\mathbf{x}$ (equivalently the support $\Omega_{\mathbf{s}}$ of $\mathbf{s}$) is identified, by removing components associated with the zero entries in $\mathbf{s}$, an over-determined system model to decode the symbol $\mathbf{s}$ can be obtained. In the decoding of $\mathbf{s}$, for example, conventional technique such as the linear minimum mean square error (LMMSE) estimator followed by the symbol slicer can be used.

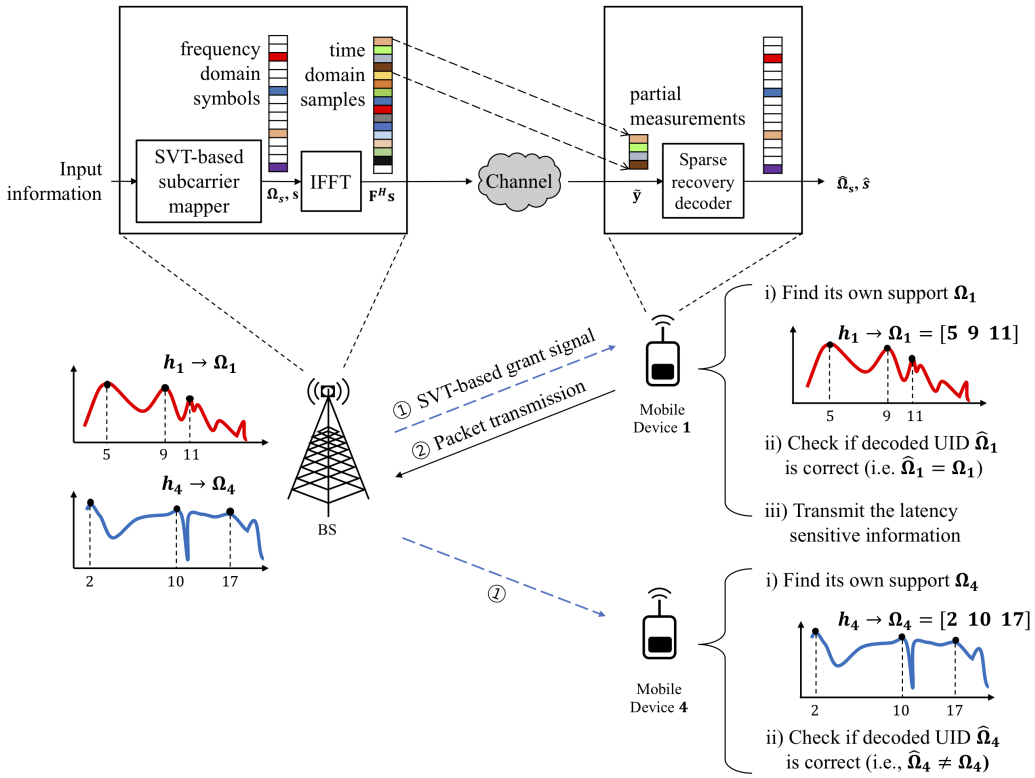


Figure 2.2: Illustration of the SVT-based short packet transmission in the TDD systems ($k = 3$).

In the mapping of the UID to the support, one can consider the environment-aware user identification (EA-UI). Key idea of EA-UI is to use the support $\Omega_{\mathbf{s}}$ derived from the environmental information as a UID. By the environmental information, we mean the information obtained from wireless environments such as channel impulse response (CIR), angle (AoD, DoA), location, delay spread, to name just a few. EA-UI is conceptually similar to the biometric user identification. Biometric identifier, such as iris or fingerprint, intrinsically representing the unique identity of individual's body, can greatly simplify the user identification process. Principle of EA-UI is conceptually similar since the environmental information is reflected in the support of the transmit vector. One simple example, illustrated in Fig. 2.2, is to choose positions of k subcarriers having the largest channel gain as a support and use this as a UID. For example, in TDD systems, base station (BS) can acquire the channel information (and thus UID) of all mobile devices from the uplink pilot signals due to the channel reciprocity (see details in Chapter 3). In the uplink scenario, therefore, BS can identify which mobile device has sent the packet by checking the support (UID) of the received packet. Similarly, in the downlink scenario, mobile device can easily check whether the packet is for itself by comparing the decoded support $\hat{\Omega}_{\mathbf{x}}$ and its own support $\Omega_{\mathbf{x}}$.

EA-UI has several advantages; first, it improves the security since the UID is derived from naturally acquired environmental (channel) information. Second, in many physical channels (e.g., PUSCH or PDSCH in 4G LTE/5G NR [8]), BS sends the data together with UID or after the complicated user scheduling process. Since the EA-UI mechanism separates the user identification and data decoding elegantly, time and effort to decode whole packet just for the user identification purpose can be saved. Indeed, since EA-UI is done by the identification of the support in $\mathbf{x}$, not by the accurate recovery of sparse vector $\mathbf{s}$, support of $\mathbf{x}$ can be recovered using $\tilde{\mathbf{y}}$ and $\mathbf{D} = \mathbf{P}\mathbf{F}^H$ (the submatrix of IDFT). Recalling that $\mathbf{D}$ is independent of the channel, the channel estimation is unnecessary in the support detection. Third, since k is in general very small, sparse recovery algorithm can quickly identify the support. Further, transmission and

Table 2.1: System setup for SVT simulations.

	SVT	PDCCH with convolution and turbo codes (1/3 rate)
System model	5 MHz bandwidth, 15 kHz spacing, and 1 subframe = 1 ms	
FFT size	512 (k=36 in SVT)	
Channel model	i.i.d Rayleigh fading channel	
Number of bit	144	160 (144 for control information and 16 for UID)
Modulation scheme	16-QAM	QPSK

decoding latency can be greatly reduced since only a small fraction of early arrived (time-domain) samples is used for the packet decoding (see Fig. 2.2).

As a final note, one can easily add the error correction capability to EA-UI since the sparsity of subcarriers lends itself to the addition of error correction mechanism. For example, when the correlation between adjacent columns in $\mathbf{D}$ is large, which is true for the submatrix of IDFT, an index of a column adjacent to the correct one might be chosen as a support element incorrectly. Incorrect support element can also be chosen when the supports of BS and UE are slightly different due to the channel estimation error or imperfect channel reciprocity. Since k is small ($k \ll n$) in $\mathbf{s}$, by relaxing the success condition in the support identification, an error can be corrected. Basic idea of this strategy is to replace the selected index $\hat{\omega}$ with the nearest support element $\omega \in \Omega_{\mathbf{x}}$. In other words, as long as the mismatch level $\hat{\omega} - \omega$ is smaller than the properly designed threshold, the error caused by the different supports can be corrected [9].

2.4 Numerical Performance Evaluation

In this subsection, we present the numerical results to evaluate the performance of SVT. In our simulations, the OFDM systems (with 512 subcarriers) under the i.i.d. Rayleigh fading channels are used. As performance measures, the block error rate

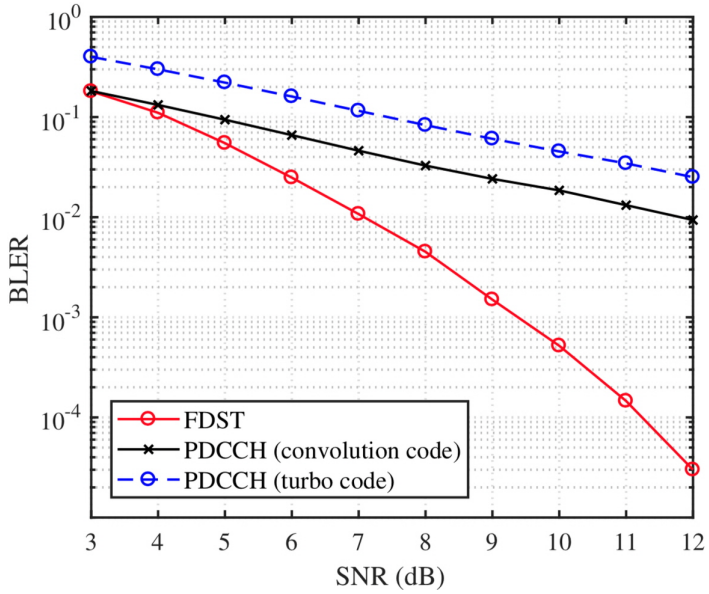


Figure 2.3: BLER of URLLC packet transmission ($m = 256$).

(BLER) and the processing latency (see Table I for detailed setup) are considered. In Fig. 2.3, the BLER of SVT and PDCCH in 4G LTE are compared. In our simulations, sizes of payload and non-payload are set to 144 and 16 bits, respectively. Due to the selective use of good subchannels and also properly designed error correction mechanism, SVT outperforms PDCCH by a large margin, achieving more than 5 dB gain when BLER is 10^{-2}. We next evaluate the average processing latency defined as the sum of the buffering latency and decoding latency for one OFDM symbol. The processing latencies of SVT for $m = 256$ $(73.4\mu s)$ and $m = 128$ $(36.7\mu s)$ are reduced by the factor of 56% and 78% over the LTE PDCCH $(166.8\mu s)$, respectively.

It is worth mentioning that, to decode a packet in 4G LTE/5G NR systems, we need to receive 7 (4G LTE) or 2 (5G NR) OFDM symbols. Whereas, only one symbol (more accurately, small part of a symbol as shown in Fig. 2.2) is enough for the proposed SVT. Additionally, since the required number of samples in the receiver is small, the BS does not need to transmit whole samples and thus the transmit power can be saved considerably.

2.5 Summary

In this chapter, we presented an overview of the sparse vector transmission suitable for the short packet transmission in machine-centric communication scenarios (mMTC and URLLC). We discussed basics of SVT with detailed operations, and also demonstrated the effectiveness of SVT in realistic wireless environments. We observed that SVT is an effective means to transmit the short packet having many advantages over the conventional transmission scheme yet much work remains to be done. For example, we did not elaborate the coding scheme in this work. Perhaps simplest option is to combine the channel coding scheme and SVT mechanically. Better option would be to consider the correlation of the sensing matrix and the quality of channel in the sparse vector generation and decoding. In designing the coding scheme, we can recycle the wasted information. In fact, when b bits are mapped into k positions of n-dimensional vector, $\binom{n}{k} - 2^b$ choices would be wasted. By the deliberate mapping of these choices to the information bits, decoding error probability can be reduced. Also, SVT can simplify complicated transmission procedure. For instance, SVT can be used as a grant signal in the user scheduling process. It can also be used as a grant-free uplink transmission of the short packet.

As communication between machines proliferates, short packet transmission will be more popular and will eventually be a dominating transmission mode in machine-centric wireless systems. We believe that the proposed SVT would serve as a useful tool in the machine communication era.

Chapter 3

Channel Aware Sparse Transmission for Ultra Low-latency Communications in TDD Systems

In this chapter, a low latency uplink access scheme suitable for TDD-based URLLC systems is introduced. Key feature of the proposed scheme is to transmit the latency sensitive information without waiting for the transmit direction change. To be specific, the base station switches the transmit direction to UL right after sending the URLLC grant signal and hence a mobile device having the latency sensitive information can access the UL resources quickly. To support the fast uplink access, we introduce a new grant signaling scheme, referred to as *channel-aware sparse transmission* (CAST). Key idea of CAST is to encode the URLLC grant information into a small number of subcarriers in the OFDM symbol. In doing so, we make the frequency-domain OFDM symbol vector *sparse* (see Fig. 3.1). This together with the fact that the sensing matrix is a submatrix of the inverse discrete Fourier transform (IDFT) matrix allows us to use the compressed sensing (CS) principle in the decoding of the grant signal. It is now well-known from the theory of CS that an accurate recovery of a sparse vector is guaranteed with a relatively small number of measurements as long as the sensing (measurement) process preserves the energy of an input sparse vector [14]. In our

The work of Chapter 3 has been published in part in [11, 12, 13]

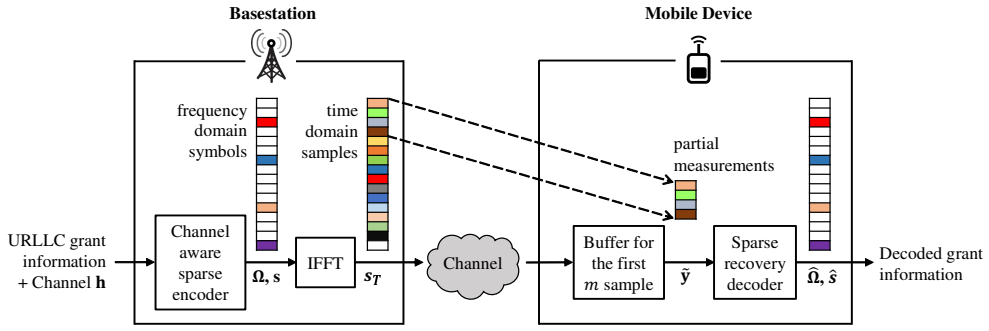


Figure 3.1: Overall description of channel-aware sparse transmission (encoding and decoding) based on compressed sensing technique. The base station encodes the grant information (e.g., user ID, timing offset, and transmission band) into the small number of frequency-domain subcarriers (symbols). After receiving the early measurements $\tilde{\mathbf{y}}$, mobile device can decode the information using the sparse signal recovery algorithm.

context, this means that a mobile device can accurately decode the grant information with a small number of *early arrived* received samples (see Fig. 3.1), which in turn means that UL access latency (latency of transmission and processing of the grant signal) can be reduced dramatically. From the performance analysis in terms of the decoding success probability and also numerical evaluations on the latency sensitive data transmission, we demonstrate that the proposed CAST scheme is very effective and achieves fast uplink access. In particular, in a realistic simulation setup, we observe that CAST achieves more than 80% reduction in the uplink access latency over the 4G LTE and LTE-Advanced TDD systems.

3.1 Introduction

Future mobile communication systems are expected to change our life by supporting wide variety of services and applications such as tactile internet, remote control, smart factories, and driverless vehicles, to name just a few [6]. In order to support these diverse services and applications, new types of requirements other than the classical

throughput requirement are needed [15]. One such requirement is the reduction of latency down to a millisecond level while ensuring reliability of the transmission [16]. To cope with this new requirement and related services, ITU introduced new use case called *ultra-reliable and low latency communications* (URLLC) [1]. Since it is not possible to satisfy the stringent latency requirement by a small makeshift of current 4G LTE systems, an entirely new uplink transmission scheme to support URLLC is required.

Recently, there have been some studies to achieve the latency reduction in the downlink transmission [8, 10, 17, 18]. One simple approach is to transmit an urgent data without any reservations [10]. Also, an approach reserving resources in prior to the data scheduling has been proposed [8]. In [17], an approach to dynamically multiplexing the enhanced mobile broadband (eMBB) and URLLC services has been proposed. Also, a receiver technique to improve the reception quality and latency has been proposed in [18].

In the uplink direction, however, these approaches might not be applicable since the uplink transmission is subject to the complicated handshaking procedure with heavy signaling overhead. Note that the signaling process requires a complicated interplay between the base station and mobile device, and thus it takes quite a bit of time for a mobile device to initiate the data transmission. Indeed, it has been reported that the signaling for LTE scheduling takes more than 7ms even for the best scenario [7].

In the future cellular systems, time division duplexing (TDD) system is expected to be a popular duplexing scheme due to the improved spectrum efficiency, better adaptation quality to asymmetric uplink/downlink traffics, low transceiver cost, and better support of the massive MIMO due to the channel reciprocity [19, 20]. In fact, since the main NR frequency band (e.g., the mid (3.3-3.8GHz) and high (24.25-29.5GHz) bands) is allocated as a TDD mode, supporting the URLLC in TDD system is of great importance [8]. However, satisfying the latency requirement in the TDD systems is far more difficult since the mobile device cannot transmit the data when the subframe

is directed to the downlink (DL). Thus, even though there is an urgent information to transmit, mobile device has no way but to wait until the transmit direction is switched to the uplink (UL). For example, current 4G LTE TDD systems switch from DL to UL with half-frame-level (5ms) or frame-level (10ms) period so that the URLLC requirements cannot be satisfied with an ordinary processing [21, 22]. One can naturally infer from this observation that a direct way to reduce the physical layer latency is to shorten the switching period up to the subframe-level (1ms) period or less. Even in this case, it is not easy to support the short switching period in current 4G LTE systems due to the time-consuming and complicated handshaking process.

3.2 Uplink Access Latency in TDD systems

In this section, we briefly review the latency of TDD-based uplink transmission [23]. As mentioned, scheduling procedure is needed in 4G LTE systems to initiate the UL data transmission. As illustrated in Fig. 3.2[1], a mobile device sends a scheduling request (SR) signal to the base station when there is an information to transmit. After receiving SR, the base station allocates resources and then sends the resource grant (RG) signal to the mobile device. After receiving and decoding the RG signal, a mobile device begins to transmit the information to the base station in the assigned timing (resources).

In the scheduling process, uplink access latency T_{up}, defined as the time duration from the transmission of the grant signal to the initiation of the data transmission, can be expressed as the sum of three distinct latency components (see Fig. 3.2):

$$T_{up} = T_{prop} + T_{proc} + T_{wait}. \tag{3.1}$$

[1]In 4G LTE systems, the length of one radio frame is 10ms. Since one radio frame is divided into 10 subframes, the length of each subframe is 1ms. Also, each subframe consists of 14 OFDM symbols whose length is 66.7μs. Whereas, in the 5G New Radio (NR) systems, multiple numerologies are supported according to the various subcarrier spacing. In this paper, we consider the standard setting of 1ms subframe length with 15kHz subcarrier spacing.

- T_{prop}, called the propagation latency, is the time for a signal to travel from the base station to the mobile device
- T_{proc} is the processing latency for the grant signal
- T_{wait} is the waiting latency for the transmit direction change

Among these latency components, we put our emphasis on the reduction of the major components T_{proc} and T_{wait}[2]. First, T_{proc} can be divided into two components: 1) the buffering latency T_{buff} (the time to receive the grant signal) and 2) the decoding latency T_{dec} (the time to decode the grant information). For example, it takes around 1ms to buffer and decode the grant signal in the current 4G LTE systems [7]. Clearly, this time would be too large to satisfy the URLLC latency requirement[3]. T_{wait} is caused by the periodic direction change in the TDD systems (see Fig. 3.2). Since the current LTE TDD systems switch the transmit direction every 5ms or 10ms, a mobile device should wait until the direction is switched to UL to transmit the urgent data (even if the grant signaling is finished successfully). Since this long switching period cannot satisfy the URLLC latency requirement, an access scheme with ultra short DL-to-UL switching period is needed for the success of URLLC. When the switching period is short, one can notice that T_{proc} would be a bottleneck to support fast UL access. This is because a mobile device has enough time to decode the grant signal in the conventional TDD systems since the switching period (e.g., 5ms in LTE TDD systems) is much larger than T_{proc}. However, when the switching period is very short (e.g, 1ms subframe-level switching), conventional grant signaling mechanism requiring all the received samples (e.g., 1024 samples in one OFDM symbol) to decode the grant information would not be a viable option due to the large T_{proc} (e.g., 1ms in LTE systems). In the following

[2]The propagation latency T_{prop} depends on the distance between the base station and mobile device. Hence, we consider it as a constant when the cell size is given.

[3]In order to support URLLC services, 3rd Generation Partnership Project (3GPP) sets an aggressive requirement that a packet should be delivered with 10^{-5} packet error rate within 1ms transmission period [16].

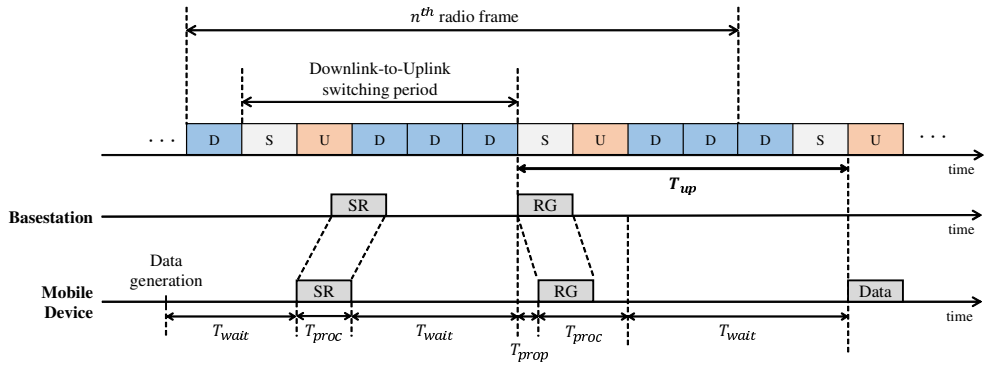


Figure 3.2: An example of the scheduling-based uplink transmission in TDD systems. D and U denote the downlink subframe and uplink subframe, respectively. S is a special subframe required for switching the transmit direction. We assume that the uplink data is generated at the beginning of n-th radio frame.

section, we describe the proposed CAST scheme to reduce T_{proc} of the grant signal.

3.3 Channel-aware Sparse Transmission

3.3.1 System Description of CAST

Fig. 3.1 depicts the block diagram of the proposed CAST scheme. When designing the grant signal $\mathbf{s}$, the base station picks a small number, say k out of N, of subcarriers. For example, if the second and fifth subcarriers are chosen in the grant signal $\mathbf{s}$, then $\mathbf{s} = [0\ s_1\ 0\ 0\ s_2\ 0\ \cdots\ 0]$ (s_1 and s_2 are the symbols) and thus the support of $\mathbf{s}$ is $\Omega = \{2, 5\}$. In the CAST scheme, the granted (scheduled) user ID is encoded to the positions of the selected subcarriers[4] and the remaining grant information (e.g., uplink timing and transmission band) is encoded into the symbols. We will say more about the encoding operation of CAST in Section III-B.

As mentioned, by using only small number of subcarriers, we make the grant sig-

[4]When the base station picks k subcarriers out of N, then there are $\binom{N}{k}$ user IDs in total. In the above example, $\Omega = \{2, 5\}$ is a user ID.

nal vector $\mathbf{s}$ *sparse*. After the inverse fast Fourier transform (IFFT), the time-domain sample vector $\mathbf{s}_t = [s_t(1) \cdots s_t(N)]^T$ is transmitted through the fading channel. The relationship between the transmit sparse grant signal $\mathbf{s}$ and the received time-domain samples $\mathbf{y}$ can be expressed as

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\mathbf{s}_t + \mathbf{v} \\ &= \mathbf{H}\mathbf{F}^*\mathbf{s} + \mathbf{v} \end{aligned} \tag{3.2}$$

where $\mathbf{H} \in \mathbb{C}^{N\times N}$ is the channel matrix, $\mathbf{F}^* \in \mathbb{C}^{N\times N}$ is the IDFT matrix, and $\mathbf{v} \sim \mathcal{CN}\left(0, \sigma_v^2\right)$ is the additive Gaussian noise vector. Since the channel matrix $\mathbf{H}$ is the circulant matrix after removing the cyclic prefix, it can be eigen-decomposed by DFT matrix, i.e., $\mathbf{H} = \mathbf{F}^*\mathbf{\Lambda}\mathbf{F}$ where $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal entry λ_{ii} is the frequency-domain channel response for the i-th subcarrier. Thus, we have

$$\mathbf{y} = (\mathbf{F}^*\mathbf{\Lambda}\mathbf{F})\mathbf{F}^*\mathbf{s} + \mathbf{v} \tag{3.3}$$

$$= \mathbf{F}^*\mathbf{\Lambda}\mathbf{s} + \mathbf{v} \tag{3.4}$$

$$= \mathbf{F}^*\mathbf{x} + \mathbf{v} \tag{3.5}$$

where $\mathbf{x} = \mathbf{\Lambda}\mathbf{s}$. It is worth mentioning that the supports of $\mathbf{s}$ and $\mathbf{x}$ are the same (i.e., nonzero positions of $\mathbf{s}$ and $\mathbf{x}$ are the same).

In the context of CS, $\mathbf{x}$ and $\mathbf{F}^*$ serve as the input vector and sensing matrix, respectively. Since $\mathbf{F}^*$ preserves the signal energy of $\mathbf{x}$, by using properly chosen sparse recovery algorithm, the sparse vector $\mathbf{x}$ can be readily recovered from $\mathbf{y}$ with a small number of measurements. Interestingly, this means that we only need a small number of *early arrived* samples in $\mathbf{y}$ to decode the grant informations. The corresponding partial measurement vector $\tilde{\mathbf{y}} \in \mathbb{C}^{m\times 1}(m \ll N)$ constructed from early arrived samples can be expressed as

$$\tilde{\mathbf{y}} = \mathbf{\Pi}\mathbf{y} \tag{3.6}$$

$$= \mathbf{\Pi}\mathbf{F}^*\mathbf{x} + \tilde{\mathbf{v}} \tag{3.7}$$

$$= \mathbf{A}\mathbf{x} + \tilde{\mathbf{v}} \tag{3.8}$$

where $\mathbf{\Pi} = [\mathbf{I}_m \ \mathbf{0}_{m\times(N-m)}]$ is the matrix to select the first m samples among N time-domain samples, $\tilde{\mathbf{v}} = \mathbf{\Pi}\mathbf{v}$ is the modified noise vector, and $\mathbf{A} = \mathbf{\Pi}\mathbf{F}^*$ is the partial IDFT matrix consisting of the first m consecutive rows of $\mathbf{F}^*$.

As mentioned, the grant information is conveyed from both subcarrier indices and symbols and thus the decoding process is divided into two steps: 1) support identification to find out the nonzero positions of $\mathbf{s}$ vector and 2) symbol detection in nonzero positions. First, for the decoding of the granted user ID, a mobile device needs to identify the support of $\mathbf{x}$, which is done by the sparse recovery algorithm [24, 25]. After identifying the support Ω, a mobile device decodes the remaining grant information by detecting the symbol vector $\hat{\mathbf{s}}_\Omega$. Note that, after removing the components associated with the non-support elements in (3.8), the system model can be converted into the overdetermined system model ($m > k$). For example, if $\Omega = \{2, 5\}$, then the system model in (3.8) is simplified to $\tilde{\mathbf{y}} = \begin{bmatrix}\mathbf{a}_2 & \mathbf{a}_5\end{bmatrix} \begin{bmatrix} x_2 \\ x_5 \end{bmatrix} + \tilde{\mathbf{v}}$. In detecting symbols x_2 and x_5, conventional technique such as the linear minimum mean square error (LMMSE) estimator followed by the symbol slicer can be used.

The benefits of CAST can be summarized as follows. First and foremost, support identification for the decoding of the grant signal $\mathbf{s}$ is done with a small number of time-domain samples. When compared to the conventional signaling mechanism in which all received samples are needed to decode the grant information, buffering latency T_{buff} can be reduced by the factor of m/N. For example, if $m = 128$ and $N = 1024$, then T_{buff} would be reduced by the factor of $1/8$[5]. Second, a channel information is unnecessary in the support identification process. Recall that the sensing matrix $\mathbf{A}$ in (3.8) is constructed only by the submatrix of IDFT matrix and what we need to do is to find out the nonzero positions of $\mathbf{x} = \mathbf{\Lambda}\mathbf{s}$, not the actual values. Thus, we do not need the channel information in the support identification process.

[5]Based on the principle of CS, an accurate recovery of the sparse vector is possible as long as $m \geq ck \log N$ where c is the scaling constant ($c \approx 4$ as a ballpark number [14]). For instance, when $N = 1024$ and $k = 3$, one can readily apply CS technique with $m \approx 120$ measurements.

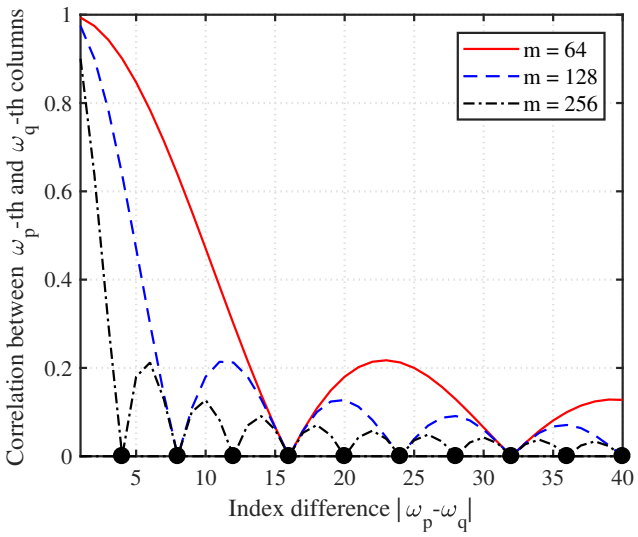


Figure 3.3: Column correlation between $\mathbf{a}_{\omega_p}$ and $\mathbf{a}_{\omega_q}$ as a function of index difference $|\omega_p - \omega_q|$ ($N = 1024$).

Third, the implementation cost and the computational complexity of CAST is very low. In particular, since the sparsity k is small[6] and also known to the mobile device, one can decode the grant information using a simple sparse recovery algorithm such as orthogonal matching pursuit (OMP) [26]. We will show in the next subsections that by choosing nonzero positions deliberately, support identification can be finished in just two iterations.

3.3.2 Encoding Operation in CAST

Since the decoding of the grant signal is done by the support identification, accurate identification of the support is of great importance for the success of CAST. In general, when the system matrix is generated at random, the support identification performance

[6]The size of grant information excluding the user ID would be tiny for most of URLLC scenarios [10]. Hence, the small number k of subcarriers is enough to convey the information. For example, when packet consists of 16 bits for grant information and 64 bits for user ID (RNTI), then we can use $N = 1024$ and $k = 8$ subcarriers with the QPSK modulation.

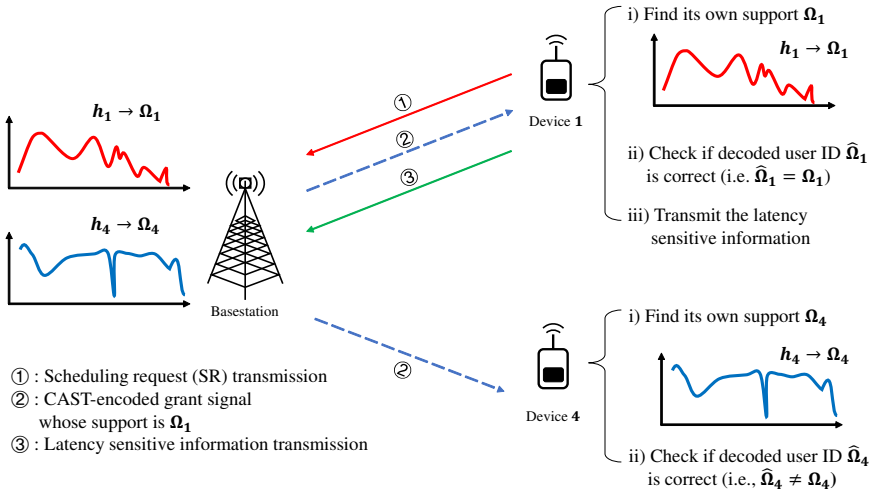


Figure 3.4: Illustration of the CAST-based access in the TDD systems.

would not be affected by the choice of support. In the CAST scheme, however, the system matrix is constructed from IDFT matrix and the sparse vector $\mathbf{x} = \mathbf{\Lambda s}$ is the product of the frequency-domain channel $\mathbf{\Lambda}$ and the sparse grant signal $\mathbf{s}$ so that both *system matrix* and *channel state* affect the decoding performance.

First, support identification performance depends heavily on the channel state. For example, if a selected subcarrier s_i undergoes a deep fading in the frequency-selective channel (i.e., $\lambda_{ii} \approx 0$), then an accurate identification of the nonzero position $x_i = \lambda_{ii} s_i$ would not be possible. Since the DL channel information can be derived from the UL channel estimation via the channel reciprocity in TDD systems [20], it would be desirable to choose indices of subcarriers having the highest subchannel gains as support elements (i.e., $\Omega = \arg\max_{|\Omega|=k} \|\mathbf{h}_\Omega\|_2$). In doing so, one can reduce the chance of the decoding failure significantly.

Second, the support identification performance depends also on the correlation between columns in the system matrix $\mathbf{A}$. In many greedy sparse recovery algorithms, such as OMP, an index of a column in $\mathbf{A}$ that is maximally correlated to the partial measurement $\tilde{\mathbf{y}}$ is chosen as an estimate of the support element [26]. Therefore, if

two columns of $\mathbf{A}$ are strongly correlated and only one of these is associated with the nonzero values in $\mathbf{x}$, then it might not be easy to distinguish the right column (column associated with the nonzero value) from wrong one in the presence of noise. Fortunately, since all entries of $\mathbf{A} = \mathbf{\Pi F}^*$ are known in advance, we can alleviate this event by considering the column correlation of $\mathbf{A}$ in the support selection. Specifically, let $f(\omega_p, \omega_q)$ be the correlation between ω_p and ω_q-th columns in $\mathbf{A}$, then we have

$$\begin{aligned} f(\omega_p, \omega_q) &= \frac{1}{m} \left| \sum_{l=1}^{m} e^{-j2\pi(\omega_p - 1)(l-1)/N} e^{j2\pi(\omega_q - 1)(l-1)/N} \right| \\ &= \frac{1}{m} \left| \frac{\sin \frac{\pi m(\omega_p - \omega_q)}{N}}{\sin \frac{\pi(\omega_p - \omega_q)}{N}} \right|. \end{aligned} \tag{3.9}$$

Since $f(\omega_p, \omega_q)$ depends only on the absolute difference between ω_p and ω_q, we will henceforth denote it as $f(|\omega_p - \omega_q|)$. One can easily see that columns $\mathbf{a}_{\omega_p}$ and $\mathbf{a}_{\omega_q}$ are (near) orthogonal (i.e., $f(|\omega_p - \omega_q|) \approx 0$) if $|\omega_p - \omega_q| \approx c\frac{N}{m}$ for some integer c (see Fig. 3.3). Thus, by choosing the subcarrier indices from the set of the orthogonal columns in $\mathbf{A}$, accuracy of the support identification can be improved significantly.

In summary, the support selection rule considering the channel state and system matrix is given by

$$\Omega = \arg \max_{|\Omega|=k, \Omega \subseteq \Gamma} \|\mathbf{h}_\Omega\|_2 \tag{3.10}$$

where Γ is the index set of the orthogonal columns. Overall grant procedure can be summarized as follows. First, each and every mobile device finds its own support Ω (user ID) using (3.10). Exploiting the channel reciprocity, the base station can also figure out the user IDs of all mobile devices using (3.10). Second, after receiving SR, the base station transmits the CAST-based grant signal to the desired mobile device. Using a small number of early arrived received samples, the mobile device can decode the grant signal. Specifically, if the decoded support $\hat{\Omega}$ is equivalent to its own support Ω (i.e., $\hat{\Omega} = \Omega$), the grant signal is decoded successfully and thus the mobile device sends the (latency sensitive) information immediately (see Fig. 3.4). The proposed CAST-based access procedure is summarized in Algorithm 1.

Algorithm 1 The proposed CAST-based access

Input: $\mathbf{h} \in \mathbb{C}^N$, $\mathbf{A} \in \mathbb{C}^{m \times N}$, $k \in \mathbb{N}$, $\Sigma = \{1, \cdots, N\}$

1: Mobile device finds its own support Ω and base station selects support of the granted user via the following 3 steps

2: $\omega^* = \arg\max_{\omega \in \Sigma} \|\mathbf{h}_\omega\|_2$ [Select index corresponding to the maximal channel gain]

3: $\Gamma = \{\gamma \in \Sigma \mid f(\gamma, \omega^*) \approx 0\} \cup \{\omega^*\}$ [Design the index set of (near) orthogonal columns]

4: $\Omega = \arg \max_{|\Omega|=k, \Omega \subseteq \Gamma} \|\mathbf{h}_\Omega\|_2$ [Determine Ω corresponding to the k largest channel gains]

5: Base station transmits the CAST-based grant signal $\mathbf{s}$ using Ω

6: Using a small number of early arrived samples, the mobile device decodes the CAST signals

7: After the decoding, a mobile device sends the latency sensitive information immediately

3.3.3 Decoding Process in CAST

Basic Decoding

As mentioned, key operation of the CAST decoding is to find out the support Ω. In other words, main task of decoding is to find k nonzero positions of $\mathbf{x}$ vector from the received vector $\tilde{\mathbf{y}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{v}}$. Note that this setup is common in many CS studies [2]. In our case, by exploiting the orthogonality of the columns associated with nonzero positions of $\mathbf{x}$, we can further simplify the support identification process.

To be specific, in the first iteration, a column maximally correlated with $\tilde{\mathbf{y}}$ is chosen as an estimate of support element $\hat{\omega}_i$. Since columns associated with the support Ω are chosen from the set of orthogonal columns, remaining columns should be orthogonal to the column chosen in the first iteration. In the second iteration, therefore, we choose $k - 1$ best columns among those orthogonal to the firstly chosen column. Thus, in contrast to the conventional greedy sparse algorithm in which k iterations are required,

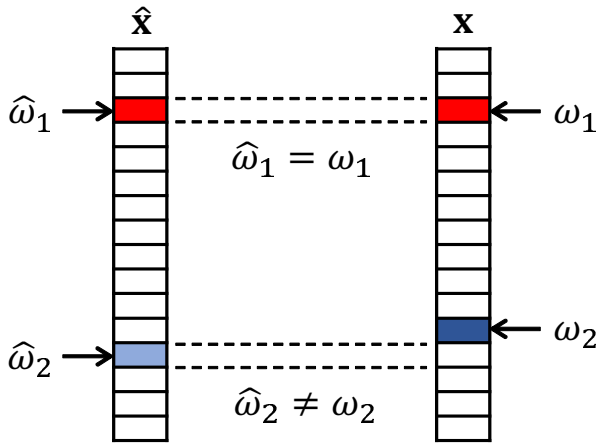
the proposed CAST decoding is finished with only two iterations. After this, a mobile device checks whether it is granted or not by comparing the decoded support $\hat{\Omega}$ and its own support Ω. If $\hat{\Omega} = \Omega$, remaining grant information is obtained by decoding the symbols associated with the support position.

τ-close Support Identification

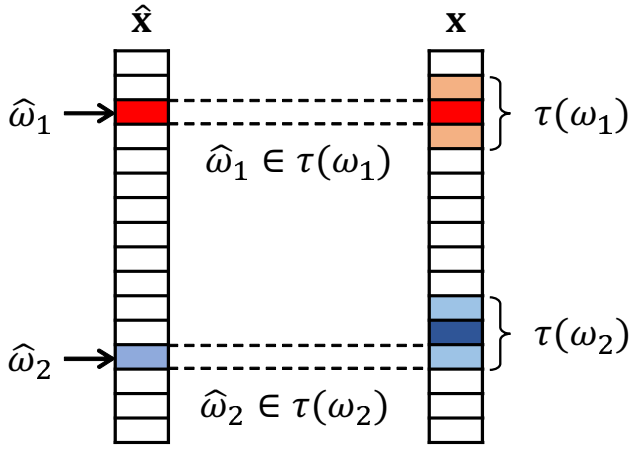
Since the correlation between the adjacent columns in $\mathbf{A} = \mathbf{\Pi F}^*$ is large (see (3.9)), a column adjacent to the correct one might be chosen as a support element by mistake. To avoid this type of mistake, we propose an improved scheme relaxing the success condition in the support identification. Basic idea of the proposed strategy, called τ-close support identification, is to regard the selected index as the correct one if the selected position is close to the true one. That is, a chosen index $\hat{\omega}_i$ is considered as the correct one if it is not too far away from the true index $\omega_i \in \Omega$, i.e., $\hat{\omega}_i \in \{\omega_i - \tau + 1, \cdots, \omega_i, \cdots, \omega_i + \tau - 1\}$ (see Fig. 3.5)[7]. In fact, as long as τ is smaller than the half of the minimum distance between any two orthogonal columns, a chosen index $\hat{\omega}_i$ can be replaced by ω_i and thus the decoding error can be prevented. Since $\mathbf{x}$ is the sparse vector and hence the number of nonzero elements is small, as long as the difference between $\hat{\omega}_i$ and ω_i is small, there would not be any confusion caused by the τ-close support identification.

In Fig. 3.6, we plot the success probability for the first iteration. As discussed, since $k - 1$ columns chosen in the second iteration are orthogonal to the column chosen in the first iteration, successful decoding in the first iteration is crucial for the success of the overall CAST decoding. In our simulations, we compare the CAST decoding performance with and without the τ-close support identification. We observe that the τ-close support identification is very effective and outperforms the conventional sup-

[7] In a practical scenario, due to the channel variation or mismatch in the transmitter and receiver circuitry, the channel reciprocity might not be perfect. Due to this reason, the true support chosen by the mobile device might be slightly different from that chosen by the base station. By using the τ-close support identification, this type of decoding failure can be also prevented.



(a) exact support identification



(b) τ-close support identification

Figure 3.5: When $k = 2$, $\Omega = \{\omega_1, \omega_2\}$, $\hat{\Omega} = \{\hat{\omega}_1, \hat{\omega}_2\}$, and $\tau = 2$, success decisions for the exact support identification and τ-close support identification are described : (a) The support identification is failed since $\hat{\omega}_2 \neq \omega_2$. (b) The support identification is successful since $\hat{\omega}_1 \in \{\omega_1 - 1, \omega_1, \omega_1 + 1\}$ and $\hat{\omega}_2 \in \{\omega_2 - 1, \omega_2, \omega_2 + 1\}$.

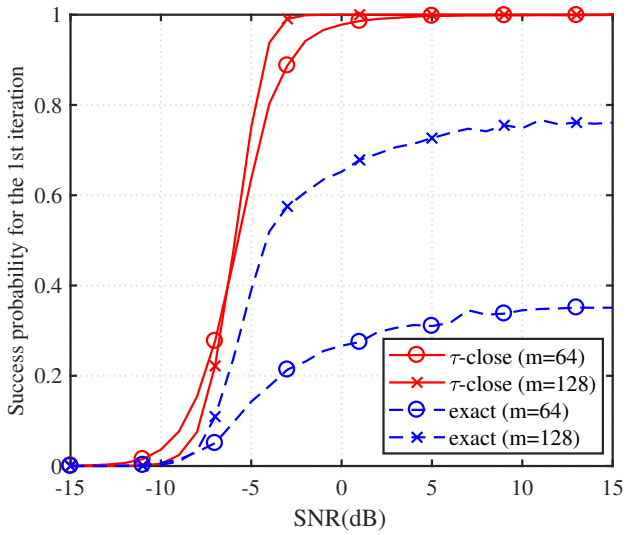


Figure 3.6: Comparison between τ-close support identification and conventional (exact) support identification in the first iteration using $\tau = \frac{N}{2m}$ ($N = 1024$ and $k = 12$)

port identification by a large margin, which will be translated into the gain in decoding performance. For example, when $m = 128$, the τ-close support identification is perfect in most of SNR regimes under test but the conventional support identification performs poor and cannot be better than 0.8. In Algorithm 2, we summarize a refined CAST decoding algorithm incorporating the τ-close support identification.

3.3.4 CAST Performance Analysis

In this subsection, we present the success probability of user identification in the proposed CAST scheme. By the successful user identification, we mean that all the true support elements are chosen by the CAST decoding process (i.e., $\hat{\Omega} = \Omega$). As mentioned, one support element is chosen in the first iteration and the remaining $k - 1$ support elements are chosen in the second iteration. Thus, the success probability of

Algorithm 2 The proposed CAST decoding algorithm

Input: $\tilde{\mathbf{y}} \in \mathbb{C}^m$, $\mathbf{A} \in \mathbb{C}^{m\times N}$, $k \in \mathbb{N}$, $\tau \in \mathbb{N}$, $\mathbf{h} \in \mathbb{C}^N$

1: $\hat{\omega}_1 = \arg\max_{\omega} \|\mathbf{a}_{\omega}^{*}\tilde{\mathbf{y}}\|_2$

2: $\Gamma = \{\gamma \in \Sigma \mid f(\gamma, \hat{\omega}_1) \approx 0\}$

3: (**Identification**) Select indices $\{\hat{\omega}_t\}_{t=2,\ldots,k}$ corresponding to $k-1$ largest entries in $\mathbf{A}_{\Gamma}^{*}\tilde{\mathbf{y}}$

4: $\hat{\Omega} = \{\hat{\omega}_1, \hat{\omega}_2, \cdots, \hat{\omega}_k\}$

5: (τ**-close support identification**) Check $|\hat{\omega}_i - \omega_i| < \tau$ for $i \in \{1, \cdots, k\}$

6: **if** τ-close support identification is successful **then**

7: (**Estimation of** $\tilde{\mathbf{s}}_{\Omega}$) $\tilde{\mathbf{s}}_{\Omega} = \arg\max_{\mathbf{u}} \|\tilde{\mathbf{y}} - \mathbf{A}_{\Omega}\mathbf{\Lambda}_{\Omega}\mathbf{u}\|_2$

8: (**Symbol slicing**) $\hat{\mathbf{s}}_{\Omega} = Q(\tilde{\mathbf{s}}_{\Omega})$

9: **end if**

Output: $\hat{\Omega}, \hat{\mathbf{s}}_{\Omega}$

user identification is expressed as

$$\mathrm{P}_{succ} = \mathrm{P}(\hat{\Omega} = \Omega) \tag{3.11}$$

$$= \mathrm{P}(S^1, S^2) \tag{3.12}$$

$$= \mathrm{P}(S^1)\mathrm{P}(S^2 \mid S^1), \tag{3.13}$$

where S^1 is the event that the index chosen in the first iteration is successful and S^2 is the event that $k-1$ indices chosen in the second iteration are successful.

Our main result for the first iteration $\mathrm{P}\left(S^1\right)$ is as follows.

Proposition 1 *The success probability of the first iteration in the CAST decoding satisfies*

$$\mathrm{P}(S^1) \geq \mathrm{P}\left(\|\tilde{\mathbf{v}}\|_2 \leq \sqrt{\frac{\alpha m}{2k}}\,(1-\rho)\,\|\mathbf{h}\|_{\infty}\right), \tag{3.14}$$

where $\|\tilde{\mathbf{v}}\|_2$ *is the* ℓ_2*-norm of the noise* $\tilde{\mathbf{v}}$*,* α *is the desired SNR,* m *is the number of measurements,* $\rho = \sum_{p=1}^{k} \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_p}+1)}{2m}\right|}$ *where* i_{ω_p} *(*$\omega_p \in \Omega$*) depends on the index*

chosen in the first iteration, k is the number of nonzero elements, and $\|\mathbf{h}\|_\infty$ is the maximum channel gain.

Proof: See Appendix A. □ Since the obtained lower bound of $\mathrm{P}\left(S^1\right)$ in (3.14) depends on two random variables $\|\tilde{\mathbf{v}}\|_2$ and $\|\mathbf{h}\|_\infty$, to compute the lower bound of $\mathrm{P}\left(S^1\right)$, we take the expectation of the conditional probability $\mathrm{P}\left(S^1 \mid \|\mathbf{h}\|_\infty\right)$ with respect to the condition $\|\mathbf{h}\|_\infty = r$. That is,

$$\mathrm{P}\left(S^1\right) = \int_0^\infty \mathrm{P}\left(S^1 \mid \|\mathbf{h}\|_\infty = r\right) f_{\|\mathbf{h}\|_\infty}(r)dr \tag{3.15}$$

$$\geq \int_0^\infty \mathrm{P}\left(\|\tilde{\mathbf{v}}\|_2^2 \leq \frac{\alpha m}{2k}\left(1-\rho\right)^2 r^2\right) f_{\|\mathbf{h}\|_\infty}(r)dr \tag{3.16}$$

where $f_{\|\mathbf{h}\|_\infty}(r) = Nre^{-\frac{r^2}{2}}\left(1-e^{-\frac{r^2}{2}}\right)^{N-1}$ [8]. Since $\tilde{\mathbf{v}} \sim \mathcal{CN}(0,1)$, $\|\tilde{\mathbf{v}}\|_2^2$ is a Chi-squared random variable with $2m$ degree of freedom (DoF). Using the cumulative distribution function (CDF) of $\|\tilde{\mathbf{v}}\|_2^2$, we have

$$\mathrm{P}\left(S^1\right) \geq \int_0^\infty \frac{\gamma\left(m, \frac{\alpha m}{2k}\left(1-\rho\right)^2 r^2\right)}{\Gamma(m)} Nre^{-\frac{r^2}{2}}\left(1-e^{-\frac{r^2}{2}}\right)^{N-1} dr, \tag{3.17}$$

where $\Gamma(a)$ and $\gamma(a,b)$ are a complete gamma function and an incomplete gamma function, respectively.

We next present the success probability for the second iteration when the first iteration is successful.

Proposition 2 *The success probability of the second iteration in the CAST decoding satisfies*

$$\mathrm{P}\left(S^2|S^1\right) \geq \left[1 - F\left(1|2,2,\zeta\right)\right]^{(k-1)(m-k)}, \tag{3.18}$$

[8] For analytic simplicity, we use the i.i.d Rayleigh fading channel model for $\mathbf{h}$ [7].

where $F(\cdot)$ is the CDF of the non-central F-distribution[9] *and ζ is the noncentrality parameter depending on the channel realization.*

Proof: See Appendix B. □ From Proposition 1 and Proposition 2, we obtain the final result for P_{succ} as follows.

Theorem 3 *The probability that the CAST-encoded packet is decoded successfully satisfies*

$$\mathrm{P}_{succ} \geq \left[1 - F\left(1|2,2,\zeta\right)\right]^{(k-1)(m-k)} \cdot \int_0^{\infty} \frac{\gamma\left(m, \frac{\alpha m}{2k}\left(1-\rho\right)^2 r^2\right)}{\Gamma(m)} N r e^{-\frac{r^2}{2}} \left(1 - e^{-\frac{r^2}{2}}\right)^{N-1} dr. \tag{3.20}$$

Proof: Using (3.17) and (3.18), we obtain the desired result. □

In order to judge the effectiveness of the obtained lower bound in (3.20), we plot the theoretical bound and empirical simulation results as a function of SNR for m (see Fig. 3.7). In this figure, we plot the error probability of user identification defined as $1 - \mathrm{P}_{succ}$. In our simulations, we compute the empirical averages to approximate the expectations with respect to ρ and ζ. From these results, we observe that the obtained bound is tight, in particular for high SNR regime. In the middle SNR regime, on the other hand, we observe some gap between the theoretical and empirical simulation results. The gap is because the use of 1) an upper bound of column correlation and 2) the inequalities such as triangular inequality and Cauchy-Schwarz inequality. From this figure, we also observe that the success probability increases sharply when the number of measurements m increases. For example, if m is doubled from 128 to 256, we can achieve more than 5 dB gain in performance.

[9]The non-central F-distribution is described by the quotient $(X/n_1)/(Y/n_2)$ with the CDF given by

$$F\left(x|n_1,n_2,\lambda\right) = \sum_{r=0}^{\infty} \left(\frac{\left(\frac{1}{2}\lambda\right)^j}{j!} \exp\left(-\frac{\lambda}{2}\right)\right) I\left(\frac{n_1 x}{n_2 + n_1 x}\Big|\frac{n_1}{2} + j, \frac{n_2}{2}\right) \tag{3.19}$$

where the numerator X has a non-central chi-squared distribution with n_1 degrees of freedom and the denominator Y has a central chi-squared distribution n_2 degrees of freedom.

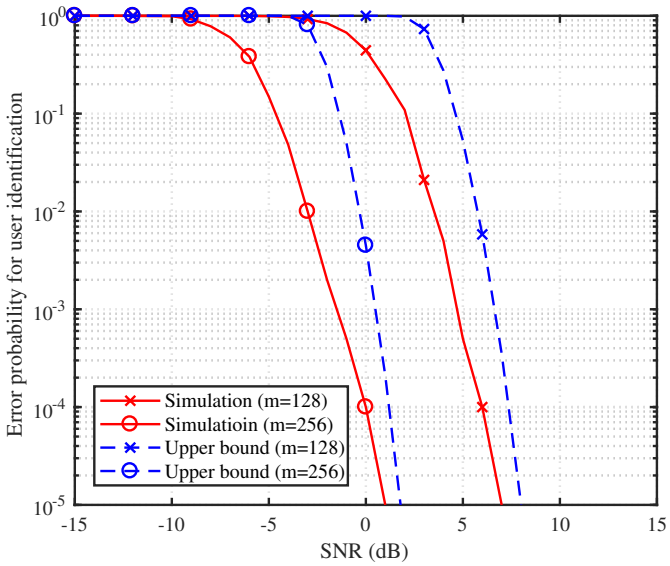


Figure 3.7: Empirical simulation results and upper bound of the error probability of support identification ($N = 1024$ and $\tau = 2$).

In many URLLC applications, latency and reliability are equally important and thus both should be considered in the system design and evaluation [27]. In the proposed scheme, when m increases, the reliability will be improved but the latency will also increase due to the increase of the buffering latency and decoding latency. In Fig. 3.8, we plot the mean access latency required to complete the CAST procedure for different values of m. Note, if either the support identification or symbol detection is failed, the CAST procedure is repeated. We observe that the proposed CAST scheme achieves the low access latency and also good decoding performance. For example, when m is reduced from 1024 to 256, the access latency is reduced by the factor of 35%. However, when m is too small, the access latency is rather increased, in particular for low SNR regime, since in this case the CAST decoding can be failed and hence the entire process needs to be repeated.

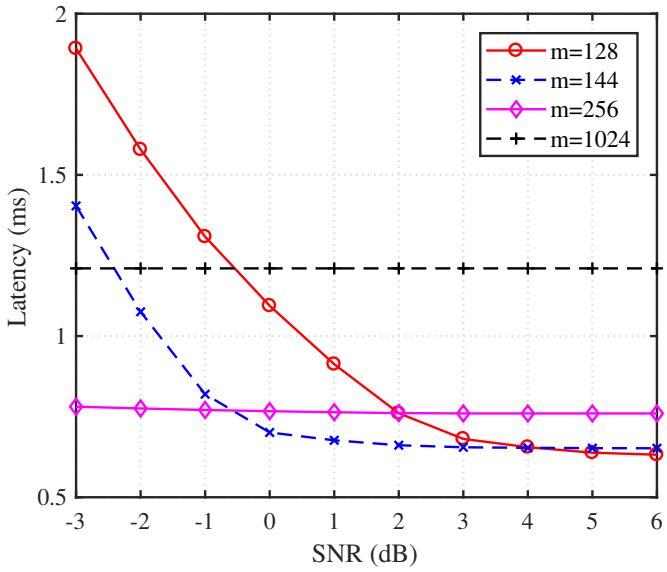


Figure 3.8: Average access latency for the uplink transmission as a function of SNR ($N = 1024$, $k = 9$, and $\tau = 2$)

3.4 Simulation Results

In this section, we present the numerical results to evaluate the decoding performance and access latency of the proposed CAST. In our simulations, we consider the OFDM-based TDD systems with $N = 1024$ subcarriers. As a channel model, we use the i.i.d Rayleigh fading channels. For comparison, we use two different approaches in the support selection. In the first approach, we choose the subcarriers uniformly at random among N subcarriers. In the second approach, we choose the support by the proposed selection rule (Algorithm 1). In the decoding process, we use the proposed decoding algorithm (Algorithm 2) with τ-close support identification ($\tau = 2$). As performance metrics, we use the success probability of support identification, symbol error rate (SER), and also average access latency. The access latency is defined as the sum of the waiting latency T_{wait} and processing time T_{proc} in (3.1).

In Fig. 3.9, we evaluate the success probability of the support identification as a

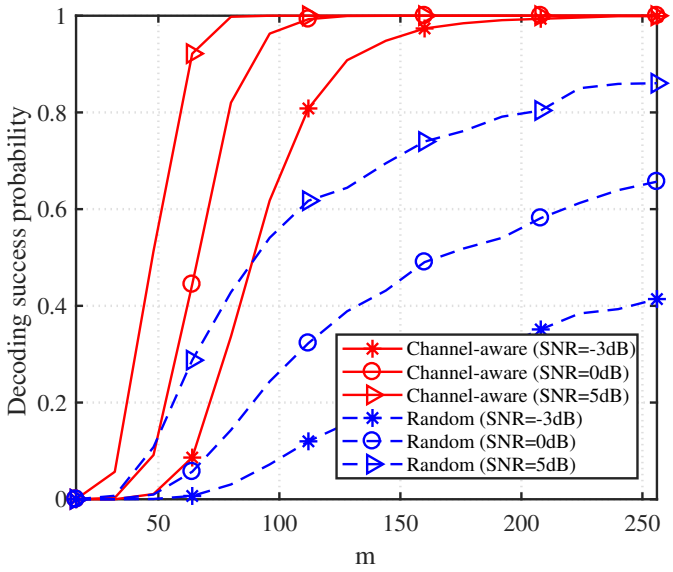
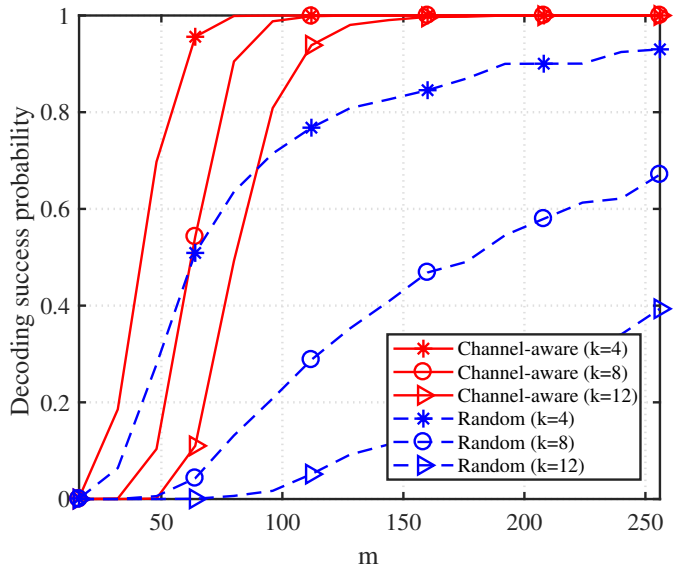


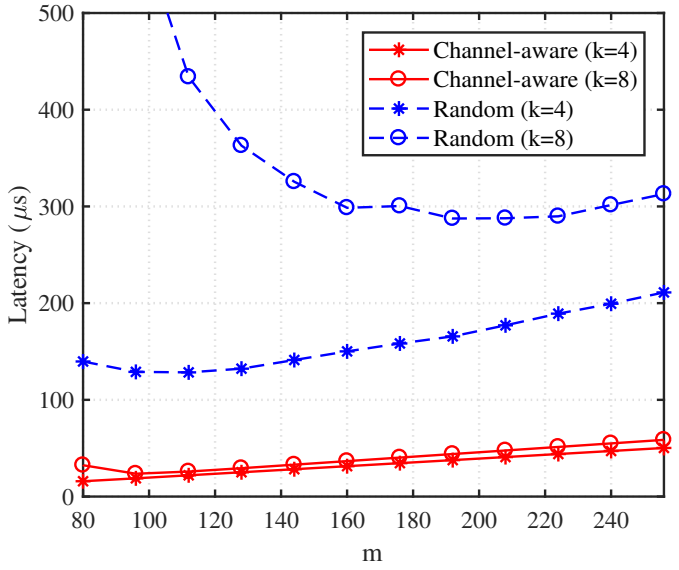
Figure 3.9: Decoding success probability of the proposed CAST scheme as a funtion of m under three different SNRs ($N = 1024$, $k = 6$, and $\tau = 2$).

function of m for various SNRs (SNR = -3dB, 0dB, and 5dB). Simulation results demonstrate that the proposed CAST scheme achieves a significant reduction in the number of received samples. When compared to the conventional signaling mechanism in which all received samples are needed to decode the grant information, CAST requires much smaller number of samples. For example, CAST requires only 7.8% ($m = 80$ at 5 dB) of the received samples, which directly implies that the buffering latency T_{buff} can be reduced by the factor of 92.2% (see Section III.A).

In Fig. 3.10(a), we evaluate the success probability of the support identification for various sparsity levels ($k = 4, 8$, and 12). We observe that only 10% ($k = 4$) and 15% ($k = 12$) of the received samples are needed to decode the grant information. This behavior, however, cannot be achieved in the random support selection approach. For instance, if k increases from 4 to 12, the required number of samples to achieve 40% success probability increases from 38 samples to 75 samples in the proposed support selection rule but that for the random support selection rule increases from 57



(a)



(b)

Figure 3.10: CAST performances as a function of m ($N = 1024$, SNR $= 3$dB, and $\tau = 2$) : (a) Decoding success probability for different sparsity level ($k = 4, 8$ and 12). (b) Average latency for the CAST procedure.

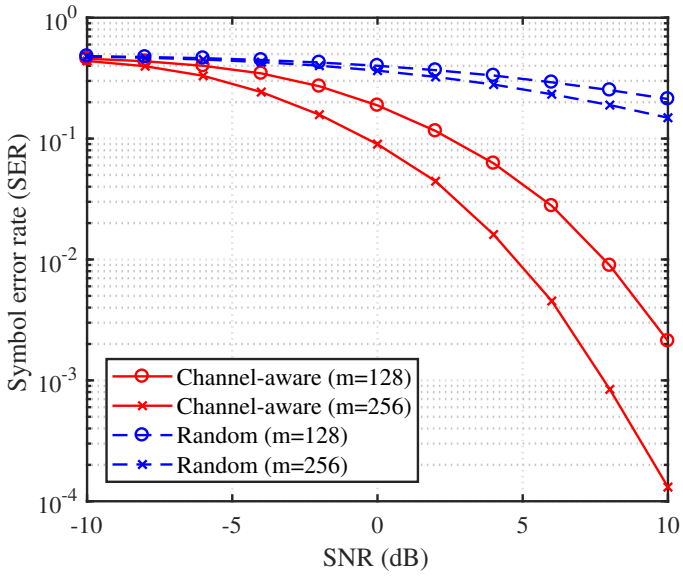


Figure 3.11: Symbol error rate for various number of received samples ($N = 1024$, $k = 10$, and $\tau = 2$). In these simulations, the quadrature phase shift keying (QPSK) modulation is used.

to 256. Also, we investigate the average latency for performing the CAST process (see Fig. 3.10(b)). These results clearly demonstrate that the proposed support selection rule (in Sec III.B) is very effective in reducing the latency. For example, if k increases from 4 to 8, the latency for the proposed support selection rule is about the same but that for the random support selection increases 2 times at $m = 160$.

In Fig. 3.11, we plot the SER performance of the proposed CAST scheme for two different number of measurements ($m = 128$ and 256). We observe that the proposed selection rule outperforms the random selection rule by a large margin. For example, when $m = 256$, the proposed selection rule achieves 10^{-4} SER performance at SNR $=$ 10 dB but the random selection approach cannot achieve this level of reliability even at high SNR.

In order to verify the robustness of CAST in real scenario, we test the block error rate (BLER) of CAST and the physical downlink control channel (PDCCH) in 4G

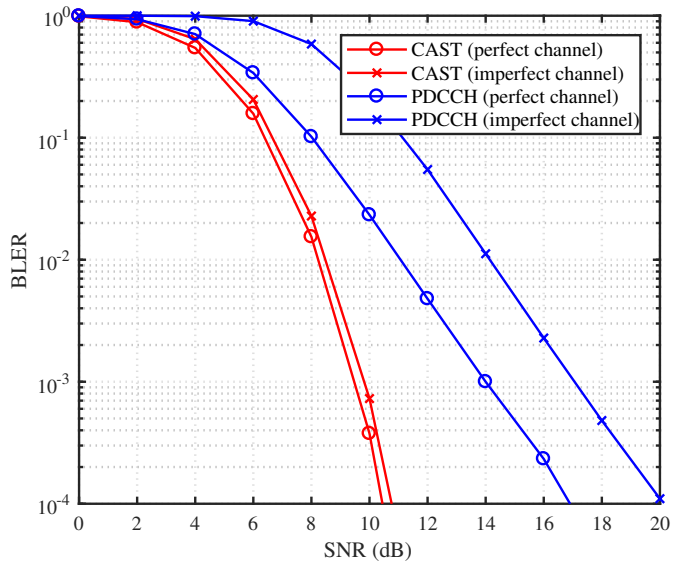


Figure 3.12: Block error rate of the CAST scheme and PDCCH using the perfect channel information and the estimated channel information.

when the channel is estimated. As shown in Fig. 3.12, we observe that the CAST scheme outperforms the PDCCH, achieving more than 6 dB gain over the conventional PDCCH at 10^{-4} BLER point. We also observe that the proposed scheme is insensitive to the channel estimation error. For example, when BLER$= 10^{-4}$, the gap between the perfect channel and imperfect channel for the proposed scheme is less than 1 dB but that for the PDCCH is around 3 dB.

Finally, we evaluate the access latency of CAST-based TDD system in Table. 3.1. In our simulation, we consider the LTE-TDD system (Rel. 13) and minislot-based NR TDD system (Rel. 15)[10] as references. The access latency in (3.1) can be expressed as $T_{up} = T_{wait} + T_{prop} + T_{proc} = T_{wait} + T_{prop} + \left(\frac{m}{f_s} + T_{dec}\right)$ where m is the number

[10]NR TDD system can flexibly schedule the UL data using the mini-slot (2,4 or 7 OFDM symbols) transmission. Using the mini-slot transmission, the switching period of NR TDD systems is shortened significantly and hence quick transmit direction change is possible. In this simulation, we use 2 OFDM symbols as a mini-slot.

Table 3.1: Average latency under two different TDD configuration

	Conventional LTE TDD	**Minislot-based NR TDD**	**CAST-based TDD**
DL:UL=9:1	5.56ms	1.19ms	0.71ms
DL:UL=8:2	3.82ms	1.16ms	0.68ms

of received samples and f_s is the sampling frequency. When carrying out the mini-slot based access and CAST-based access, the base station changes the transmit direction into UL right after sending the grant signal and thus the mobile device can transmit the latency sensitive data without waiting for the periodic transmit direction change (i.e., $T_{wait} \approx 0$). We use two TDD configurations with the different DL-UL ratio (9:1 and 8:2) and generate one URLLC packet in every two subframes. In case of DL:UL=9:1 configuration, the access latency of the CAST-based TDD system (0.71 ms) is reduced by the factor of 87% and 40% over the LTE TDD system (5.56 ms) and NR TDD system (1.19 ms), respectively. In a similar way, the access latency is also reduced by the factor of 82% and 41% for the DL:UL=8:2 configuration. These results demonstrate that the CAST-based access is effective in the URLLC packet transmission. In particular, when compared to the minislot-based NR TDD systems, we observe that the latency reduction obtained from CAST is non-negligible and meaningful. This is because T_{proc} is reduced substantially by using a small number of the received samples and simple decoding algorithm (see Section III.C).

3.5 Summary

In this paper, we proposed the ultra low latency access scheme based on the CAST for URLLC. Our work is motivated by the observation that waiting time to switch the transmit direction and processing time for the grant signal are quite large in TDD systems. The key idea behind the proposed CAST scheme is to transform a URLLC grant

information into the sparse symbol vector and to exploit the sparse recovery algorithm in decoding the sparse signal. As long as the number of subcarriers is small enough and the measurements contain enough information to figure out the support and decode the grant information, accurate decoding of the CAST scheme can be guaranteed. We demonstrated from the numerical evaluations that the proposed CAST scheme is very effective in TDD-based URLLC scenarios. In this paper, we restricted our attention to the URLLC scenario but we believe that there are many interesting extensions worth investigating, such as the diversity support, machine learning-based CAST, and CAST for the FDD systems.

Chapter 4

Partial Sample Transmission and Deep Neural Decoding for URLLC V2X System

This chapter will propose a ultra low-latency packet transmission scheme suitable for the mission-critical V2X scenarios. In the proposed scheme, dubbed *partial sample transmission* (PST), a receiving vehicle can decode a packet without waiting for the arrival of a whole packet. Key distinctive feature of PST over the conventional transmission scheme is that the transmit information is converted into a sparse symbol vector and then decoded using a small number of received (time-domain) samples. In the transmit vehicle, information is mapped to the position of the subcarrier vector in the frequency domain. This together with the fact that the submatrix of inverse discrete Fourier transform (IDFT) matrix serves as a system matrix (a.k.a. sensing matrix) allows us to use the compressed sensing (CS) principle in the decoding of sparse transmit vector. Main premise of CS is that an input sparse vector can be recovered with a small number of measurements under the proper sensing mechanism [2]. Beauty of the proposed approach is that while we maintain the OFDM-based mechanism, meaning that with a minimal change in the encoder, a receiving vehicle can accurately decode the PST packet using a small portion of time-domain samples. In particular, by pick-

The work of Chapter 4 has been published in part in [28, 29].

ing *firstly arrived* samples in an OFDM symbol (say only 25% of whole samples), we can significantly reduce the latency associated with the transmission, buffering, and decoding.

In the decoding process, instead of using the conventional sparse recovery algorithms, we employ a novel approach based on the deep learning (DL). In the receiving vehicle's perspective, when the number of received samples decreases, the PST system model becomes more underdetermined, resulting in a highly correlated columns in the sensing matrix. To mitigate the decoding error caused by this, we exploit the DL, a learning-based approach to approximate the complicated and nonlinear function [30, 62, 32]. In our decoding scheme, called *deep PST* (D-PST), a deep neural network (DNN) learns the nonlinear mapping between the received signal vector and nonzero position of transmit sparse vector (a.k.a. support). In the test phase (i.e., real decoding phase), by using the learned correlation structure as a prior information, an ambiguity among correlated supports can be better resolved and thus the D-PST scheme identifies the support accurately. Since the learning process is performed offline, time and effort in the training phase does not affect the real operation.

4.1 Introduction

With the rapid development of intelligent transportation systems (ITS), a growing number of vehicular applications have emerged to provide an entirely new experience in our daily life [33, 34, 35]. Among various vehicular applications, mission-critical vehicular services such as vehicle platooning, safety alarming, and remote driving play a vital role in the blueprint of the future ITS. These services are accelerating the need for the well-organized vehicle-to-everything (V2X) communications including vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N). For example, in the platooning scenarios where vehicles form a coordinated group with low inter-vehicle spacing, autonomous vehicles share their trajectories and

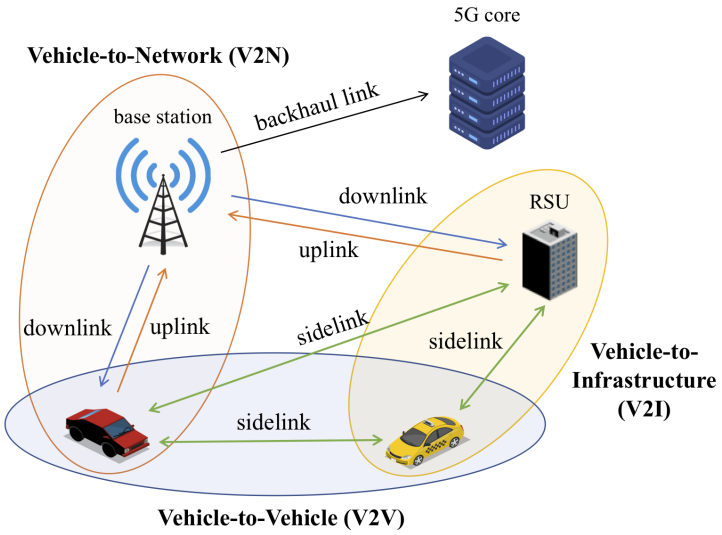


Figure 4.1: Description of the V2X systems supported by the vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) communications.

driving intentions with each other to guarantee the vehicle safety [36]. Since the intervened vehicles should decode the information accurately and quickly, reliability and latency are of great importance to ensure the quality of services (QoS) of V2X system.

To accommodate the emerging services requiring low end-to-end latency with high reliability, ultra-reliable and low-latency communications (URLLC) has been introduced as a new service category in 5G New Radio (NR) [**?**, 10]. In order to support URLLC, 3GPP sets a strict requirement that a packet should be delivered with 10^{-5} packet error rate within 1 msec end-to-end latency [37, 38]. In the current 4G LTE-based V2X systems, referred to as cellular V2X (C-V2X), it is very difficult to satisfy the URLLC requirements since multiple OFDM symbols should be processed to decode a data packet. In fact, a group of 7 symbols spanning 12 subcarriers (0.5 msec $\times$ 180 kHz) called a resource block (RB) is used in LTE as a basic scheduling unit [7]. In order to decode RB, a mobile device has to receive 7 OFDM symbols, which takes 0.5 msec just for the buffering of samples. Since it takes almost 0.5 msec to perform

the control signaling (e.g., PDCCH transmission), it is not possible to satisfy the latency requirement of URLLC with an ordinary receiver processing in 4G LTE [18]. This situation has been relaxed in 5G NR due to the short transmission mode called the *minislot transmission* but it is still not enough to support the mission-critical applications since the time for buffering samples is too large to satisfy the stringent latency requirement (e.g., less than 0.1 msec for the fully-automated driving [36]).

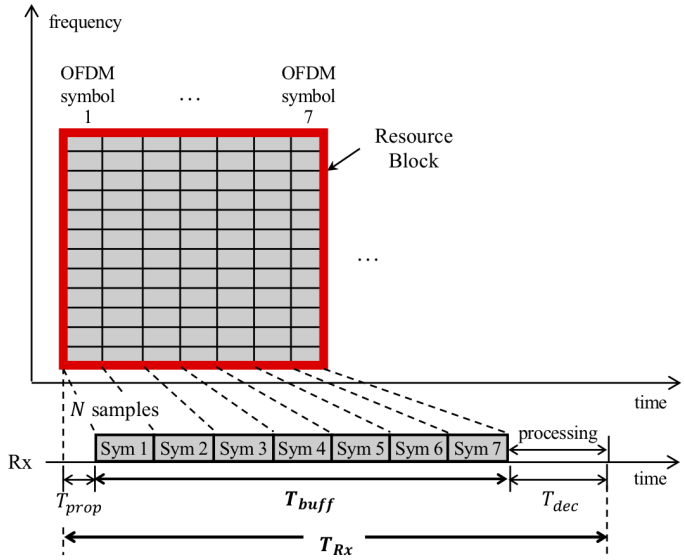
4.2 Receiver Processing Latency in Sidelink Transmission

In this section, we briefly review the receiver processing latency T_{Rx} in the V2X sidelink transmission. In Fig. 4.1, we describe the V2X systems consisting of the V2V network, the V2I network, and the V2N network. In order to send a packet using the sidelink, a UE needs to acquire the control information (e.g., sidelink scheduling information and sidelink allocation information) from the base station (BS). Then, the transmit UE directly delivers the packet to the receiving UE without intervening the BS [36]. In the sidelink transmission, since the complicated network procedures such as the link establishment and the resource allocation are already completed by the core network, a time caused by the physical layer operations becomes a major component in the end-to-end latency.

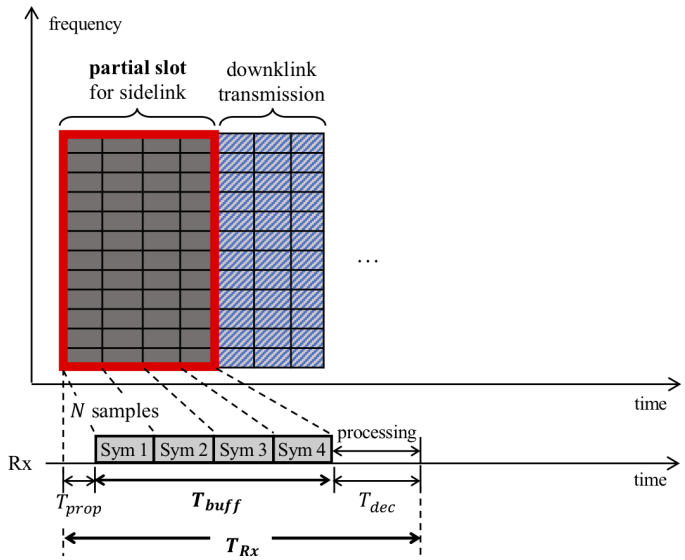
At the receiving UE, a duration from the beginning of the sample transmission to the end of the decoding process can be expressed as the sum of three distinct latency components (see Fig. 4.2):

$$T_{\mathrm{Rx}} = T_{prop} + T_{buff} + T_{dec}. \tag{4.1}$$

- T_{prop} is the propagation latency, which corresponds to the time for a signal to travel from the transmit UE to the receiving UE
- T_{buff} is the time to receive the transmitted signal
- T_{dec} is the time to decode the transmit information



(a)



(b)

Figure 4.2: An illustration of the receiver processing latency T_{Rx} in (a) the RB-based sidelink transmission and (b) partial slot-based sidelink transmission. In both scenarios, the buffering latency T_{buff} accounts for a significant portion of T_{Rx}.

Among these delay components, we primarily focus on the reduction of the buffering latency T_{buff} since T_{buff} is much larger than T_{prop} and T_{dec}. Indeed, when the distance between the transmit UE and the receiving UE is in the order of tens $\sim$ hundreds of meters, T_{prop} becomes very tiny (e.g., 0.1 μs for 30 m spacing). When compared to the latency requirement of the physical layer (i.e., less than 0.1 msec), T_{prop} can be negligible. Also, by using the low-complexity decoding and parallel processing, T_{dec} can be controlled under a few microseconds.

As mentioned, when delivering a packet in a form of RB in LTE C-V2X systems, a receiving UE needs to receive 7 symbols so that T_{buff} equals one slot period (i.e., 0.5 msec) (see Fig. 4.2(a)). Clearly, this time is too large to meet the latency requirement in physical layer. In order to reduce T_{buff} in NR V2X system, a short transmission mode called *partial slot transmission* has been introduced [36]. In the partial slot transmission, only a few symbols in a slot are used for the sidelink and the remaining symbols are reserved for the downlink or uplink transmissions (see Fig. 4.2(b)). In this scheme, a receiving UE buffers the time-domain samples corresponding to the partial slot symbols to initiate the decoding process so that the buffering time T_{buff} can be reduced substantially. For example, when $2 \sim 3$ symbols are used for the partial slot transmission, T_{buff} would be $0.15 \sim 0.3$ msec. One can deduce from this discussion that conventional transmission scheme might not be a viable option to support the mission-critical V2X applications (e.g., accident alarming and emergency response).

4.3 Partial Sample Transmission

In this section, we present the proposed PST scheme. Key feature of the proposed scheme is that the transmit UE encodes the input sidelink information in a form of the sparse OFDM symbol and the receiving UE decodes the information using a partially-buffered samples. While 4G LTE requires 7 OFDM symbols and 5G NR needs 2 symbols for the packet decoding, only small part of one symbol is enough to decode the

packet in PST.

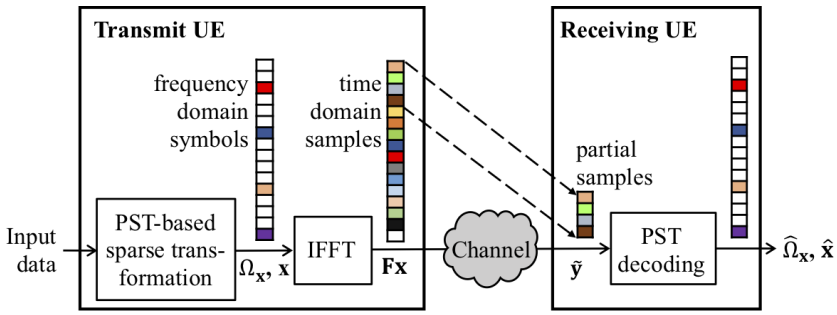
4.3.1 System Description of PST

Fig. 4.3(a) depicts the overall description of the PST scheme. In order to convert the transmit information into a sparse vector $\mathbf{x}$, a transmit UE chooses a small number of subcarriers (say k out of N). As an example, when the first and fourth subcarriers are picked, then $\mathbf{x} = [x_1\ 0\ 0\ x_2\ \cdots\ 0]^T$ where x_i is the i-th symbol in $\mathbf{x}$ and the support (i.e., set of nonzero positions) of $\mathbf{x}$ is $\Omega_{\mathbf{x}} = \{1, 4\}$. Distinctive feature of PST over the conventional transmission scheme is that both *positions* as well as *symbols* are used to convey the information. When we choose k nonzero elements in N positions ($k \ll N$), we have $\binom{N}{k}$ choices and thus $\lfloor \log_2 \binom{N}{k} \rfloor$ bits information can be encoded into the position of $\mathbf{x}$. For simplicity, we assume that the modulation order is the same for all nonzero positions (if QPSK transmission is assumed, then $b_s = 2$ bit per symbol). Then kb_s bits can be encoded to the active symbols (symbols in the nonzero positions) and thus one PST block transmits $\lfloor \log_2 \binom{N}{k} \rfloor + kb_s$ bits in total.

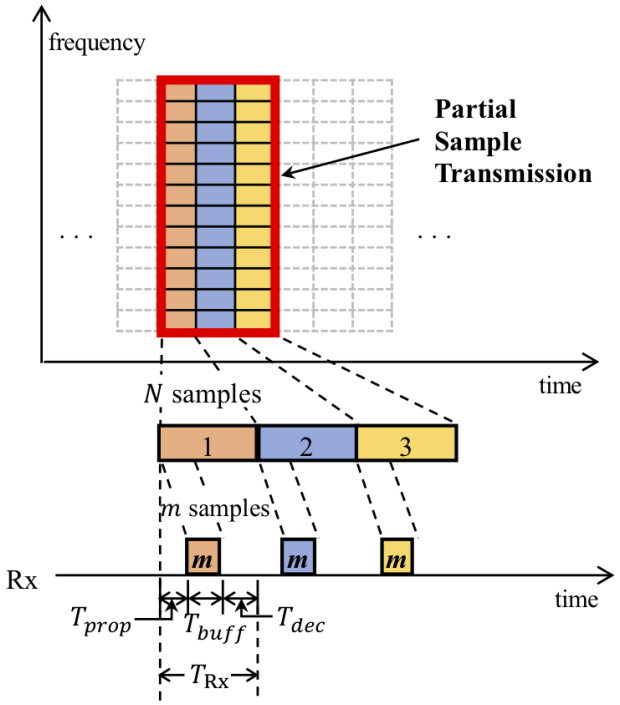
After the sparse transformation, rest operations are the same as normal OFDM operation. After the inverse fast Fourier transform (IFFT), the time-domain sample vector is $\mathbf{x}_t = \mathbf{F}\mathbf{x}$ where $\mathbf{F} \in \mathbb{C}^{N \times N}$ is the IDFT matrix. After adding the cyclic prefix (CP), the relationship between the transmit sparse vector $\mathbf{x}$ and the received time-domain sample vector $\mathbf{y}$ is

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\mathbf{x}_t + \mathbf{n} \\ &= \mathbf{H}\mathbf{F}\mathbf{x} + \mathbf{n} \end{aligned} \tag{4.2}$$

where $\mathbf{H} \in \mathbb{C}^{N \times N}$ is the circulant channel matrix and $\mathbf{n}$ is the additive Gaussian noise vector. Since $\mathbf{H}$ is a circulant matrix, it can be eigen-decomposed by the DFT matrix $\mathbf{F}^*$ (i.e., $\mathbf{H} = \mathbf{F}\mathbf{\Sigma}\mathbf{F}^*$ where $\mathbf{\Sigma}$ is the diagonal matrix whose diagonal entry σ_{ii}



(a)



(b)

Figure 4.3: The proposed PST scheme: (a) the block diagram of the PST and (b) the reception latency T_{Rx} of PST.

corresponds to the frequency channel of the i-th subcarrier). Therefore, we have

$$\mathbf{y} = (\mathbf{F}\boldsymbol{\Sigma}\mathbf{F}^*)\mathbf{F}\mathbf{x} + \mathbf{n} \tag{4.3}$$

$$= \mathbf{F}\boldsymbol{\Sigma}\mathbf{x} + \mathbf{n}. \tag{4.4}$$

Let $\mathbf{s} = \boldsymbol{\Sigma}\mathbf{x}$ be a composite of the sparse vector and the frequency channel, then we have

$$\mathbf{y} = \mathbf{F}\mathbf{s} + \mathbf{n}. \tag{4.5}$$

Note that the supports (sets of nonzero positions) of $\mathbf{x}$ and $\mathbf{s}$ are the same (i.e., $\Omega_{\mathbf{x}} = \Omega_{\mathbf{s}}$).

Based on the CS theory, as long as the sensing mechanism preserves the energy of an input sparse vector, k-sparse vector can be accurately recovered with a small number of measurements $m = ck \log N$ (c is a constant) [2]. In our context, since $\mathbf{s}$ and $\mathbf{F}$ correspond to the sparse vector and the sensing matrix, $\mathbf{s}$ can be recovered from partial measurements of $\mathbf{y}$. In other words, a small portion of the received time-domain samples in $\mathbf{y}$ is enough to decode the transmit information in the receiving UE. Since the decoding performance would not be affected by the choice of partial samples (see Section IV-A), it would be better to use *firstly arrived* samples in $\mathbf{y}$. In doing so, the receiver processing latency T_{Rx} can be reduced significantly (see Fig. 4.3(b)). The corresponding partial measurement vector $\tilde{\mathbf{y}} \in \mathbb{C}^{m \times 1} (m \ll N)$ is

$$\tilde{\mathbf{y}} = \boldsymbol{\Gamma}\mathbf{y} \tag{4.6}$$

$$= \boldsymbol{\Gamma}\mathbf{F}\mathbf{s} + \tilde{\mathbf{n}} \tag{4.7}$$

$$= \boldsymbol{\Phi}\mathbf{s} + \tilde{\mathbf{n}} \tag{4.8}$$

where $\boldsymbol{\Gamma} = [\mathbf{I}_m \ \mathbf{0}_{m \times (N-m)}]$ is the selection matrix to take the first m samples among N time-domain samples, $\tilde{\mathbf{n}} = \boldsymbol{\Gamma}\mathbf{n}$ is the sampled noise vector, and $\boldsymbol{\Phi} = \boldsymbol{\Gamma}\mathbf{F}$ is the submatrix of IDFT constructed from the first m consecutive rows of $\mathbf{F}$ (see Fig. 4.4).

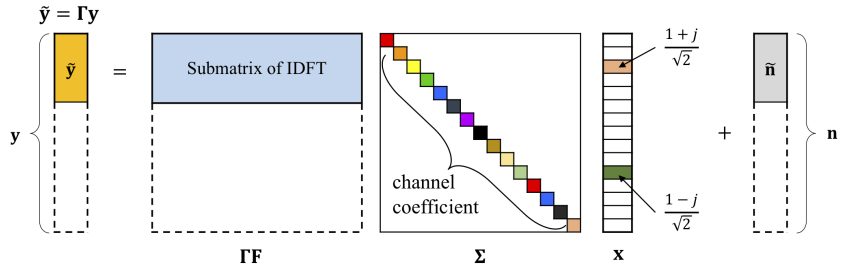


Figure 4.4: System model for the proposed PST scheme.

Since the information is encoded into both subcarrier positions and symbols, two operations (i.e., support identification and symbol detection) are needed for the decoding of the PST packet. First, to find out the nonzero positions of $\mathbf{s}$, a receiving UE needs to identify the support of $\mathbf{s}$, and this job can be done by the sparse signal recovery algorithm [39, 2] (for details, see Section III-B). After identifying the support $\hat{\Omega}$, rest of information can be decoded by detecting the symbol vector $\hat{\mathbf{x}}_{\hat{\Omega}}$. Note, by eliminating the columns corresponding to the zero elements of $\mathbf{s}$ from $\mathbf{\Phi}$, the underdetermined system in (6.5) is converted to the over-determined system $(m > k)$. Using the standard approach such as the linear minimum mean square error (LMMSE) estimator followed by the symbol slicer, one can obtain the symbol estimate.

The advantages of PST can be summarized as follows. 1) First, the decoding process is done with a small number of time-domain samples. When compared to the RB-based (4G LTE) and minislot (5G NR)-based transmission, T_{buff} of the PST scheme is much smaller (more accurately, T_{buff} is less than one symbol period). For example, when the half of the transmitted samples is used in the decoding (e.g., $m = 128$ and $N = 256$), the PST achieves 92% and 75% reduction in T_{buff} over the 4G LTE (using 7 symbols RB) and 5G NR minislot (2 symbols) transmissions. 2) Second, in the support identification, the channel information is unnecessary. This is because the sensing matrix $\mathbf{\Phi}$ in (8) is constructed only by the submatrix of IDFT matrix and what we need to do is to find out the nonzero positions of $\mathbf{s} = \mathbf{\Sigma x}$, not the actual values.

Therefore, in the V2X systems where the vehicles' channels vary rapidly due to the mobility[1], the PST scheme provides more reliable decoding performance. 3) Third, the transmit power can be saved considerably. This is because the required number of samples in the receiver is small ($m \ll N$), a transmit UE does not need to transmit whole samples. For example, if $m = 128$ and $N = 512$, then the transmit power is reduced by 75%. It is worth mentioning that this power saving is beneficial in increasing the battery life of low-power vehicles (e.g., drone consuming 5 Watt).

4.3.2 PST Decoding

As mentioned, the PST-based packet decoding is divided into two parts: *support identification* and *symbol detection*. Since the decoding of the PST scheme is initiated by the support identification, an accurate identification is crucial for the reliable decoding performance. For the support identification, any sparse recovery algorithm can be employed (e.g., OMP, CoSaMP, MMP [39, 40, 41]). In many sparse recovery algorithms, an index set of the columns in $\mathbf{\Phi}$ that are the most correlated to the measurement $\tilde{\mathbf{y}}$ is considered as an estimate of the support. Hence, when two columns in $\mathbf{\Phi}$ are highly correlated and only one of these contributes to $\tilde{\mathbf{y}}$, then it would not be easy to distinguish the right column from wrong one, in particular under the noise and interference. In fact, the support identification performance depends highly on the column correlation of $\mathbf{\Phi}$.

As a metric to evaluate the correlation of the sensing matrix, mutual coherence, defined as the largest magnitude of normalized inner product between two distinct columns of sensing matrix, is widely used [2]. Since all elements of $\mathbf{\Phi}$ are known ($\mathbf{\Phi}$

[1]For example, when the carrier frequency is $f_c = 28$ GHz and the mobile speed is $\nu = 80$ km/h, then the channel coherence time $T_c = \frac{9c}{16\pi\nu f_c} = 0.03$ ms is much smaller than the slot period $T_s = 0.5$ ms.

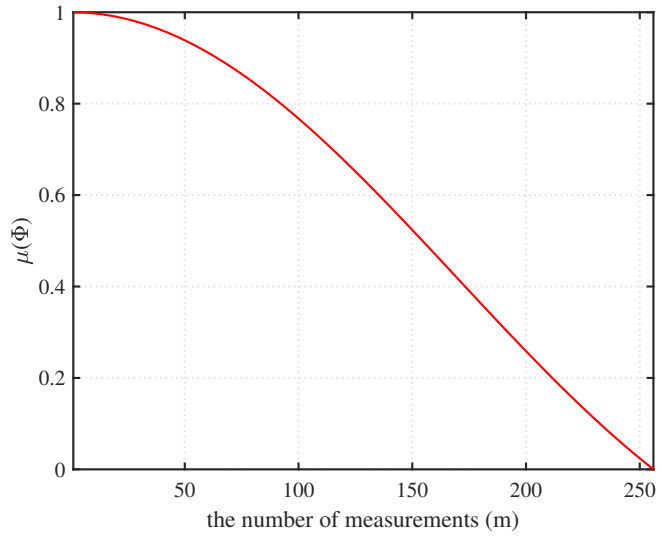


Figure 4.5: Mutual coherence $\mu(\mathbf{\Phi})$ of the IDFT submatrix as a function of the number of measurements m ($N = 256$).

is the IDFT submatrix), we can compute the mutual coherence $\mu(\mathbf{\Phi})$ of $\mathbf{\Phi}$:

$$\begin{aligned}
\mu(\mathbf{\Phi}) &= \max_{p \neq q} \frac{|\langle \boldsymbol{\phi}_p, \boldsymbol{\phi}_q \rangle|}{\|\boldsymbol{\phi}_p\|_2 \|\boldsymbol{\phi}_q\|_2} \\
&= \max_{p \neq q} \frac{1}{m} \left| \sum_{l=1}^{m} e^{\frac{-j2\pi(p-1)(l-1)}{N}} e^{\frac{j2\pi(q-1)(l-1)}{N}} \right| \\
&= \max_{p \neq q} \frac{1}{m} \left| \frac{\sin \frac{\pi m(p-q)}{N}}{\sin \frac{\pi(p-q)}{N}} \right| \\
&= \frac{1}{m} \left| \frac{\sin \frac{\pi m}{N}}{\sin \frac{\pi}{N}} \right|.
\end{aligned}$$

In Fig. 4.5, we plot the mutual coherence $\mu(\mathbf{\Phi})$ as a function of m. One can easily see that $\mu(\mathbf{\Phi})$ increases sharply when m decreases. In our context, when only a few samples are used in the PST decoding, underdetermined ratio $\frac{N}{m}$ of the system matrix increases sharply and hence $\mu(\mathbf{\Phi})$ will also be very large (e.g., $\mu(\mathbf{\Phi}) = 0.9745$ for $m = 32$ and $N = 256$). Clearly, this would cause a severe degradation in the decoding performance.

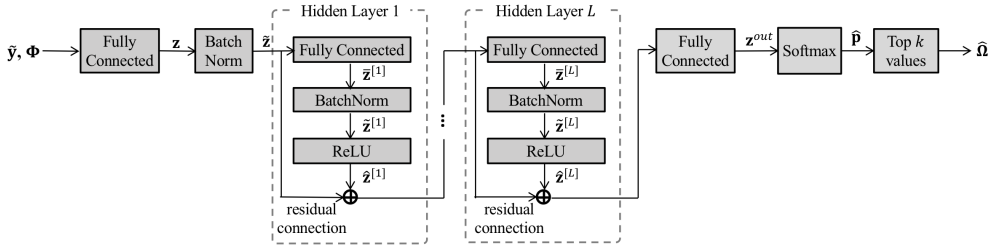


Figure 4.6: Detailed architecture of the D-PST decoding scheme.

To handle this issue, we employ the DL, a data-driven approach to learn a (nonlinear) relationship between the input and the output. Specifically, by using the training data as an input and then updating the network parameters in a way to minimize the loss function, DNN learns the desired operation (support identification). Indeed, since the DNN extracts the correlation structure of $\mathbf{\Phi}$ from the training samples, D-PST can better discriminate the correlated supports in the test phase. Let Q be the nonlinear mapping between the input (the received sample vector $\tilde{\mathbf{y}}$ and the sensing matrix $\mathbf{\Phi}$) and the output (the support of $\mathbf{x}$), then the support identification problem of D-PST is expressed as

$$\hat{\Omega} = Q(\tilde{\mathbf{y}}, \mathbf{\Phi}; \mathbf{\Theta}), \tag{4.9}$$

where $\mathbf{\Theta}$ is the set of weights and biases of D-PST network.

4.3.3 D-PST Decoder Architecture

Fig. 4.6 depicts the structure of the D-PST network. D-PST consists of multiple building blocks including fully-connected (FC) layer, batch normalization layer, rectified linear unit (ReLU) layer, and softmax layer. To construct the input vector $\mathbf{c}$, we vectorize $\mathbf{\Phi}$ and then concatenate it with $\tilde{\mathbf{y}}$ (i.e., $\mathbf{c} = [\tilde{\mathbf{y}}^T \ \boldsymbol{\phi}_1^T \ \cdots \ \boldsymbol{\phi}_N^T]^T$). In each training process, we use D training data $\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(D)}$. In the first FC layer, $\mathbf{c}$ is transformed to the input of hidden layers $\mathbf{z} \in \mathbb{R}^{\alpha \times 1}$:

$$\mathbf{z}^{(d)} = \mathbf{W}^{in}\mathbf{c}^{(d)} + \mathbf{b}^{in}, \qquad \text{for } d = 1, \cdots, D \tag{4.10}$$

where $\mathbf{W}^{in}$ and $\mathbf{b}^{in}$ are the initial weight and the initial bias, respectively. After passing the FC layer, D output vectors are stacked in the batch $\mathbf{B} = [\mathbf{z}^{(1)} \cdots \mathbf{z}^{(D)}]^T$. Then, the normalization process called the batch normalization is performed [42]. In this process, each element $z_i^{(d)}$ $(i = 1, \cdots, \alpha)$ in $\mathbf{B}$ is normalized to have zero mean and unit variance. After that, the normalized element is scaled and shifted by the trained internal parameters. The normalized output vector $\tilde{\mathbf{z}}^{(d)}$ is expressed as

$$\tilde{z}_i^{(d)} = \beta \left(\frac{z_i^{(d)} - \mu_{\mathbf{B},i}}{\sqrt{\sigma_{\mathbf{B},i}^2}} \right) + \gamma, \tag{4.11}$$

where $\mu_{\mathbf{B},i} = \frac{1}{D}\sum_{d=1}^{D} z_i^{(d)}$ and $\sigma_{\mathbf{B},i}^2 = \frac{1}{D}\sum_{d=1}^{D}(z_i^{(d)} - \mu_{\mathbf{B},i})^2$ are the batch-wise mean and variance, respectively, β is the scaling parameter, and γ is the shifting parameter. Note, when the input variation is large, weights of DNN might not be updated properly so that the network might converge slowly or will not converge properly. Since the batch normalization enforces the input to have a constant mean and variance, the irregular and bumpy update of weights caused by the large input variation can be mitigated.

After the batch normalization, the output vector $\tilde{\mathbf{z}}$ passes through L hidden layers for the extraction of the systematic feature from the input-output pair. Based on the universal approximation theorem, by training the DNN consisting of multiple hidden layers, the desired relationship between the input and output can be obtained [43]. In our context, this implies that the whole PST decoding process can be done by the trained DNN with deeply-stacked hidden layers. Each hidden layer consists of the FC layer, batch normalization layer, ReLU layer with a residual connection[2]. The output

[2]Key idea of residual connection is to insert the direct identity connection between the stacked hidden layers. Since the input vector is linked to the output of hidden layer, the feature of input vector can be delivered to all the hidden layers without attenuation and distortion. Thus, we can reduce the training error.

vector of the l-th FC layer $\bar{\mathbf{z}}^{[l]}$ is given by[3]

$$\bar{\mathbf{z}}^{[l]} = \mathbf{W}^{[l]} \left(\tilde{z} + \sum_{i=1}^{l-1} \hat{\mathbf{z}}^{[i]} \right) + \mathbf{b}^{[l]}, \tag{4.12}$$

where $\mathbf{W}^{[l]}$ and $\mathbf{b}^{[l]}$ are the weight and bias of the l-th FC layer, respectively, and $\hat{\mathbf{z}}^{[i]}$ is the output vector of the previous hidden layer. Then, similar to the previous batch normalization, $\bar{\mathbf{z}}^{[l]}$ is normalized to reduce the variation of $\bar{\mathbf{z}}^{[l]}$. After that, a non-linear activation function f is applied to $\tilde{\mathbf{z}}^{[l]}$ to determine whether each hidden node (unit component of hidden layer) $(\tilde{z}_1^{[l]}, \cdots, \tilde{z}_\alpha^{[l]})$ is activated or not. In our network, the ReLU function $f_{\text{ReLU}}(x) = \max(0, x)$ is used as an activation function [44]. By selectively turning on/off the hidden node, the feature-related information (e.g., column correlation structure of $\mathbf{\Phi}$) is delivered to the next hidden layer and undesired information (e.g., values associated the non-support element) is discarded in the current layer. Further, when compared to well-known activation function such as the sigmoid function $(f_{sig}(x) = \frac{1}{1+e^{-x}})$ and the tanh function $(f_{tanh}(x) = \frac{2}{1+e^{-2x}} - 1)$ in which all hidden nodes are activated regardless of x, only a certain number of nodes are activated by the ReLU function so that the computational complexity of backpropagation process can be reduced significantly.

After passing through L hidden layers, the final FC layer is used to match the dimension of output to the dimension of the sparse vector $\mathbf{s}$. That is, the last FC layer produces N-dimensional output vector $\mathbf{z}^{out} \in \mathbb{R}^{N \times 1}$ given by

$$\mathbf{z}^{out} = \mathbf{W}^{out} \left(\tilde{\mathbf{z}} + \sum_{i=1}^{L} \hat{\mathbf{z}}^{[i]} \right) + \mathbf{b}^{out}, \tag{4.13}$$

where $\mathbf{W}^{out}$ and $\mathbf{b}^{out}$ are the weight and bias, respectively. Then, we use the softmax layer to transform $\mathbf{z}^{out}$ to the probabilities used for the support decision. Specifically, the softmax layer generates N probabilities $(\hat{p}_1, \cdots, \hat{p}_N)$ representing the likelihood

[3]For simplicity, we omit the training data index d.

of being the true support element:

$$\hat{p}_i = \frac{e^{z_i^{out}}}{\sum_{j=1}^{N} e^{z_j^{out}}}, \qquad \text{for } i = 1, \cdots, N. \tag{4.14}$$

Since the sparsity k is known to the receiving UE in a priori, an estimate of the support $\hat{\Omega}$ is obtained by choosing k elements having the largest probabilities:

$$\hat{\Omega} = \arg\max_{|\Omega|=k} \sum_{i \in \Omega} \hat{p}_i. \tag{4.15}$$

4.3.4 D-PST Training

In the training phase, we need to find out the network parameter set $\hat{\Theta}$ minimizing the loss function $J(\Theta)$ (i.e., $\hat{\Theta} = \arg\min_{\Theta} J(\Theta)$). In the update of the network parameters, we employ the stochastic gradient descent (SGD) method [30]. In this scheme, parameters in the i-th training iteration Θ_i are updated in the direction of the steepest descent:

$$\Theta_i = \Theta_{i-1} - \frac{\eta}{D} \sum_{d=1}^{D} \nabla_{\Theta} J^{(d)}(\Theta), \tag{4.16}$$

where $J^{(d)}(\Theta)$ is the loss for the d-th training data, $\nabla_{\Theta} J^{(d)}(\Theta)$ is the gradient of $J^{(d)}(\Theta)$ with respect to Θ, and η is the learning rate determining the step size. In (4.16), the gradients for D training examples are averaged out and then used for the update. If Θ is updated with respect to each training example (i.e., $D = 1$), the variation in Θ is large. In this case, the time to obtain the optimal parameters corresponding to the loss minimum is too long, resulting in a significant training overhead. By using the average gradient, the training speed can be increased.

Since the final output of D-PST is the N-dimensional vector $\hat{\mathbf{p}}$ whose element represents the probability of being the support element, $\hat{\mathbf{p}} = [\hat{p}_1, \cdots, \hat{p}_N]^T$ needs to be compared against the true probability $\mathbf{p}$ in the loss function calculation. Since k support elements are assumed to be equiprobable, we set the true probability as $p_i = \frac{1}{k}$ for $i \in \Omega$ and $p_i = 0$ for the rest. In the generation of the loss function, we use the

cross entropy-based loss function $J(\mathbf{p}, \hat{\mathbf{p}})$:

$$J(\mathbf{p}, \hat{\mathbf{p}}) = -\sum_{i=1}^{N} p_i \log \hat{p}_i = -\frac{1}{k} \sum_{j=1}^{k} \log \hat{p}_{\omega_j}, \tag{4.17}$$

where $\omega_j \in \Omega$. In order to minimize this loss function, $\sum_{j=1}^{k} \log \hat{p}_{\omega_j}$ should be maximized. Since the sum of the softmax output values is 1 (i.e., $\sum_i \hat{p}_i = 1$), the maximum can be achieved when $\hat{p}_{\omega_j} = p_{\omega_j} = \frac{1}{k}$.

Similar to the most of DNN, in order to train D-PST, an abundant dataset is required. Using the received signals as a training dataset would be a straightforward option, but doing so will cause a significant overhead in the V2X system. For example, gathering 10 million received signals in the sidelink will take more than 1 hour in LTE systems (assuming that 0.5 ms slot consisting of 7 symbols). In order to reduce the data collection overhead, we use synthetically generated signals for the training dataset. One might concern that the synthetically generated signal is different from the actual received signal since the wireless channel for each sidelink is different. Luckily, we can circumvent this issue since the D-PST decoding is the support identification and channel components are contained in the transmit signal $\mathbf{s} = \mathbf{\Sigma x}$, not the system matrix $\mathbf{\Phi}$ (see (8)). Thus, D-PST does not need to learn the individual V2V sidelink channel, meaning that we can safely use the artificially generated received vector for the training purpose. Clearly, doing so will save the time and effort to collect a huge dataset.

Since the D-PST training is performed offline using the synthetically generated data, we train multiple D-PST networks for various system parameters such as the number of measurements and the number of non-zero elements. From the pre-training process, we can obtain the network parameters (e.g., weight and bias) for each setting. When applying the D-PST to the actual transmission, we thus use the pre-trained network corresponding to the system environment. As long as we have a trained model mating to the system setting, without the re-training process, we can perform the PST decoding.

4.4 Practical PST Implementation For Low-Latency V2X

In this section, we go over two major issues when applying D-PST in the low-latency V2X scenarios. We first discuss the retransmission issue in the V2X sidelink. This issue is crucial for the mission-critical V2X applications since the complicated retransmission procedure causes a significant increase in T_{Rx} so that the latency requirement of URLLC cannot be satisfied. We next discuss the sidelink synchronization issue. In order to decode the OFDM-based packet, the receiving UE should align its timing with the transmit UE's timing reference. To this end, the complicated synchronization signaling between the transmit UE and the receiving UE is needed (e.g., sidelink SSB transmission in 5G NR). Since this process takes substantial time for the receiving UE and thus cannot satisfy the low-latency requirement, we need a new scheme to simplify the synchronization.

4.4.1 Basic Principle of PST Decoding

Before examining the retransmission and the synchronization issues, we discuss a useful property that the support identification performance of PST is not affected by the choice of samples in $\mathbf{y}$. Using this property, we can avoid retransmission and also save extra signaling overhead for synchronization. Our main result is as follows.

Proposition 4 *Consider the system model* $\mathbf{y} = \mathbf{F}\mathbf{x}$ *where* $\mathbf{F} = [\mathbf{f}_1^T \ \mathbf{f}_2^T \ \cdots \ \mathbf{f}_N^T]^T$ *is the IDFT matrix. Let* $\mathbf{\Psi} = [\mathbf{f}_p^T \ \mathbf{f}_{p+1}^T \ \cdots \ \mathbf{f}_{p+m-1}^T]^T$ *be a matrix consisting of* m *consecutive rows in* $\mathbf{F}$*. Suppose* f *is a function (algorithm) returning the support of any vector* $\mathbf{s}$ *for the input* $\mathbf{z}(=\mathbf{\Psi}\mathbf{s})$ *and* $\mathbf{\Psi}$*, that is,* $\Omega_{\mathbf{s}} = f(\mathbf{z}, \mathbf{\Psi}) = f(\mathbf{\Psi}\mathbf{s}, \mathbf{\Psi})$*. Let* $\tilde{\mathbf{y}}_i = [y_i \ y_{i+1} \ \cdots \ y_{i+m-1}]^T$ *be a vector constructed from* i*-th to* $(i+m-1)$*-th samples in* $\mathbf{y}$*. Then, for any* i*,* f *returns the support of* $\mathbf{x}$ *for the input* $\tilde{\mathbf{y}}_i$ *and* $\mathbf{\Psi}$*. That is,* $\Omega_{\mathbf{x}} = f(\tilde{\mathbf{y}}_i, \mathbf{\Psi})$*.*

Proof: First, recall that

$$\tilde{\mathbf{y}}_i = \mathbf{\Phi}_i \mathbf{x} \tag{4.18}$$

where $\mathbf{\Phi}_i = [\mathbf{f}_i^T \ \mathbf{f}_{i+1}^T \ \cdots \ \mathbf{f}_{i+m-1}^T]^T$ is a matrix constructed from i-th to $(i+m-1)$-th rows of $\mathbf{F}$. The ℓ-th column $(\mathbf{\Phi}_i)_\ell$ of $\mathbf{\Phi}_i$ is expressed as

$$\begin{aligned}
(\mathbf{\Phi}_i)_\ell &= \begin{bmatrix} e^{-j(i-1)\frac{2\pi}{N}(\ell-1)} \\ \vdots \\ e^{-j(i+m-2)\frac{2\pi}{N}(\ell-1)} \end{bmatrix} \\
&= e^{-j(i-p)\frac{2\pi}{N}(\ell-1)} \begin{bmatrix} e^{-j(p-1)\frac{2\pi}{N}(\ell-1)} \\ \vdots \\ e^{-j(p+m-2)\frac{2\pi}{N}(\ell-1)} \end{bmatrix} \\
&= e^{-j(i-p)\frac{2\pi}{N}(\ell-1)} (\mathbf{\Psi})_\ell,
\end{aligned} \tag{4.19}$$

and thus we have

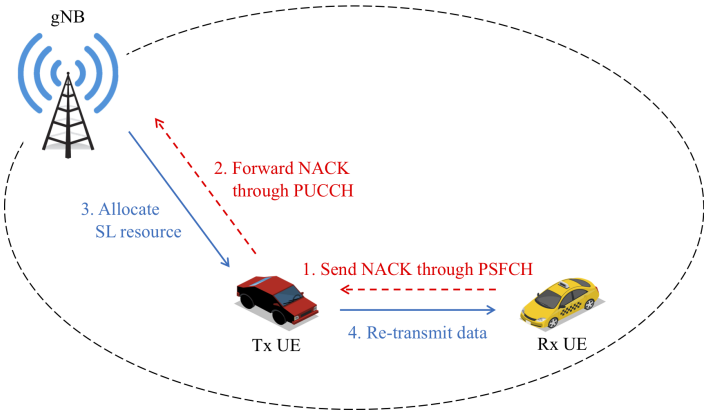
$$\mathbf{\Phi}_i = \mathbf{\Psi}\mathbf{D}_i,$$

where $\mathbf{D}_i = \text{diag}(1, e^{-j(i-p)\frac{2\pi}{N}}, \cdots, e^{-j(i-p)\frac{2\pi}{N}(N-1)})$. From (4.18) and (4.19), we have $\tilde{\mathbf{y}}_i = \mathbf{\Phi}_i \mathbf{x} = \mathbf{\Psi}(\mathbf{D}_i \mathbf{x})$. Thus, it is clear that the function f returns the support $\Omega_{(\mathbf{D}_i\mathbf{x})}$ of $\mathbf{D}_i\mathbf{x}$ for the input $\tilde{\mathbf{y}}_i (= \mathbf{\Psi}(\mathbf{D}_i\mathbf{x}))$ and $\mathbf{\Psi}$. Since $\mathbf{D}_i$ is diagonal, $\Omega_{(\mathbf{D}_i\mathbf{x})} = \Omega_{\mathbf{x}}$, meaning that f returns the same output support $\Omega_{\mathbf{x}}$ for any $\tilde{\mathbf{y}}_i$. □

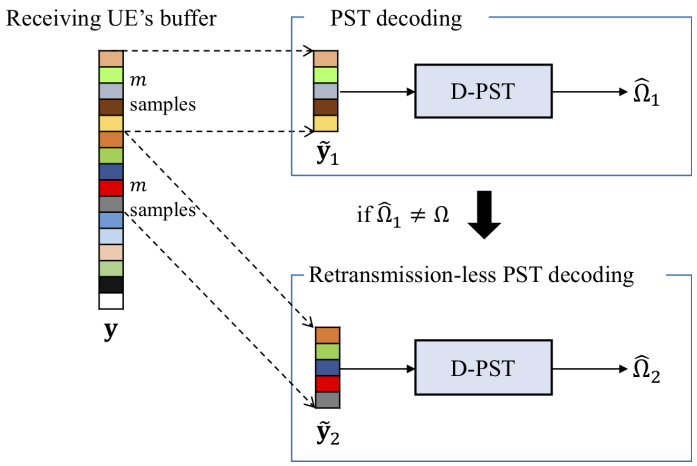
Interestingly, if we use a properly designed sparse recovery (more accurately, support identification) algorithm, then the support identification performance would not be changed by the choice of samples as long as we use m consecutive received samples. In the following subsection, we will explain the benefits of this property.

4.4.2 Retransmission-less PST

In the LTE/NR V2X systems, when the error occurs in the decoding, the receiving UE sends the negative acknowledgment (NACK) message to the transmit UE using



(a)



(b)

Figure 4.7: Description of the sidelink retransmission; (a) An illustration of the sidelink retransmission mechanism. After the decoding failure, the BS allocates the retransmission resources. Then, the transmit UE re-transmits the packets and the receiving UE initiates the decoding process. (b) One simple option of the retransmission-less PST.

the physical sidelink feedback channel (PSFCH) [36]. After receiving NACK message, the transmit UE forwards the message to the BS to request the resources for the retransmission. Then, the BS assigns the transmit resources and the transmit UE re-transmits the packet to the receiving UE (see Fig. 4.7(a)). In this process, clearly, the retransmission latency is large due to the complicated signaling process between the BS and UEs. In fact, it takes around 1 ms to send the NACK message to the BS and initiate the retransmission [36], which is too large to satisfy the URLLC latency requirement (0.1 ms end-to-end latency).

Recalling that D-PST uses only m partial samples in the received signal vector $\mathbf{y}$, remaining $N - m$ samples stored in the receiving UE's buffer are wasted. Thus, when the D-PST decoding fails (i.e., $\hat{\Omega} \neq \Omega$), we can exploit the next m samples in the buffer instead of requesting the re-transmission (see Fig. 4.7(b)). Since the transmit UE does not need to re-transmit the packet, there would be no propagation latency T_{prop} and the buffering latency T_{buff} in (4.1). Noting that D-PST operations in the test phase are just simple multiplication and addition in DNN (see Section III.C), T_{proc} would be very tiny.

4.4.3 Synchronization-free PST

In the decoding of the OFDM-based packet, the symbol timing estimation (a.k.a. synchronization) to find out the first sample of an OFDM symbol is an important process since otherwise the orthogonality among all subcarriers are violated and inter-carrier interference will increase sharply. In the synchronization process, the BS periodically sends a sidelink synchronization signal block (SSB) containing the primary sidelink synchronization signal (PSSS) and the secondary sidelink synchronization signal (SSSS) to cell-in-coverage UEs [36]. By decoding the sidelink SSB, the transmit UE and the receiving UE acquire the cell-specific timing reference, thereby completing the synchronization of the transmit UE and the receiving UE. In the actual sidelink transmission, the receiving UE aligns the received OFDM signal with the synchronized

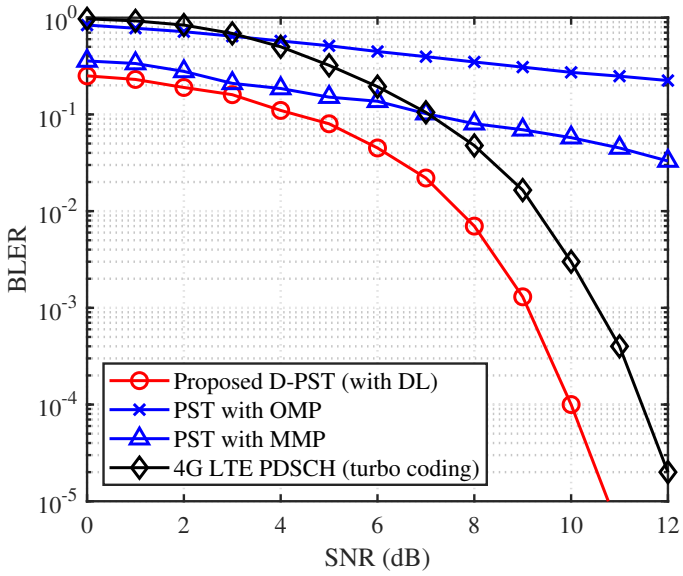
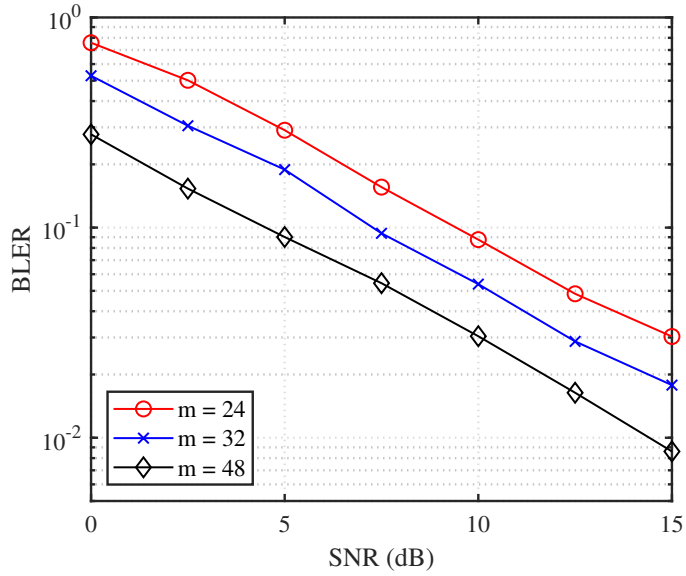


Figure 4.8: BLER performance of the proposed PST scheme as a function of SNR.

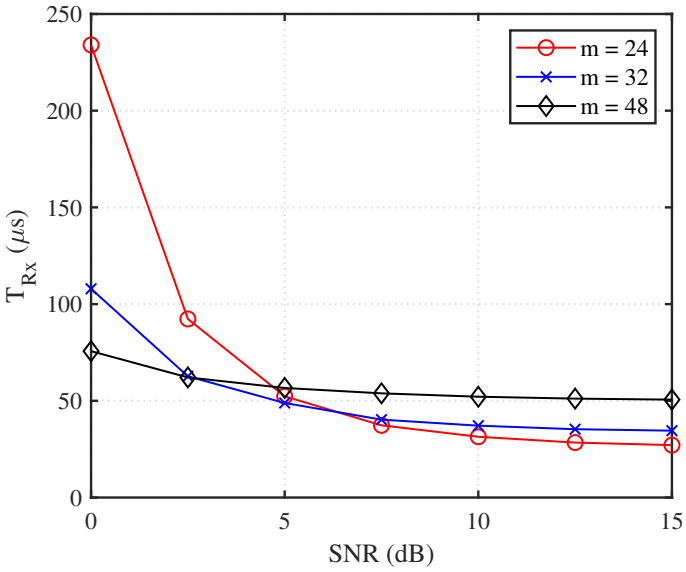
timing and then decodes the transmit information.

When the transmit and receiving UEs are located in different cells, however, the receiving UE cannot decode the transmitted packet using its own timing reference since the timing reference is different for each cell. In this case, the transmit UE needs to deliver the SSB to the receiving UE before the data transmission. Since the sidelink SSB transmission takes 1 ms (assuming 15 kHz subcarrier spacing), the URLLC latency requirement cannot be satisfied by this mechanism.

From the Proposition 4, since the PST decoding performance is not affected by the choice of samples in $\mathbf{y}$, the receiving UE does not need to identify the first sample of the OFDM symbol. This means that, without the sample timing synchronization, the decoding process can be initiated right after buffering any m consecutive samples.



(a)



(b)

Figure 4.9: PST performances for various number of measurements ($m = 24, 32, 48$): (a) BLER performance of PST as a function of SNR. (b) The signaling latency for the PST.

4.5 Numerical Results

In this section, we investigate the decoding performance and latency reduction of the proposed PST. Our simulation setup is based on the OFDM systems. When generating the sparse signal $\mathbf{x}$, we choose $k = 4$ active subcarriers among $N = 96$ subcarriers (i.e., a PST block conveys $\lfloor \log_2 \binom{N}{k} \rfloor + kb_s = 29$ bit information using QPSK symbol). In the D-PST network, we set $L = 6$ (the number of hidden layers), $\alpha = 500$ (the width of hidden layer), and $D = 1000$ (batch size) obtained from the grid search-based hyperparameter tuning (see Fig. 4.11). In order to guarantee the model stability of D-PST, we use K-fold cross validation in the training phase. In the K-fold cross validation, total training examples are randomly partitioned into K equal-sized sets. Among K sets, a single set is used for the model validation and the remaining $K - 1$ sets are used for the D-PST training. In our simulations, we use 10^7 examples and set $K = 10$. When training the D-PST, we use an Adam optimizer, well-known optimization tool to keep the robustness of learning process [45]. As performance measures, the block error rate (BLER) and the reception latency T_{Rx} in (3.1) are considered.

In Fig. 4.8, we evaluate the BLER performance of the proposed PST scheme as a function of SNR. For comparison, we examine the performance of the conventional physical downlink shared channel (PDSCH) transmission, OMP-based PST, and MMP-based PST[4]. We observe that the D-PST scheme outperforms conventional schemes by a large margin. For example, the D-PST scheme achieves 1.4 dB gain over the 4G LTE PDSCH transmission at BLER=10^{-5}.

In Fig. 4.9(a), we evaluate the BLER performance of the PST scheme for various number of measurements ($m = 24, 32, 48$). When m increases, we see that the BLER is decreased sharply. For example, if m is doubled from 24 to 48, we can achieve a large gain (more than 5 dB) in BLER performance. We also investigate T_{Rx} required to complete the PST transmission for different values of m (see Fig. 4.9(b)). We observe

[4]MMP has been proposed as a near-ML sparse recovery algorithm. In a nutshell, MMP performs an efficient tree search to find out the near-ML solution in a sparse signal recovery.

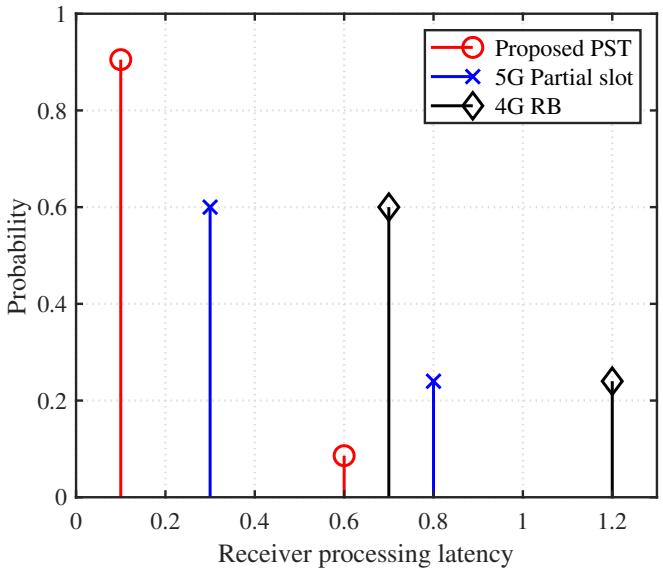


Figure 4.10: Probability of receiver processing latency to complete the packet transmission (SNR=5 dB).

that for high SNR regime, T_{Rx} can be reduced significantly when m decreases. For example, when m is reduced from 48 to 24, the latency is reduced by the factor of 46% at SNR=15 dB. However, when m is too small, T_{Rx} increases little bit, in particular for low SNR regime, since in this case the PST decoding can be failed and hence the PST transmission needs to be repeated.

Next, we investigate the latency performance of the PST. For comparison, we use the partial slot transmission (5G NR) and RB-based transmission (4G LTE). In Fig. 4.10, we plot the distribution of T_{Rx} which corresponds to the time from the initial transmission to the successful packet decoding at the receiving UE. Note, if the decoding is failed, the packets are re-transmitted in next slot and the receiver operations need to be repeated. From the results, we observe that T_{Rx} of PST is much smaller (52% and 82% on average) than that of partial transmission and RB transmission, respectively.

In Fig. 4.11, we evaluate the D-PST performance for various number of hidden layers ($L = 3, 6, 9$) to see the effect of the number of hidden layers on the support

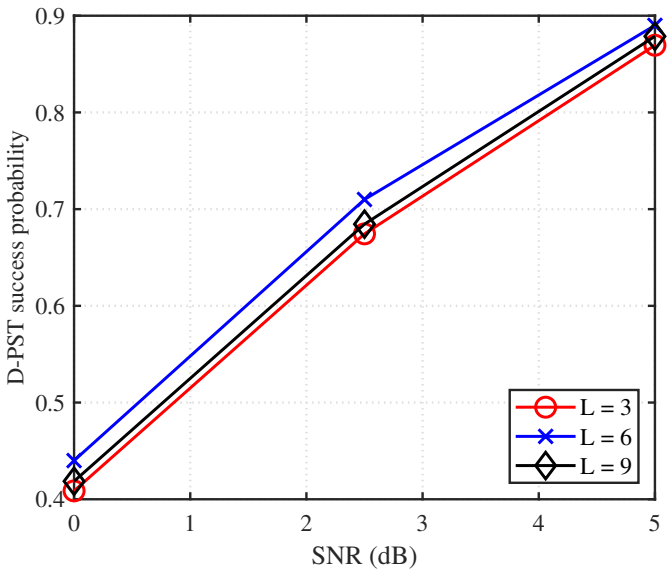


Figure 4.11: Support identification performance of the D-PST scheme for various number of hidden layers ($L = 3, 6, 9$).

identification performance. For example, when we increase the depth of the hidden layer, then D-PST success probability improves until $L = 6$ but after that the success probability will not be improved further.

4.6 Summary

In this chapter, we proposed a novel low-latency transmission scheme suitable for the URLLC-based V2X scenarios. The key idea behind the proposed PST scheme is to encode the mission-critical sidelink information in a form of the sparse symbol vector and then decode the information with a partially-buffered samples using deep learning-based decoder. When the number of subcarriers is small enough and the measurements contain enough information to figure out the transmit information, accurate decoding of the PST-encoded packet can be guaranteed. In the decoding process, we exploit the DNN architecture to learn the nonlinear mapping between the received signal vector

and the support of input sparse signal. As long as we train the deeply stacked network using a properly designed loss function along with the backpropagation weight update, we can identify the accurate support in the test phase. We demonstrated from the numerical evaluations that the proposed D-PST is very effective in terms of both the reliability and latency. In this paper, we restricted our attention to the V2X sidelink transmission but we believe that there are many interesting applications of the proposed approaches in mission-critical machine-type communications.

Chapter 5

Deep Learning-based Wireless Communication Systems: Design Perspective

In this chapter, we will present an overview of the DL-based wireless systems to serve as a starting point to facilitate the use of DL in the wireless system design. The successful design of DL-based system comes down to the choice of a proper DL model for the target wireless application, detailed neural network architecture design, and training data acquisition along with training strategy selection. With this purpose in mind, we discuss the key principles of DL-based design and then provide several useful design tips learned from our experience with plentiful wireless communication examples including channel estimation, MIMO beamforming, power management, and angle-of-arrival (AoA) detection. After the introduction, we organize the rest of this article as follows. In Section 5.2, we briefly review the design principles of conventional and discuss DL-based wireless systems and learning techniques used for the wireless system design. In Section 5.3, we explain the training dataset collection and neural network architecture design issues. We discuss future issues and conclude the paper in Section 5.4.

5.1 Introduction

Artificial intelligence (AI) is a powerful tool to perform tasks that seem to be simple for human being but are extremely difficult for conventional (rule-based) computer program. Deep learning (DL), a branch of AI techniques introduced by Lecun, Bengio, and Hinton [46], has shown great promise in many practical applications. In the past few years, we have witnessed great success of DL in various fields such as traditional Go game, image classification, speech recognition, language translation, among others [47, 48, 49]. Recently, DL techniques have also been applied to various wireless communication applications such as multiple-input-multiple-output (MIMO) detection, channel estimation, spectrum sensing, and resource scheduling.

When one tries to use AI technique to the wireless applications, one can be easily overwhelmed by so many knobs to control and small details to be aware of. In contrast to the conventional communication systems where the performance analysis and the algorithm design are done analytically, DL requires lots of hands-on experience and heuristic knowledge in the design of neural network, training dataset generation, and also choice of the training strategy. In fact, since the DL design process is data-driven and inductive in nature, one can easily get lost or stuck in the middle when they try to solve the wireless communication problem using the DL technique.

5.2 Artificial Intelligence-Based Wireless Communications

In this section, we briefly discuss two design principles: conventional wireless systems and the AI-based systems. We then discuss how specific communication function is mapped to the DL techniques.

5.2.1 Design Principles of Conventional and AI-based Wireless Systems

When designing wireless systems, whole system is divided into several functional blocks such as channel encoder, symbol mapper, channel estimator, MIMO detec-

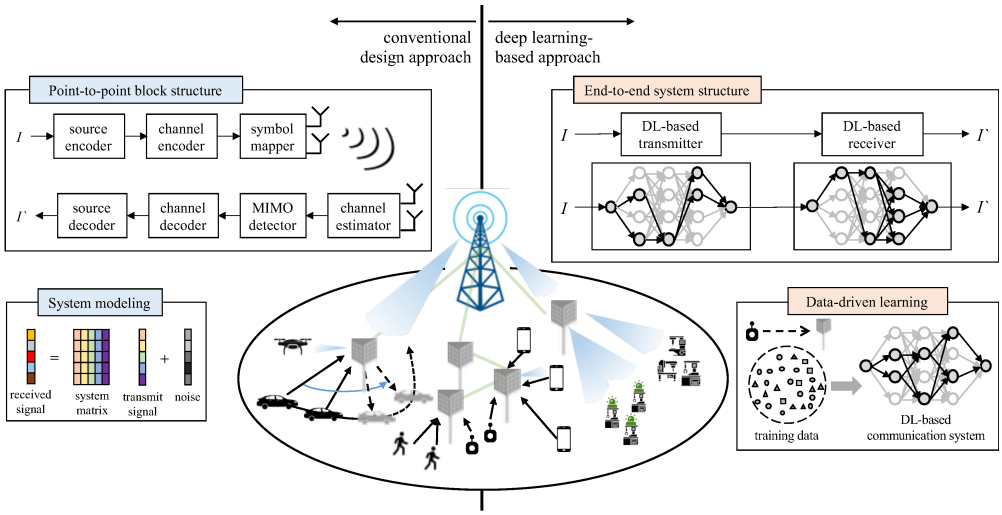


Figure 5.1: Design principles of traditional communication system and AI-based communication system.

tor, and channel decoder (see Fig. 5.1). In each functional block, system modeling, performance analysis, and algorithm design are performed. System model, typically expressed as a clean-cut linear equation, defines the relationship between observation (e.g., received signal) and latent variables to be recovered (e.g., transmit signal). Using this, theoretical analysis is conducted to obtain the performance limit such as the capacity bound or achievable degree of freedom (DOF) and then a proper algorithm achieving near optimal performance is developed (e.g., MMSE channel estimator, maximum-likelihood (ML)-based symbol detector). For example, in the mmWave channel estimator design, a propagation channel is modeled by the geometric parameters such as angle-of-departure/arrival (AoD/AoA), path delay, and path gain and then the compressed sensing (CS) technique is employed to find out sparse parameters used to reconstruct the mmWave channel.

When the wireless environments and systems are becoming more complicated, it is very difficult to come up with a simple yet tractable system model. Further, due to the excessive assumptions in fading/noise/interference distribution, input statistics,

and traffic/mobility pattern, obtained analytic result will leave a considerable gap from the real-world performance.

As an entirely-new paradigm to deal with the problem, AI has been popularly used in various applications such as computer vision, speech recognition, robot control, and autonomous driving [48, 49]. A holy grail of AI is to let machine learn the complicated, often highly nonlinear, relationship between the input dataset and the desired output without human intervention. As a technique to implement AI, DL, an approach to use deeply stacked neural network in the training, has been widely used in recent years. In a nutshell, DL-based systems are distinct from the conventional systems in two main respects: *data-driven training* and *end-to-end learning* of the black box (see Fig. 5.1). Instead of following the analytical avenue, the DL model approximates the desired function as a whole using the training dataset. In the training phase, DL parameters (weights and biases) are updated to identify the end-to-end mapping between the input dataset and the desired output. Once the training is finished, DL returns the predicted output for the input in the inference phase. This means that what we essentially need to do is to just feed a training dataset into the properly designed DL model. It seems to be simple but requires lots of hands-on experience to get the most bang for the buck.

5.2.2 Learning Techniques for DL-based Wireless Communication

When one tries to use DL to the wireless systems, perhaps the first thing to consider is to determine what learning technique to use. Depending on the design goal, training dataset, and learning mechanism, DL techniques can be roughly divided into three categories: supervised learning, unsupervised learning, and reinforcement learning.

1) **Supervised learning**: primary goal of the supervised learning is to learn the mapping function between the input dataset and the desired solution called *label*. To scrutinize the quality of a designed neural network and reflect it in the weight update process, we need a loss function that measures how far the predicted output is from the label. The difference between the predicted output and the label, in a form of cross

entropy or mean squared error (MSE), is used as a loss function. Typically, there are two types of the supervised learning: *classification* to find out the categorical class of given input (e.g., device is active or not) and *regression* to return the numerical value (e.g., estimated channel). The classification task is suitable for the detection problem such as the MIMO detection, automatic modulation classification (AMC), and active user detection (AUD) and the regression task is a good fit for the estimation problem such as the channel estimation, angle (DoA/AoD) estimation, and log likelihood ratio (LLR) generation [50, 51, 52, 53].

In AUD, for example, we identify a few active (data-transmitting) users among all possible users in a cell so that it can be well interpreted as a multi-label classification problem identifying a few, say k, labels among N classes. By employing the set of received vectors $\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$ as inputs and the active user index Ω as an output, deep neural network (DNN) is trained to find out indices of active users. In the channel estimation problem, on the other hand, desired task is to produce the real-valued channel estimate $\hat{\mathbf{H}}$ from the received pilot signals $\mathbf{y}$. Using the MSE between the real channel $\mathbf{H}$ and the DL output $\hat{\mathbf{H}}$ (i.e., estimate of $\mathbf{H}$) as a loss function, DNN learns the regression mapping between $\mathbf{y}$ and $\mathbf{H}$.

2) **Unsupervised learning**: unsupervised learning is used when the ground-truth label is unavailable. In this case, clearly, one cannot compute the difference between the generated output and the label so that the design goal (i.e., objective function) is used as a loss function instead. In the resource allocation problem, for example, it is very difficult to find out an optimal resource scheduling maximizing the quality-of-service (QoS) since the problem is highly nonlinear mixed-integer programming [54]. In this case, by employing the QoS function itself as a loss function, the DL model can be trained. Another example fitting to this category is the MIMO beamforming problem. Essence of this problem is to find out the downlink beamforming vector $\mathbf{w}$ maximizing the users' sum rate [55]. Since the sum rate maximization problem is non-convex, it is in general very difficult to find out the optimal vector $\mathbf{w}^*$ so that

Table 5.1: Summary of DL Techniques

Learning technique	Applicable problem	Loss function	Application example
Supervised learning	Detection problem using the classification training	Cross entropy, Kullback-Leibler (KL) divergence,	MIMO detection, Active user detection
	Estimation problem using the regression training	Mean squared error (MSE), Mean absolute error (MAE)	Channel estimation, DoA estimation
Unsupervised learning	Optimization problem	Objective function to be optimized (e.g., sum-rate, cell throughput)	MIMO beamforming, Resource scheduling
Reinforcement learning	Sequential decision making problem	Cumulative reward (e.g., total power consumption)	Power management, Spectrum sensing

the supervised learning might not be an appropriate option. When we try to solve the problem using the unsupervised learning, we set the downlink channel $\mathbf{h}_{k,n}$ between the k-th BS and the n-th user as a training dataset, the beamforming weight $\mathbf{w}$ as an output, and the negative sum rate $-R = -\sum_{n} \log_2 \left(1 + \frac{\|\mathbf{h}_{k,n}^H \mathbf{w}_k\|^2}{\sum_{i \neq k} \sum_{j \neq n} \|\mathbf{h}_{i,j}^H \mathbf{w}_i\|^2 + N_0}\right)$ as a loss function. In short, the unsupervised learning is useful when the desired output used as a label is unavailable for the reasons such as nonlinearity/nonconvexity of the problem.

3) **Reinforcement learning** (RL): RL is a goal-oriented learning technique where an agent learns how to solve a task by trials and errors. In the learning process, the agent observes the state of an environment, takes an action, and then receives a reward for the action. RL is suitable for the sequential decision-making problem whose purpose is to find out a series of actions maximizing the performance metric. Recently, deep RL (DRL) has been popularly used since it can effectively handle the large-scale state-action pair in dynamically varying wireless environments. For example, when we try to improve the energy efficiency of the ultra-dense network (UDN), we can use DRL to control the on/off mode or user association pattern of small-cell base stations (SBSs) [56]. In the DRL implementation, a digital unit (DU), playing the role of the agent, observes the state (e.g., channel state and user rate constraint) and then determines the action (on/off mode selection of SBSs) based on the reward. To minimize

the energy consumption, the reward should be set to be high/low when the consumed energy is small/large. By playing sufficient number of episodes (sequence of states, actions, and rewards), DRL-based DNN learns the SBS control policy minimizing the long-term energy consumption of the wireless network.

5.3 Issues To Be Considered For DL-based Wireless Communication Systems

Two key ingredients in the DL-based wireless systems are sufficient and comprehensive training samples and properly designed neural network. In this section, we delve into these issues.

5.3.1 Training Dataset Acquisition

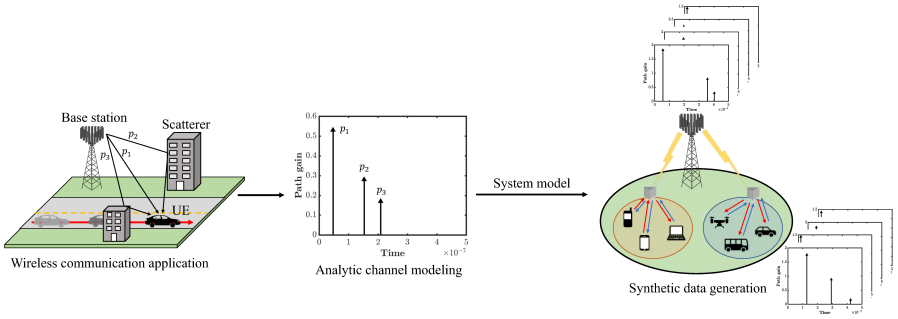
When the number of samples is not sufficient enough, the designed DL model would be closely fitted to the training dataset, making it difficult to make a reasonable inference for the unseen data. This problem that the trained DL model lacks the generalizing capability is often called *overfitting*. In the modulation classification task, for example, if the received signals are generated from the BPSK and QPSK modulation exclusively, then the trained classifier cannot accurately identify the 16-QAM modulated symbols. To prevent the overfitting problem, the dataset should be large enough to cover all possible scenarios. This is not easy, in particular for wireless systems, since the number of real transmissions will be humongous. In acquiring the training dataset, we basically have three options:

- Collection from the actual received signals
- Synthetic data generation using the analytic system model
- Real-like training set generation using generative adversarial network (GAN)

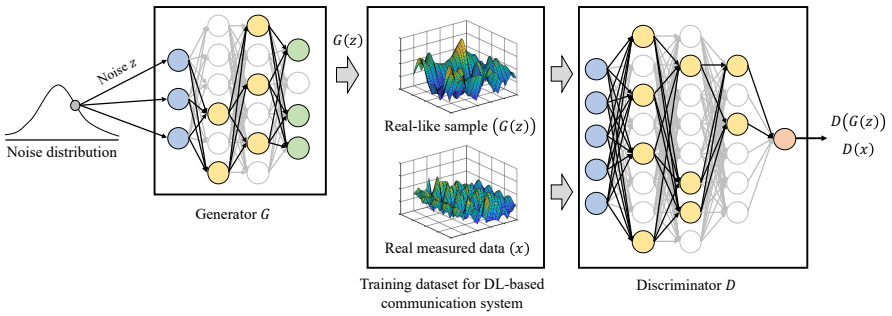
In the data acquisition, a straightforward option is to collect the real transmit/receive signal pair. Doing so, however, will cause a significant overhead since it requires too many training data transmissions. For example, when collecting one million received signals in 5G NR systems, it will take more than 15 minutes (10^6 symbols $\times$ 0.1 subframe/symbol $\times$ 8 ms/subframe).

To reduce the overhead, one can consider synthetically generated dataset (see Fig. 5.2(a)). In fact, in the design, test, and performance evaluation phase of most wireless systems, analytic models have been widely used. For example, propagation channels such as the extended pedestrian A (EPA) channel or extended vehicular A (EVA) channel has been popularly employed in the generation of training dataset [57]. Since the synthetic data can be generated easily using a computer, time and effort to collect huge training data can be saved. However, there might be some, arguably non-trivial, performance degradation caused by the model mismatch.

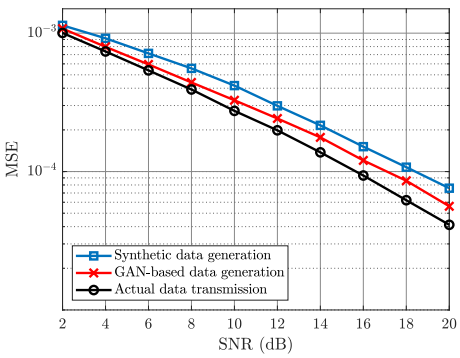
Yet another intriguing option is to use the artificial but realistic samples generated by the DL technique. This approach is in particular useful when the analytic model is unknown or non-existent (e.g., underwater acoustic communication and satellite communication) and real measured data is not enough. In this case, GAN technique, an approach to generate samples having the same distribution with the input dataset, can be employed [58]. GAN consists of two neural networks: generator G and discriminator D (see Fig. 5.2(b)). G produces real-like data $G(z)$ from the random noise z and D tries to distinguish whether the generated output $G(z)$ is real or fake. To train these two networks, the min-max loss function, typically expressed as the cross-entropy distance between the distribution of $G(z)$ and that of the real data x, is often used (i.e., $\min_G \max_D V(G,D) = \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$ where $D(x)$ is the discriminator's probability for x being real (non-fake)). When the training is finished properly, the generator output is fairly reliable and hence the discriminator cannot judge whether the generator output is real or fake, which means that we can readily use the generator output as a training data.



(a)



(b)



(c)

Figure 5.2: Illustration of data acquisition strategies: (a) synthetic data generation; (b) GAN-based data generation; (c) MSE performances of the DL-based channel estimator using three distinct strategies.

In order to observe the validity of the training data generation strategies we discussed, we evaluate the MSE performance of the DL-based channel estimator (see Fig. 5.2(c)). As shown in Fig. 5.2(c), we observe that the GAN-based data generation and the synthetic data generation are effective and fairly competitive. Interestingly, the performance gap between actual data transmission and GAN-based data generation is insignificant since the distribution of the GAN-generated data matches well with that of the real data. Whereas, the gap between synthetic data generation and actual data transmission is a bit large (around 2 dB at MSE$= 10^{-4}$) due to the model mismatch between real channels and synthetic channels generated from the analytic model.

5.3.2 DNN Architecture Design

In the design of DNN-based wireless systems, one should consider the input characteristics (e.g., temporal/spatial/geometric correlation), wireless environments (e.g., mmWave/THz/V2X/ UAV link), and system configurations (e.g., bandwidth, power, number of antennas).

1) **Baseline network**: a natural first step of the DNN design is to choose the baseline architecture. Based on the connection shape between neighboring layers, the neural network can be divided into three types: fully-connected network (FCN), convolutional neural network (CNN), and recurrent neural network (RNN).

FCN can be used universally since each hidden unit (neuron) is connected to all neurons in the next layer. When the input dataset has a spatial structure (e.g., 2D-resource time/frequency grid and the 2D antenna array in MIMO systems), CNN might be an appealing option. In CNN, each neuron is computed by the convolution between the 2D spatial filter and a part (e.g., rectangular shaped region) of neurons in the previous layer. Due to the local connectivity within the convolution filter, CNN facilitates the extraction of spatial correlated feature. For example, in the mmWave MIMO beamforming, 2D beam radiation patterns among the uniform rectangular array (URA) antennas can be extracted using CNN.

Whereas, when the input sequence is temporally correlated, which is true for most of communication channels, RNN or long-short term memory (LSTM) might be a good choice. By employing the current inputs together with outputs of the previous hidden layer, temporally correlated feature can be extracted. For instance, by applying RNN to the mmWave beam tracking, change of the Doppler frequency caused by the mobile's movement can be extracted.

2) **Activation layer**: activation layer is used to 1) embed the nonlinearity in the hidden layer and 2) generate the desired type of output in the final layer.

In each hidden layer, weighted sum of inputs passes through the activation layer to determine whether the information generated by the hidden unit is activated (delivered to the next layer) or not. To this end, rectified linear unit (ReLU) function $f(x) = \max(x, 0)$ or hyperbolic tangent function $f(x) = \tanh(x)$ can be used. By imposing the nonlinearity to the linearly transformed input, one can better promote the nonlinear operation (e.g., successive interference cancellation) and systematic nonlinearity (e.g., amplifier distortion or nonlinear RF filtering).

In the final layer of DNN, the activation layer is used to ensure that the generated output is the desired type. To compute the loss function, we should make sure that the DL output and the true label should be the same type. In the classification problem, the ground-truth label for each class is the probability so that the final output should be a form of the probability. When there are several active users in AUD or the occupied/empty bands are non-unique in the spectrum sensing problem, it would be desirable to use the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ returning the individual probability for each class. Whereas, when the problem is modeled as a multi-class classification problem such as the MIMO detection problem, a softmax function $f_i(\mathbf{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ would be a good fit since it normalizes the output vector into the probability distribution over all classes.

3) **Input normalization**: in the training process, the neural network computes the gradient of a loss function with respect to any weight and then updates the weight in

the negative direction of the gradient. Therefore, when the input varies in a wide range (e.g., multi-user communication scenario), variation in the weight update process will also be large, degrading the training stability and convergence speed severely. To prevent this ill-behavior, one should perform the normalization for the outputs of each layer. Typically, there are two types of the normalization strategies: layer normalization and batch normalization.

When the input vector contains signals from multiple users with different wireless geometries, variation of the received signals would be quite large. The layer normalization is a good fit for this case. By normalizing each input vector (i.e., replacing $\mathbf{a}$ with $\frac{\mathbf{a}}{\|\mathbf{a}\|_2}$), the layer normalization scheme ensures that the normalized input distribution has the fixed mean and variance.

Whereas, when the input data consists of several different types of information, the batch normalization (BN) can be a better option. In the mini-batch $\mathbf{B} = [\mathbf{y}^{(1)} \cdots \mathbf{y}^{(N)}]^T$ consisting of multiple input samples $\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(N)}$, elements in each row of $\mathbf{B}$ (i.e., elements with the same input type) are normalized. For example, in the DRL-based power management problem discussed in Section II.B, both the channel state information (CSI) and required data rate are used as inputs to DRL. Since the scale of two components would be quite different, the layer normalization will not work and simply mess up the input dataset. To avoid the problem, CSI and data rate need to be normalized separately using BN.

4) **Dropout layer**: when we use DNN consisting of multiple hidden layers, the final output is determined by the activated hidden units in each layer. So, for the highly correlated inputs (e.g., samples generated from the non-orthogonal sparse codebook and bit streams with small Hamming distance), their activation patterns will also be similar so that the final inference can be easily corrupted in the presence of perturbations (e.g., noise, inter-user interference, and channel estimation error). In order to mitigate this problem, the dropout layer where the activated hidden units are dropped out randomly can be employed in the training phase [59]. In this scheme, by tem-

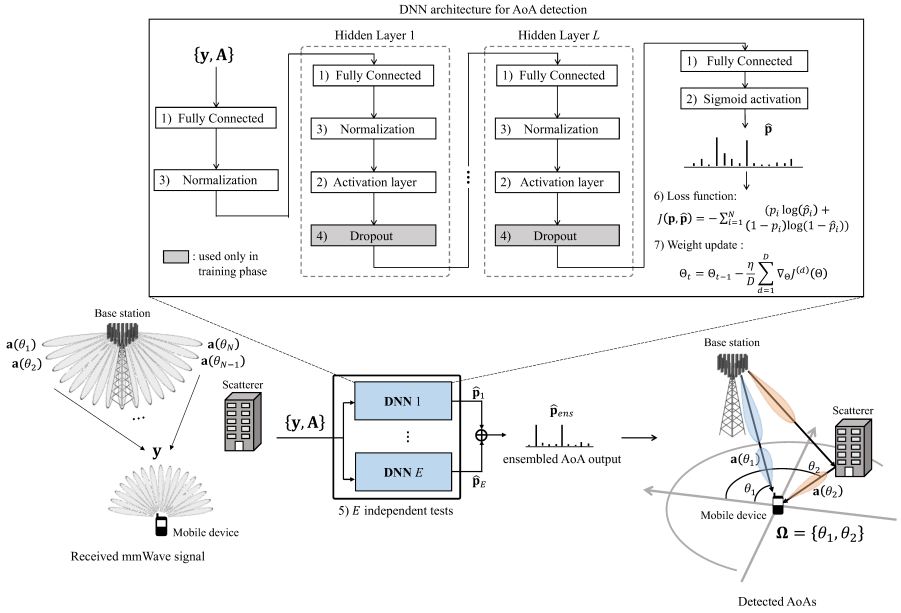
porarily removing part of incoming and outgoing connections randomly, ambiguity (similarity) of the activation patterns among correlated dataset can be better resolved.

5) **Ensemble learning**: ensemble learning, a method to average out multiple outputs (inferences) of independently-trained networks, is conceptually analogous to the receiver diversity technique in that it enhances the output quality without requiring additional wireless resources (e.g., frequency, time, and transmitter power). In the multi-user communication scenario, the trained network might be closely fitted to the certain wireless environment so that the trained DNN cannot make a reliable prediction for the inputs generated from unobserved wireless scenario in the training. In this case, ensemble learning comes to the rescue. Key idea of the ensemble learning technique is to train the multiple neural networks with different training sets and initial parameters obtained from different wireless conditions and then combine the generated outputs to improve the quality of the final inference. Using the ensemble learning-based DNN, one can mitigate the overfitting caused by the wireless environments considerably.

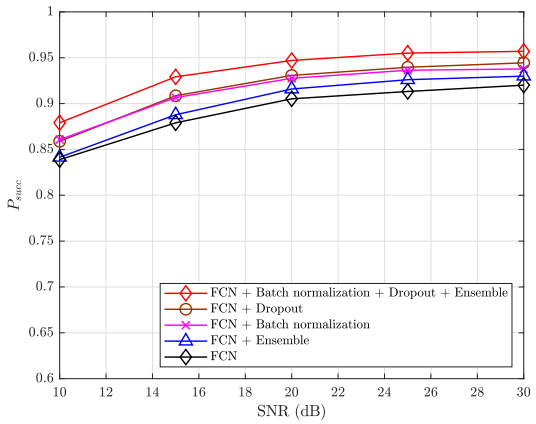
6) **Loss function**: since DNN weights are updated in a direction to minimize the loss function, the loss function should well reflect the design goal. When the ground-truth label is available, one can use the cross entropy, MSE, or mean absolute error (MAE). If this is not the case, as we discussed in unsupervised learning, one might use the design goal (e.g., throughput or energy consumption) as a loss function.

If there exist multiple constraints for the problem at hand, these constraints should be combined together in the loss function. For example, in the DL-based power management in UDN, the DNN is trained to minimize the total consumed power and at the same time should satisfy the rate requirement of a mobile. To do so, we set the loss function as the weighted sum of power consumption loss J_{pow} and the rate constraint loss J_{rate} (i.e., $J(\Theta) = J_{pow} + \lambda J_{rate}$). By controlling the regularization weight λ, one can achieve the trade-off between the consumed power and data rate.

7) **Weight update strategy**: in order to update the network parameter set Θ, the gradient of the loss function $J(\Theta)$ should be computed first. A straightforward way to



(a)



(b)

Figure 5.3: (a) Exemplary DNN architecture designed for the AoA detection. (b) AoA detection performance of various DNNs as a function of SNR. P_{succ} denotes the success probability which corresponds to the percentage of detected AoAs among all angles.

update the parameters is the batch gradient descent (BGD) method where the gradient of $J(\Theta)$ is computed for the entire training dataset. Since the whole dataset is used in each and every training iteration, the training cost is quite expensive and the training speed will be very slow. Further, in the non-static scenario where the channel characteristics are varying, parameters corresponding to the dynamically changing wireless environments (e.g., Doppler spread, scatter location) would not be updated properly. A better option, in fact widely used option, is the stochastic gradient descent (SGD) method. In contrast to BGD, SGD uses a small number, say D, of samples in each training iteration (i.e., $\Theta_t = \Theta_{t-1} - \frac{\eta}{D}\sum_{d=1}^{D} \nabla_{\Theta} J^{(d)}(\Theta)$) so that it can update the network parameters as soon as D samples are obtained.

8) **Knowledge distillation**: when we train the DL model in the Internet of Things (IoT) device, on-device energy consumption is a big concern since most of the IoT devices are battery-powered. To reduce the training overhead, knowledge distillation (KD) technique [60], an approach to generate a relatively small-sized DL model from a trained large model, can be employed. Key idea of KD is to train a small network (a.k.a student network) using the output of a large network (a.k.a teacher network). In the generation of the loss function, output of the student network is compared against the output of the teacher network as well as the ground-truth label. In doing so, the student network, implemented in IoT device, can easily capture the underlying feature (e.g., similarity and difference among the classes) extracted by the teacher network implemented in the digital unit (DU). For the properly designed student network using KD, we see that the performance of student network is fairly comparable to that of teacher network.

In Fig. 5.3(a), we present the DNN architecture for the AoA detection using the techniques we discussed. Due to the sparse scattering in the mmWave band, a propagation path can be characterized by a few AoAs. By identifying these angles, the receiver can align the beam direction to the transmitter, thereby maximizing the signal-to-noise ratio (SNR). In DNN, we use the received signal $\mathbf{y}$ and the steering matrix

$\mathbf{A} = [\mathbf{a}(\theta_1) \ \cdots \ \mathbf{a}(\theta_N)]$ ($\mathbf{a}(\theta_i) = [1 \ e^{j\pi \sin\theta_i} \ \cdots \ e^{j\pi(m-1)\sin\theta_i}]^T$ is the steering vector corresponding to θ_i) as inputs and the set of the detected AoAs Ω as outputs. Since the input is a composite of $\mathbf{y}$ and $\mathbf{A}$, we use BN to normalize each component separately. Also, to generate the individual probability for each angle, we use the sigmoid activation function in the final layer.

To judge the effectiveness of the DNN architecture consisting of the normalization, dropout, and ensemble learning, we evaluate the detection success probability P_{succ} of the AoA detection. In our simulations, we train 1) FCN, 2) FCN with BN, 3) FCN with the dropout layer, 4) FCN with the ensemble network, and 5) FCN with all techniques we discussed. As shown in Fig. 5.3(b), the performance gain introduced by the detailed DNN techniques is considerable. For example, FCN with the dropout layer achieves a significant gain (4.8 dB gain at $P_{succ} = 0.9$) over the conventional FCN since the correlation between the closely located steering vectors can be better resolved using the dropout technique. The gain obtained from BN is also significant (4.7 dB gain at $P_{succ} = 0.9$) since the variation of $\mathbf{y}$ caused by the device location change can be alleviated. Finally, when the gains induced by all techniques are combined together, we can achieve very accurate performance ($P_{succ} \approx 0.97$), which can never be obtained by the basic FCN even in high SNR regime.

5.4 Summary

In this chapter, we presented an overview of DL-based wireless system with emphasis on the design issues related to DL model selection, training set acquisition, and DNN architecture design. As the automated services and applications using machines, vehicles, and sensors proliferate, we expect that DL will be more popular and eventually become a dominating design paradigm in 6G era. To deal with various frequency bands (i.e., sub-6GHz/mmWave/THz), wireless resources (massive MIMO antennas, intelligent reflecting surface, relays), and geographical environment, we need to go

beyond the state-of-the-art DL technique used mainly for the purpose of the function approximator and exploit more aggressive and advanced DL techniques. For example, when we try to train a DL model for the desired task, transfer learning, an approach to use the pre-trained model for a similar task, can be employed. By recycling most of parameters in the pre-trained model and training only a small part of parameters, new model can learn the distinct information for the desired task while utilizing the common feature between two tasks. Another approach worth investigation is the meta learning, a technique to learn the desired task quickly using the DL models of similar tasks. By setting the parameters minimizing the sum loss functions of similar tasks as initial parameters, DL model for the desired task can be learned with reduced training overhead.

Our hope is that this article will serve as a useful guide for communication researchers who want to apply the DL technique in their wireless application.

Chapter 6

Deep Neural Network Based Active User Detection for Grant-free NOMA Systems

This chapter proposed the DL-based active user detection (AUD) for the grant-free NOMA scenario. For an efficient and accurate AUD, we exploit the deep neural network (DNN), a learning-based tool to approximate the complicated and nonlinear function. Over the years, DNN has been successfully applied in numerous applications such as image classification [48], machine translation [63], automatic speech recognition [49], and Go game [47]. Recently, DNN has been also applied to various wireless systems such as multiple-input and multiple-output (MIMO) detection, wireless scheduling, direction-of-arrival (DoA) estimation, and multi-user detection [64, 65, 66]. In these works, DNN is used to learn a desired nonlinear function (e.g., classification and decision) through the training process. In [64], for instance, the DNN structure to learn the mapping between the interference pattern and the optimized scheduling has been proposed. In [65], a DNN architecture for the symbol generation, encoding, and decoding in grant-free NOMA systems has been proposed. In [66], the long short-term memory (LSTM) network performing the channel estimation and data detection in grant-based NOMA systems has been presented.

The work of Chapter 6 has been published in part in [61, 62].

In our framework, DNN learns the complicated mapping between the received NOMA signal and the indices of active users in the transmit signal. To be specific, the proposed AUD scheme, henceforth referred to as deep AUD (D-AUD), learns the sparse structure of device activity using a deliberately designed training dataset. It is now well-known from the *universal approximation theorem* that DNN processed by the deeply stacked hidden layers can well approximate the desired function [67]. In our context, this means that the trained DNN with multiple hidden layers can handle the whole AUD process, resulting in an accurate detection of the active users.

6.1 Introduction

In recent years, massive machine-type communication (mMTC) has received much attention due to the variety of applications such as smart factory and building, public safety and monitoring, smart metering, to name just a few. As the term speaks for itself, mMTC concerns the access of massive machine-type communication (MTC) devices (e.g., sensors, robots, drones, machines) to the base station (BS) [1]. Main goal of mMTC is to support the massive connectivity in the uplink-dominated communication. However, this task is too demanding in the conventional wireless systems (e.g., Long Term Evolution-MTC (LTE-M) and narrow-band Internet-of-Things (NB-IoT) [68]) for the heavy signaling overhead caused by the complicated handshaking in the scheduling process and the lack of time/frequency resources caused by the orthogonal resource allocation to a large number of MTC devices [69, 70]. In the NB-IoT, for example, only scheduled transmission is performed in a narrowband spectrum so that it cannot properly handle the massive access of machine-type devices.

As a solution to support the massive connectivity, *grant-free access* and *non-orthogonal multiple access* have been proposed in recent years [71], [72]. Grant-free access allows the transmission of MTC device to the BS without the granting process. Since each device transmits information without scheduling, a process to identify ac-

tive devices (i.e., devices transmitting information) among all potential devices in a cell is required. This process, often referred to as the *active user detection* (AUD), is an important problem in the grant-free mMTC since without this process the BS cannot figure out the active devices transmitting information. In order to support the massive connectivity with limited amount of resources, an approach to use non-orthogonal sequences, called non-orthogonal multiple access (NOMA), has been proposed [72]. In this scheme, by the superposition of multiple devices' signals, orthogonality of transmit signals is intentionally violated. To control the interuser interference caused by the orthogonality violation, NOMA employs device specific non-orthogonal sequences and deliberately designed nonlinear detector (e.g., successive interference cancellation (SIC) and message passing algorithm (MPA) [73]).

By exploiting the fact that only a few active devices in a cell transmit the information concurrently (see Fig. 6.1), the AUD problem can be readily formulated as a sparse recovery problem [2, 74, 75, 76]. In [74], the AUD problem is modeled as a single measurement vector (SMV) problem and MPA is used to solve the problem. In this CS-based AUD scheme, BS detects active devices based on the correlation between the received signal and device specific sequence. However, performance of the CS-based AUD is not that appealing when the columns of a system matrix (a.k.a. sensing matrix) are highly correlated and sparsity (the number of nonzero elements) of the underlying input vector increases[1]. In fact, in the practical NOMA-based transmission, correlation among the NOMA sequences is relatively high so that the CS-based AUD might not be effective, in particular when the device activity (sparsity) is high [2]. Therefore, it is of importance to come up with a new type of AUD scheme suitable for the overloaded yet less sparse access scenarios.

[1]When the number of users increases, column dimension of sensing matrix will also increase. Thus, for the fixed number of measurements, the underdetermined ratio will also increase. In this case, the column correlation will be very large. Also, based on the principle of compressed sensing, the required number of measurements increases as the sparsity increases. Hence, the decoding performance will be degraded sharply when the sparsity increases (under the condition that the measurement size is fixed) [2].

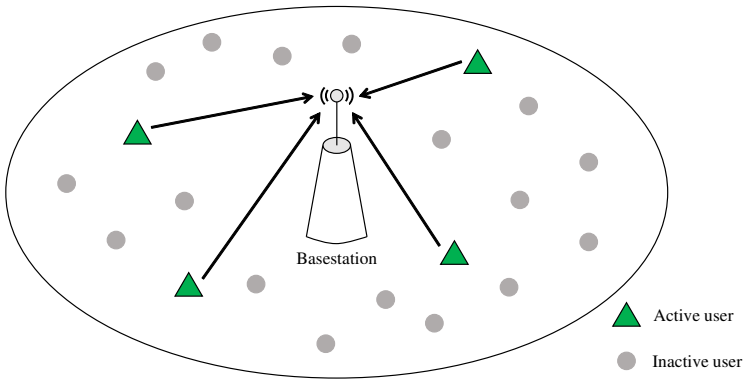


Figure 6.1: System model of the mMTC uplink scenario where only a few MTC devices are active.

6.2 AUD System Model

We consider the uplink grant-free NOMA systems in which the BS equipped with a single antenna receives information from multiple machine-type devices with a single antenna[2]. When transmitting the data, the mobile device randomly chooses a frequency subband from a set of available subbands. In our work, we consider the overloaded scenario where the number of devices N in a cell is larger than the number of frequency resources m ($m < N$). Since each device can transmit packets freely without scheduling, the BS should identify *active* devices transmitting packets. Active devices transmit the information after the spreading with the device specific (non-orthogonal) sequences[3] (see Fig. 6.2). Specifically, the bitstream is mapped to the symbol s_i and then converted to the spreading vector $\mathbf{q}_i = \mathbf{c}_i s_i$ using the device specific codeword $\mathbf{c}_i$.

[2]Extension of the system model to the multi-antenna model is straightforward (see Section V. B).

[3]In this work, we use the preconfigured sequence selection where the BS assigns a sequence to the mobile device via the random access procedure. In doing so, the collisions caused by the duplicated sequences can be prevented. Typically, since the scheduling request and the grant signaling are not performed in the grant-free transmission, we only consider the initial random access in the proposed scheme (user specific sequence is assigned in this process).

In this work, we employ the low-density signature (LDS) sequence where the codeword of a device contains lots of zeros [77]. Due to the sparse nature of a codeword, each symbol is spread into only a small number of resources, resulting in the reduction of the interuser interference. For example, the LDS codebook $\mathbf{C}_{(4,6)}$ to support 6 devices with 4 resources is

$$\mathbf{C}_{(4,6)} = \begin{bmatrix} 0 & w_0 & w_1 & 0 & w_2 & 0 \\ w_0 & 0 & w_2 & 0 & 0 & w_1 \\ 0 & w_1 & 0 & w_2 & 0 & w_0 \\ w_2 & 0 & 0 & w_1 & w_0 & 0 \end{bmatrix}, \tag{6.1}$$

where w_j is the non-zero element of the codeword [77].

Let s_i be the transmit symbol for the i-th device, then the observation vector $\mathbf{y}$ at the BS is given by

$$\mathbf{y} = \sum_{i=1}^{N} \mathrm{diag}(\mathbf{c}_i)\mathbf{h}_i s_i + \mathbf{v} \tag{6.2}$$

$$= \mathbf{C}\mathbf{q} + \mathbf{v}, \tag{6.3}$$

where $\mathbf{c}_i = [c_{i,1} \ \cdots \ c_{i,m}]^T$ is the LDS codeword vector for the i-th device, $\mathbf{h}_i = [h_{i,1} \ \cdots \ h_{i,m}]^T$ is the channel vector between the i-th device and the BS, $\mathbf{v} \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ is the complex Gaussian noise vector, $\mathbf{C} = [\mathrm{diag}(\mathbf{c}_1) \ \cdots \ \mathrm{diag}(\mathbf{c}_N)]$ is the codebook matrix of all devices in a cell, and $\mathbf{q} = \left[\mathbf{q}_1^T \ \cdots \ \mathbf{q}_N^T\right]^T = \left[(s_1\mathbf{h}_1)^T \ \cdots \ (s_N\mathbf{h}_N)^T\right]^T$ is the composite of symbol and channel vectors. It is worth pointing out that $\mathbf{q}_i$ contains the (frequency-domain) channel vector $\mathbf{h}_i$. Note also that only a few devices are active at a given time, and thus the vector $\mathbf{q}$ can be readily modeled as a sparse vector.

In performing the AUD, we use multiple, say N_d, data measurements. Let $\tilde{\mathbf{y}} =$

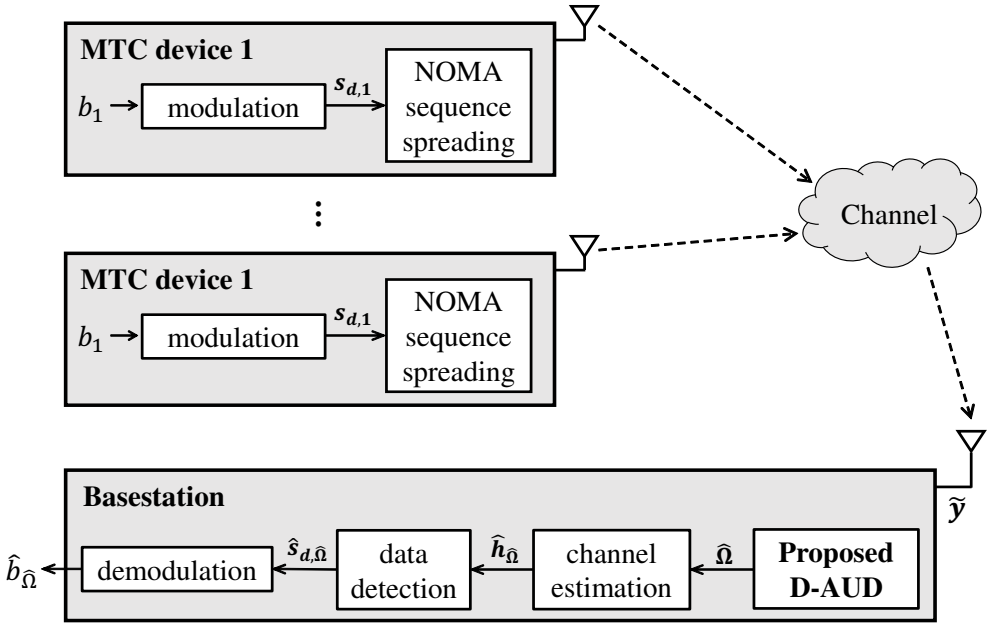


Figure 6.2: Block diagram of the proposed D-AUD scheme.

$\left[\left(\mathbf{y}^{(1)}\right)^T \ \cdots \ \left(\mathbf{y}^{(N_d)}\right)^T \right]^T$ be the stacked vector of the N_d measurements, then

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{C}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}^{(N_d)} \end{bmatrix} \begin{bmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \vdots \\ \mathbf{q}^{(N_d)} \end{bmatrix} + \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \\ \vdots \\ \mathbf{v}^{(N_d)} \end{bmatrix}, \tag{6.4}$$

where $\mathbf{C}^{(t)} = \left[\mathrm{diag}(\mathbf{c}_1^{(t)}) \ \cdots \ \mathrm{diag}(\mathbf{c}_N^{(t)})\right]$ and $\mathbf{q}^{(t)} = \left[(s_1^{(t)}\mathbf{h}_1^{(t)})^T \ \cdots \ (s_N^{(t)}\mathbf{h}_N^{(t)})^T\right]^T$ is the vector whose element is the composite of the channel vector and data symbol. Since the indices of active devices are the same for all $\mathbf{q}^{(t)}$, the supports[4] of $\mathbf{q}^{(t)}$ ($t = 1, \ldots, N_d$) will also be the same (i.e., $supp(\mathbf{q}^{(1)}) = supp(\mathbf{q}^{(2)}) = \cdots$). In order to identify the active device, therefore, it would be better to re-arrange the system model based on the index of devices. To this end, we use a device activity indicator δ_i where $\delta_i = 1$ for the active device and $\delta_i = 0$ for the rest (inactive device). Using the device activity indicator, the received vector $\tilde{\mathbf{y}}$ can be expressed as

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{\Phi}_1 & \cdots & \mathbf{\Phi}_N \end{bmatrix} \begin{bmatrix} \delta_1 \mathbf{x}_1 \\ \vdots \\ \delta_N \mathbf{x}_N \end{bmatrix} + \begin{bmatrix} \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{v}^{(N_d)} \end{bmatrix} = \mathbf{\Phi}\mathbf{x} + \mathbf{v}, \tag{6.5}$$

where $\mathbf{x}_i = \left[(s_i^{(1)}\mathbf{h}_i^{(1)})^T \ \cdots \ (s_i^{(N_d)}\mathbf{h}_i^{(N_d)})^T\right]^T$ and $\mathbf{\Phi}_i = \left[\mathrm{diag}(\mathbf{c}_i^{(1)}) \ \cdots \ \mathrm{diag}(\mathbf{c}_i^{(N_d)})\right]$ are the re-arranged sparse vector and codebook matrix for the i-th device, respectively, $\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_1 & \cdots & \mathbf{\Phi}_N \end{bmatrix}$, and $\mathbf{x} = [\delta_1 \mathbf{x}_1^T \ \cdots \ \delta_N \mathbf{x}_N^T]^T$.

Since a small number of devices (say k devices) are active, the stacked sparse vector $\mathbf{x}$ has k nonzero blocks, which implies that the received vector $\tilde{\mathbf{y}} = \mathbf{\Phi}\mathbf{x} + \mathbf{v}$ can be expressed as a linear combination of k submatrices of $\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_N$ perturbed by the noise. Note that $\mathbf{\Phi}$ is available at the BS since all entries of the codebook matrix $\mathbf{C}$ are known in advance. In light of this, main task of the BS is to identify the submatrices $\mathbf{\Phi}_i$ in $\mathbf{\Phi}$ participating in $\tilde{\mathbf{y}}$. For example, if the second and fifth devices are active

[4] If $\mathbf{s} = [0\ 1\ 0\ 0\ 1\ 0]$, then the support of $\mathbf{s}$ is $supp(\mathbf{s}) = \{2, 5\}$.

(i.e., $\Omega = \{2, 5\}$), then $\mathbf{\Phi}_2$ and $\mathbf{\Phi}_5$ participate in $\tilde{\mathbf{y}}$. Note that this setup is standard in the compressed sensing [2] and the AUD problem can be formulated as the support identification problem:

$$\tilde{\Omega} = \arg \min_{|\Omega|=k} \frac{1}{2} \|\tilde{\mathbf{y}} - \mathbf{\Phi}_\Omega \mathbf{x}_\Omega\|_2^2 . \tag{6.6}$$

In solving (6.6), greedy block sparse recovery algorithm such as block orthogonal matching pursuit (BOMP) [2] and block compressive sampling matching pursuit (B-CoSaMP) [78] can be used. In each iteration, greedy block sparse recovery algorithm identifies one submatrix of $\mathbf{\Phi}$ at a time using a greedy strategy. In j-th iteration, for example, a submatrix $\mathbf{\Phi}_l$ of $\mathbf{\Phi}$ that is maximally correlated with the residual vector $\mathbf{r}^{j-1}$ is chosen. An index of the nonzero submatrix of $\mathbf{\Phi}$ chosen at j-th iteration is

$$\omega_j = \arg \max_{l=1,\cdots,N} \left\| \mathbf{\Phi}_l^H \mathbf{r}^{j-1} \right\|_2^2 , \tag{6.7}$$

where $\mathbf{r}^{j-1} = \mathbf{y} - \mathbf{\Phi}_{\Omega^{j-1}} \hat{\mathbf{x}}^{j-1}$ is the j-th residual vector and $\hat{\mathbf{x}}^{j-1} = \mathbf{\Phi}_{\Omega^{j-1}}^{\dagger} \mathbf{y}$ is the estimate of $\mathbf{x}$ at $(j-1)$-th iteration. One can easily see that the support identification performance depends heavily on the correlation between the residual $\mathbf{r}^{(\cdot)}$ and sensing matrix $\mathbf{\Phi}$ generated from the codebook matrix $\mathbf{C}$.

After identifying the support Ω, a BS detects the symbol vector $\hat{\mathbf{s}}_{\hat{\Omega}}$ of the active device. To be specific, by removing the components associated with the non-support elements in (6.5), the system model can be converted from the underdetermined system to the overdetermined system $(m > k)$. For example, if the identified support is $\Omega = \{2, 5\}$, then the system model in (6.5) can be simplified to

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{\Phi}_2 & \mathbf{\Phi}_5 \end{bmatrix} \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{x}_5 \end{bmatrix} + \mathbf{v}$$

and thus a conventional technique such as the linear minimum mean square error (LMMSE) estimator followed by the symbol slicer can be used for the symbol detection (see Fig. 6.2).

In real scenarios, this type of CS-based AUD schemes might not be effective for the following reasons. First, correlation of codewords increases with the number of

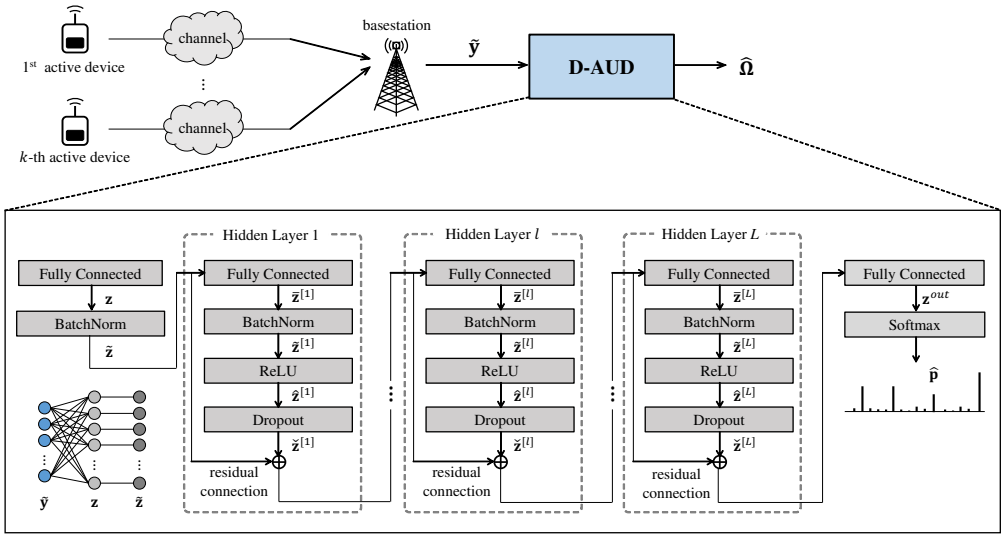


Figure 6.3: Detailed architecture of the proposed D-AUD.

devices. Indeed, when we try to support a large number of devices using small amount of resources, column dimension of the codebook $\mathbf{C}$ would be much larger than the size of measurement vector $\tilde{\mathbf{y}}$, increasing the underdetermined ratio $\frac{N}{m}$ of the system. In this case, clearly, the mutual coherence[5] of $\mathbf{C}$ will increase sharply, causing a severe degradation of the AUD performance. Second, when the activity of devices is high (i.e., k is large), required number of iterations of the greedy sparse recovery algorithm will also increase. Recalling that the residual vector is updated using the estimated support in each iteration (see (6.7)), an error caused by the incorrectly chosen support element will be propagated (this phenomenon is called *error propagation*), deteriorating the AUD performance severely. Last but not least, computational complexity and latency of the iterative algorithm are burdensome in the real-time AUD since the complexity and processing time of sparse recovery algorithm depend heavily on the number of

[5]The mutual coherence $\mu(\mathbf{\Phi})$ is defined as the largest magnitude of normalized inner product between two distinct columns of $\mathbf{\Phi}$ [2]:

$$\mu(\mathbf{\Phi}) = \max_{i \neq j} \frac{|\langle \mathbf{\Phi}_i, \mathbf{\Phi}_j \rangle|}{\|\mathbf{\Phi}_i\|_2 \|\mathbf{\Phi}_j\|_2}.$$

active devices[6]. Due to the reasons mentioned, when the number of active devices is large, the CS-based AUD scheme would not be an appealing solution. Without doubt, design of new type of AUD scheme robust to the codeword correlation and high device activity is of great importance for the success of grant-free NOMA systems in 5G and beyond[7].

6.3 Deep Neural Network Based AUD

As mentioned, main goal of AUD is to identify the nonzero positions of $\mathbf{x}$, not the recovery of nonzero elements. In this work, we use DNN, a feedforward neural network having multiple hidden units between input and output [80], to solve the problem. By using the training data as an input and then updating the parameters using backpropagation process, DNN learns the nonlinear mapping g between the input (i.e., received signal vector $\tilde{\mathbf{y}}$) and the support of $\mathbf{x}$. The resulting support identification problem of the proposed D-AUD can be expressed as

$$\hat{\Omega} = g(\tilde{\mathbf{y}}; \Theta), \tag{6.8}$$

where $\tilde{\mathbf{y}}$ is the input vector and Θ is the set of weights and biases of D-AUD network.

6.3.1 D-AUD Architecture

The primary task of the D-AUD is to find out g parameterized by Θ given $\tilde{\mathbf{y}}$, closest to the optimal mapping function g^*. Fig. 6.3 depicts the structure of the proposed D-AUD technique. D-AUD consists of multiple building blocks including fully-connected (FC) layers, rectified linear unit (ReLU) layer, dropout layer, and softmax layer with the batch normalization. In the training process, we use P training data $\tilde{\mathbf{y}}^{(1)}, \cdots, \tilde{\mathbf{y}}^{(P)}$ in

[6]For example, the computational complexity of BOMP is $O\left(m^2kN\right)$. Therefore, increase in the number of active devices will directly affect the computational complexity.

[7]Various NOMA proposals (e.g., power-domain NOMA, LDS-OFDM, and SCMA) have been proposed in 3GPP Rel. 15 [79] and standardization effort is still underway.

each training iteration. Since $\tilde{\mathbf{y}}^{(p)}$ is a complex vector, we split the real and imaginary parts and use $\hat{\mathbf{y}}^{(p)} = [\Re(\hat{y}_1^{(p)}) \ \cdots \ \Re(\hat{y}_m^{(p)}) \ \Im(\hat{y}_1^{(p)}) \ \cdots \ \Im(\hat{y}_m^{(p)})]$ as an input vector. The output vector $\mathbf{z}^{(p)} \in \mathbb{R}^{\alpha \times 1}$ of the FC layer can be expressed as[8]

$$\mathbf{z}^{(p)} = \mathbf{W}^{in}\hat{\mathbf{y}}^{(p)} + \mathbf{b}^{in}, \qquad \text{for } p = 1, \cdots, P, \tag{6.9}$$

where $\mathbf{W}^{in} \in \mathbb{R}^{\alpha \times 2m}$ is the initial weight and $\mathbf{b}^{in} \in \mathbb{R}^{\alpha \times 1}$ is the initial bias. After the FC layer, P output vectors are stacked in the mini-batch $\mathbf{B} = \left[\mathbf{z}^{(1)} \ \cdots \ \mathbf{z}^{(P)}\right]^T$ and then normalized. This process is referred to as the batch normalization [81]. In this step, each element $z_i^{(p)}$ $(i = 1, \cdots, \alpha)$ in $\mathbf{B}$ is normalized to have zero mean and unit variance. Then, the normalized element is scaled and shifted by internal parameters. The output $\tilde{\mathbf{z}}^{(p)}$ of the batch normalization is expressed as

$$\tilde{z}_i^{(p)} = \beta \left(\frac{z_i^{(p)} - \mu_{\mathbf{B},i}}{\sqrt{\sigma_{\mathbf{B},i}^2}} \right) + \gamma, \qquad \text{for } i = 1, \cdots, \alpha, \tag{6.10}$$

where $\mu_{\mathbf{B},i} = \frac{1}{P}\sum_{p=1}^{P} z_i^{(p)}$ and $\sigma_{\mathbf{B},i}^2 = \frac{1}{P}\sum_{p=1}^{P}(z_i^{(p)} - \mu_{\mathbf{B},i})^2$ are the batch-wise mean and variance, respectively, β is the scaling parameter, and γ is the shifting parameter. One can see that this normalization process enforces the input distribution to have the fixed means and variances. When the variation of input data is large, it is difficult to extract internal features (e.g., block sparse structure and codebook structure) from the input data. Indeed, since mobile devices in different wireless geometries transmit the data in grant-free NOMA scenario, variation in $\tilde{\mathbf{y}}$ is typically very large. Using the batch normalization, D-AUD can control the variation of inputs caused by the different channel state and noise level.

After the batch normalization, the output vector $\tilde{\mathbf{z}}$ passes through the multiple hidden layers[9]. Each hidden layer consists of the FC layer, batch normalization layer,

[8] α is a hyper-parameter representing the width of hidden layers. In general, when α is large, the training performance becomes high due to the large learning capacity. We will more discuss α in Section V.

[9] In the sequel, we omit the training data index p for notational simplicity.

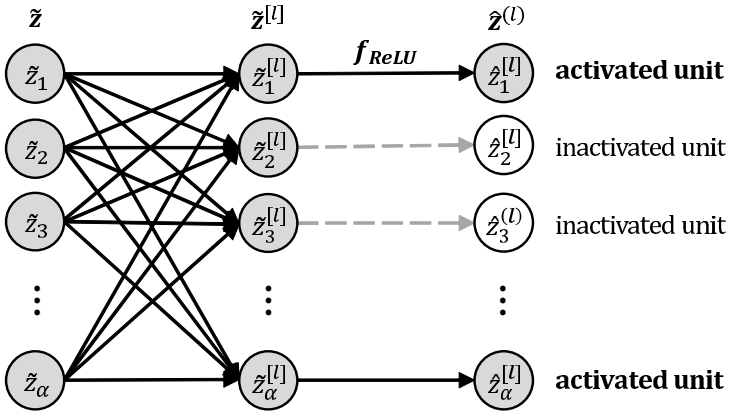


Figure 6.4: Description of the ReLU layer.

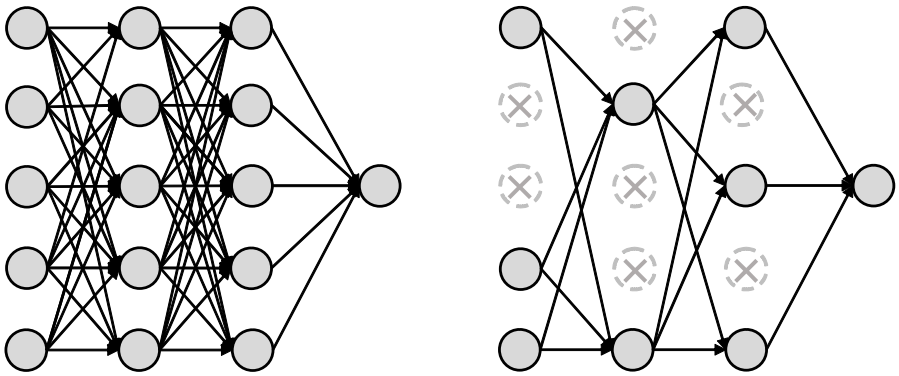
ReLU layer, dropout layer with a residual connection[10] (see Fig. 6.3). The output of the l-th FC layer $\bar{\mathbf{z}}^{[l]}$ is given by

$$\bar{\mathbf{z}}^{[l]} = \mathbf{W}^{[l]} \left(\tilde{\mathbf{z}} + \sum_{i=1}^{l-1} \check{\mathbf{z}}^{[i]} \right) + \mathbf{b}^{[l]}, \tag{6.11}$$

where $\mathbf{W}^{[l]} \in \mathbb{R}^{\alpha \times \alpha}$ and $\mathbf{b}^{[l]} \in \mathbb{R}^{\alpha \times 1}$ are the weight and bias in the l-th FC layer, respectively and $\check{\mathbf{z}}^{[i]} = f\left(\beta^{[i]} \left(\mathbf{W}^{[i]} \left(\tilde{\mathbf{z}} + \sum_{j=1}^{i-1} \check{\mathbf{z}}^{[j]} \right) + \mathbf{b}^{[i]} - \boldsymbol{\mu}^{[i]} \right) \oslash \boldsymbol{\sigma}^{[i]} + \gamma^{[i]} \right) \odot \mathbf{d}^{[i]}$ is the output of the i-th dropout layer (we will say more about this in the next page)[11]. Then, the batch normalization is performed to reduce the variation of $\bar{\mathbf{z}}^{[l]}$. After that, a nonlinear activation function is applied to $\tilde{\mathbf{z}}^{[l]}$ to determine whether the information $(\tilde{z}_1^{[l]}, \cdots, \tilde{z}_\alpha^{[l]})$ generated by the hidden unit is activated (delivered to the next layer) or not (see Fig. 6.4) [83]. To this end, an activation function such as the

[10] The key feature of residual connection is to put the direct identity (shortcut) connection between the stacked hidden layers. To be specific, denoting the input $\mathbf{x}$ and the desired underlying mapping as $H(\mathbf{x})$, the multiple hidden layers are fit to the residual mapping $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$, not $H(\mathbf{x})$ directly [82]. Since the input vector is directly linked to the output of hidden layer, the information (feature) can be delivered across the hidden layers without distortion and attenuation. Hence, we can achieve a reduction in the training error.

[11] $\boldsymbol{\mu}^{[i]} = [\frac{1}{P}\sum_{p=1}^{P} \bar{z}_1^{(p)}, \cdots, \frac{1}{P}\sum_{p=1}^{P} \bar{z}_\alpha^{(p)}]^T$ and $\boldsymbol{\sigma}^{[i]} = \left[\sqrt{\frac{1}{P}\sum_{p=1}^{P}(\bar{z}_1^{(p)} - \mu_1^{[i]})^2}, \cdots, \sqrt{\frac{1}{P}\sum_{p=1}^{P}(\bar{z}_\alpha^{(p)} - \mu_\alpha^{[i]})^2}\right]^T$.



(a) (b)

Figure 6.5: Dropout neural network model: (a) A standard neural network consists of three hidden layers. All hidden units in hidden layers are activated. (b) After applying the dropout, the activated hidden units are dropped out randomly.

sigmoid function ($f(x) = \frac{1}{1+e^{-x}}$) or ReLU function ($f(x) = \max(x, 0)$) can be used:

$$\hat{\mathbf{z}}^{[l]} = f(\tilde{\mathbf{z}}^{[l]}). \tag{6.12}$$

Since the proposed scheme learns the mapping between $\tilde{\mathbf{y}}$ and the support Ω, an estimate of the support $\hat{\Omega}$ would be strongly affected by the activation patterns, presumably on/off patterns, of hidden units. When the sensing matrix $\mathbf{\Phi}$ is less correlated (i.e., the mutual coherence of the sensing matrix $\mathbf{\Phi}$ is low), $\tilde{\mathbf{y}}$ can be expressed as a linear combination of *less correlated* columns of $\mathbf{\Phi}_\Omega$, and thus the identification of Ω from $\tilde{\mathbf{y}}$ would be relatively easy and straightforward. Whereas, when the sensing matrix $\mathbf{\Phi}$ is highly correlated, mapping between $\tilde{\mathbf{y}}$ and Ω might not be clear and can be easily confused in the presence of randomly distributed perturbations (e.g., channel estimation error, inter-user interference, and noise). Suppose two columns of $\mathbf{\Phi}$ are strongly correlated and only one of these is associated with the support, then it might not be easy to distinguish a correct support element from an incorrect one. For example, if $\Omega_1 = \{1, 8\}$ and $\Omega_2 = \{2, 6\}$ and $|\langle \mathbf{\Phi}_1, \mathbf{\Phi}_2 \rangle| \approx 1$ and $|\langle \mathbf{\Phi}_8, \mathbf{\Phi}_6 \rangle| \approx 1$, then the activation patterns of hidden units for Ω_1 and Ω_2 would be quite similar, ending up causing incorrect support identification even in the presence of a small perturbation.

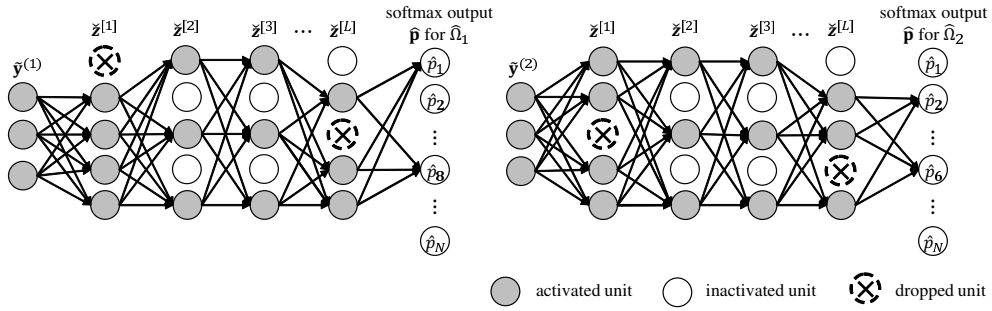


Figure 6.6: Examples of activation patterns corresponding to the strongly correlated supports Ω_1 and Ω_2. Using the dropout layer, the randomly chosen hidden units are dropped out and the activation patterns for Ω_1 and Ω_2 can be better resolved.

In order to reduce this type of mistake, we use the *dropout* layer where the activated hidden units are dropped out randomly (see Fig. 6.5) [59]. By removing incoming and outgoing connections of the dropped units, similarity (ambiguity) of the activation patterns among correlated supports can be better resolved, which implies that D-AUD can identify the support accurately (see the illustration in Fig. 6.6).

Let $\mathbf{d}^{[l]}$ be the dropout vector, then the i-th element $d_i^{[l]}$ of $\mathbf{d}^{[l]}$ and the final output of the l-th hidden layer are

$$d_i^{[l]} \sim Bern(\mathrm{P}_{drop}) \tag{6.13}$$

$$\check{\mathbf{z}}^{[l]} = \mathbf{d}^{[l]} \odot \hat{\mathbf{z}}^{[l]} \tag{6.14}$$

where $Bern(\mathrm{P}_{drop})$ is the Bernoulli random variable which takes the value 0 with the dropout probability P_{drop} and 1 with the probability $1 - \mathrm{P}_{drop}$. For example, if the second and fifth hidden units are dropped out, then $\mathbf{d}^{[l]} = [1\,0\,1\,1\,0\,1\,\cdots\,1]$ and hence $\check{z}_2^{[l]}$ and $\check{z}_5^{[l]}$ are 0.

After passing through the L hidden layers, the output FC layer produces N output values whose dimension is matched with the number of total users. The output vector

$\mathbf{z}^{out}$ is given by

$$\mathbf{z}^{out} = \mathbf{W}^{out}\left(\tilde{\mathbf{z}} + \sum_{i=1}^{L} \tilde{\mathbf{z}}^{[i]}\right) + \mathbf{b}^{out}, \tag{6.15}$$

where $\mathbf{W}^{out} \in \mathbb{R}^{N\times\alpha}$ and $\mathbf{b}^{out} \in \mathbb{R}^{N\times 1}$ are the corresponding weight and bias, respectively. Then, the softmax layer maps N output values into N probabilities $(\hat{p}_1, \cdots, \hat{p}_N)$ representing the likelihood of being the true support element. The i-th probability $\hat{p}_i$ is given by

$$\hat{p}_i = \frac{e^{z_i^{out}}}{\sum_{j=1}^{N} e^{z_j^{out}}}, \qquad \text{for } i = 1, \cdots, N. \tag{6.16}$$

Finally, an estimate of the support $\hat{\Omega}$ is obtained by picking k elements having the largest probabilities:

$$\hat{\Omega} = \arg\max_{|\Omega|=k} \sum_{i\in\Omega} \hat{p}_i. \tag{6.17}$$

6.3.2 D-AUD Training

In the training phase, we use the training dataset to find out the network parameter set Θ^* minimizing the loss function $J(\Theta)$ (i.e., $\Theta^* = \arg\min_\Theta J(\Theta)$). When the loss function $J(\Theta)$ is differentiable, network parameters can be updated by the gradient descent method in each training iteration. Specifically, parameters in the j-th training iteration Θ_j are updated simultaneously in the direction of the steepest descent:

$$\Theta_j = \Theta_{j-1} - \eta\nabla_\Theta J(\Theta), \tag{6.18}$$

where $\nabla_\Theta J(\Theta)$ is the gradient of $J(\Theta)$ with respect to Θ and η is the learning rate determining the step size.

Recalling that the final output of the D-AUD scheme is the N-dimensional vector $\hat{\mathbf{p}}$ whose element represents the probability of being the support element, $\hat{\mathbf{p}} = [\hat{p}_1, \cdots, \hat{p}_N]$ needs to be compared against the true probability $\mathbf{p}$ in the loss function calculation. Since k active users are assumed to be equiprobable, we set the true

probability as $p_i = \frac{1}{k}$ for $i \in \Omega$ and $p_i = 0$ for the rest. For example, when the second and fourth devices are active (i.e. $k = 2$ and $\Omega = \{2, 4\}$), $p_2 = p_4 = \frac{1}{2}$ and $p_i = 0, i \notin \{2, 4\}$. In the generation of the loss function, we use the cross entropy loss $J(\mathbf{p}, \hat{\mathbf{p}})$ defined as [12]

$$J(\mathbf{p}, \hat{\mathbf{p}}) = -\sum_{i=1}^{N} p_i \log \hat{p}_i = -\frac{1}{k} \sum_{j=1}^{k} \log \hat{p}_{\omega_j}, \tag{6.20}$$

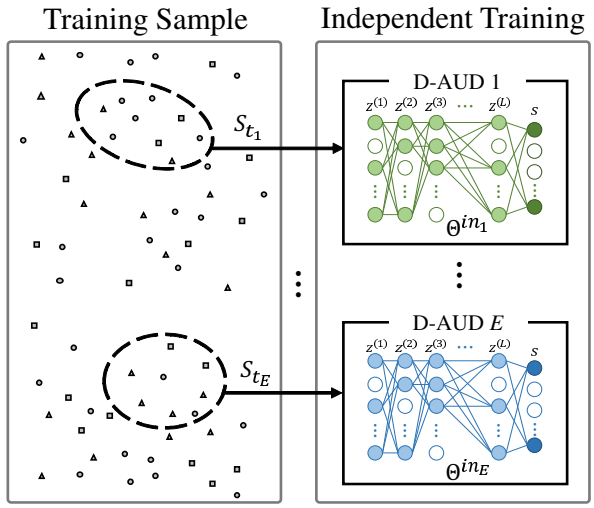
where $\omega_j \in \Omega$. In order to minimize $J(\mathbf{p}, \hat{\mathbf{p}})$, $\sum_{j=1}^{k} \log \hat{p}_{\omega_j}$ should be maximized. Since the sum of softmax output values is 1 (i.e., $\sum_i \hat{p}_i = 1$), the maximum can be achieved when $\hat{p}_{\omega_j} = \frac{1}{k}$, which is the desired training result.

One well-known problem in the training phase of DNN is the overfitting. By overfitting, we mean that the designed D-AUD is so closely fitted to the training set and thus it does not make reasonable prediction for the unobserved data. Indeed, when a user not participated in the training process transmits a packet, the overfitted neural network might fail to detect the user. In order to prevent this problem, we use multiple independently trained networks in the output generation. In this scheme, often called *ensemble* technique [84], multiple, say E, D-AUD networks are trained independently with the different training sets $(S_{t_1}, \cdots, S_{t_E})$ and initial parameters $(\Theta^{in_1}, \cdots, \Theta^{in_E})$ (see Fig. 6.7). Thus, from the same set of measurements, E independent output probabilities $(\hat{\mathbf{p}}^{(1)}, \cdots, \hat{\mathbf{p}}^{(E)})$ are generated. By averaging out these probabilities, we obtain

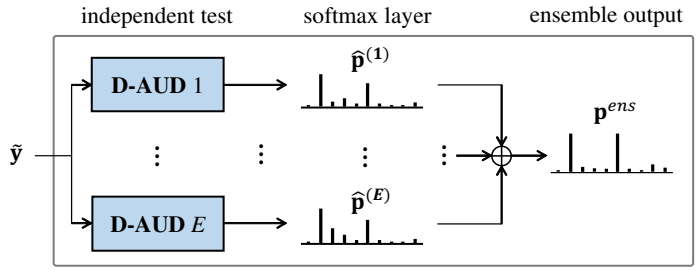
[12] Kullback-Leibler (KL) divergence can be also used for the loss function. Using the KL divergence loss $D_{KL}(\mathbf{p} \parallel \hat{\mathbf{p}})$, we have

$$\begin{aligned} D_{KL}(\mathbf{p} \parallel \hat{\mathbf{p}}) &= \sum_{i=1}^{N} p_i \log \frac{p_i}{\hat{p}_i} = -\sum_{i=1}^{N} p_i \log \hat{p}_i + \sum_{i=1}^{N} p_i \log p_i \\ &= J(\mathbf{p}, \hat{\mathbf{p}}) + \sum_{j=1}^{k} p_{\omega_j} \log p_{\omega_j} = J(\mathbf{p}, \hat{\mathbf{p}}) - \log k. \end{aligned} \tag{6.19}$$

Since $\log k$ is a constant, to minimize the cross entropy loss $J(\mathbf{p}, \hat{\mathbf{p}})$ is essentially the same as to minimize the KL divergence loss. The regression loss function (e.g., mean square error (MSE) and the mean absolute error (MAE)), however, might not be suitable for the D-AUD training since it is used for estimating a specific value.



(a)



(b)

Figure 6.7: Description of the ensemble network: (a) training phase for independent D-AUD scheme with different training set and (b) ensembling test phase using the independently trained D-AUD schemes.

Table 6.1: Comparison of Computational Complexity ($N = 80, m = 40, \alpha = 500, L = 6$)

	the number of floating point operations (flops)	Complexity for various sparsity		
		k=6	k=8	k=10
D-AUD	$2L\alpha^2 + (4m + 7L + 2N + 4)\alpha + (k+3)N - \frac{k(k+1)}{2} - 1$ $+2m + k\left(\frac{14}{3}m^3 + m^2 - m\right)$	4.99×10^6	5.59×10^6	6.19×10^6
MMSE-BOMP	$2km^2N - k + 2km + \frac{k(k+1)}{2}\left(\frac{14}{3}m^3 + 3m^2 - m\right) + k(k+1)m^2$	7.91×10^6	1.30×10^7	1.92×10^7
LS-BOMP	$2km^2N + \frac{k^4+6k^3+7k^2+2k}{12}m^3 + k(k+1)m^2 - k$	1.68×10^7	4.29×10^7	9.19×10^7

the ensemble probability $\mathbf{p}^{ens}$ as

$$\mathbf{p}^{ens} = \frac{1}{E}\sum_{j=1}^{E} \hat{\mathbf{p}}^{(j)}. \tag{6.21}$$

Finally, an estimate of the support is obtained by picking indices of k largest values in $\mathbf{p}^{ens}$. One can observe that the ensemble technique is conceptually analogous to the receiver diversity technique in wireless communication systems in the sense that it is performed in the BS side and also does not require additional wireless resources (e.g., frequency, time, and transmission energy) in the mobile side.

6.3.3 Comments on Complexity

In this subsection, we analyze the computational complexity of the proposed D-AUD scheme. In our analysis, we measure the complexity in terms of the number of floating point operations (flops). Initially, in the FC layer, the input vector $\hat{\mathbf{y}} \in \mathbb{R}^{2m\times 1}$ is multiplied by the initial weight $\mathbf{W}^{in} \in \mathbb{R}^{\alpha\times 2m}$ and then the bias $\mathbf{b}^{in} \in \mathbb{R}^{\alpha\times 1}$ is added (see (6.9)). The complexity of the initial FC layer $\mathcal{C}_{\text{in}}$ is

$$\mathcal{C}_{\text{in}} = (4m-1)\alpha + \alpha = 4m\alpha. \tag{6.22}$$

Since the element-wise scalar multiplication and addition are performed twice in the batch normalization process (see (6.10)), the complexity $\mathcal{C}_{\text{BN}}$ of batch normalization is simply

$$\mathcal{C}_{\text{BN}} = 4\alpha. \tag{6.23}$$

Next, in the hidden layer, an input vector is multiplied by the weight $\mathbf{W}^{[l]} \in \mathbb{R}^{\alpha \times \alpha}$ and then the bias $\mathbf{b}^{[l]} \in \mathbb{R}^{\alpha \times 1}$ is added (see (6.11)). After the batch normalization (4α flops), for each element, we test whether the value is larger than 0 using the ReLU function. The dropout vector $\mathbf{d}^{[l]}$ is multiplied to $\hat{\mathbf{z}}^{[l]}$ (see (14)) and then an output vector of the previous hidden layer is added to the output of the dropout layer for the residual connection. Therefore, the complexity $\mathcal{C}_{\text{hide}}$ of L hidden layers can be expressed as

$$\begin{aligned} \mathcal{C}_{\text{hide}} &= L\left((2\alpha - 1)\alpha + \alpha + 4\alpha + \alpha + \alpha + \alpha\right) \\ &= 2L\alpha^2 + 7L\alpha. \end{aligned} \tag{6.24}$$

After passing through L hidden layers, the weight multiplication and bias addition are performed in the output FC layer (see (6.15)). Since $\mathbf{W}^{out} \in \mathbb{R}^{N \times \alpha}$ and $\mathbf{b}^{out} \in \mathbb{R}^{N}$, the complexity $\mathcal{C}_{\text{out}}$ of the output FC layer is

$$\mathcal{C}_{\text{out}} = (2\alpha - 1)N + N = 2\alpha N. \tag{6.25}$$

Next, the softmax operation consisting of exponential computation (N flops), summation ($N - 1$ flops), and division (N flops) is performed (see (6.16)). The resulting computational complexity of the softmax operation is

$$\mathcal{C}_{\text{softmax}} = 3N - 1. \tag{6.26}$$

Finally, the complexity $\mathcal{C}_{\text{sort}}$ of taking k largest probabilities in $\mathbf{p}$ (see in (6.17)) is [24]

$$\mathcal{C}_{\text{sort}} = kN - \frac{k(k+1)}{2}. \tag{6.27}$$

From (6.22) to (6.27), the complexity $\mathcal{C}_{\text{D-AUD}}$ of D-AUD is summarized as

$$\mathcal{C}_{\text{D-AUD}} = \mathcal{C}_{\text{in}} + \mathcal{C}_{\text{BN}} + \mathcal{C}_{\text{hide}} + \mathcal{C}_{\text{out}} + \mathcal{C}_{\text{softmax}} + \mathcal{C}_{\text{sort}} \tag{6.28}$$

$$\begin{aligned} &= 2L\alpha^2 + (4m + 7L + 2N + 4)\alpha + (k+3)N \\ &\quad - \frac{k(k+1)}{2} - 1. \end{aligned} \tag{6.29}$$

In Table I, we compare the complexities of D-AUD, MMSE-BOMP, and LS-BOMP (see Appendix A for the detailed complexity derivation). For fair comparison, in the D-AUD, we add the complexity of the MMSE estimation $\mathcal{C}_{MMSE} = 2m+k\left(\frac{14}{3}m^3+m^2-m\right)$ for the signal detection. In order to examine overall behavior, we compute the required flops for various sparsity levels ($k = 6, 8, 10$). We observe that the complexity of D-AUD is much smaller than that of conventional approaches. For example, when $k = 8$, the complexity of the D-AUD is 57% and 87% lower than those of MMSE-based BOMP and LS-based BOMP, respectively. It is worth mentioning that the complexity of D-AUD depends heavily on the DNN network parameters (L and α), not the system parameters (k and N). For instance, when k increases from 6 to 10, the computational complexity of D-AUD increases marginally but that of LS-BOMP increases sharply. One can observe from this that in the practical NOMA-based environment where the numbers of total users and active users (e.g., $N = 100$ and $k = 10$) are large, the D-AUD scheme is competitive in terms of the computational complexity.

6.4 Practical Issues for D-AUD Implementation

In this section, we go over two major issues when applying the D-AUD scheme in the practical scenarios. We first discuss the training data collection issue. This issue is crucial since the uplink traffics are usually unpredictable and sporadic so that it takes quite a bit of time and effort to collect the training data. We next discuss a sparsity estimation issue. In order to perform the symbol detection and decoding, the BS should know the sparsity (number of active devices) in a priori, which is clearly difficult for grant-free scenarios.

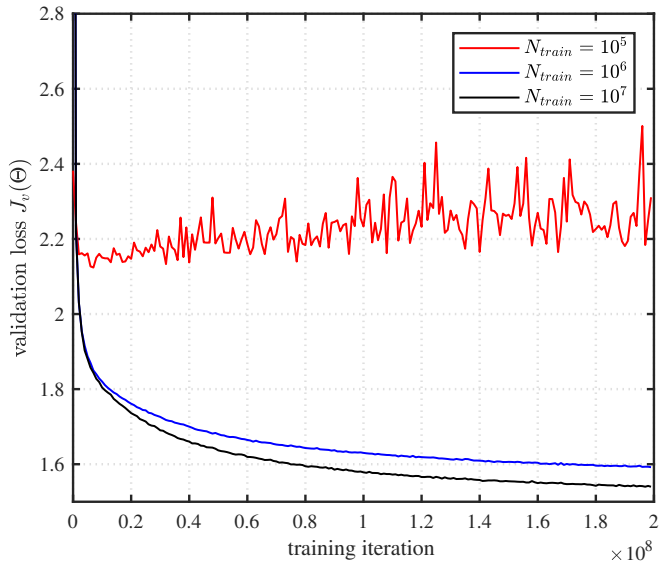


Figure 6.8: Validation loss $J_v(\Theta)$ for various number of training samples N_{train} ($k = 4$ and $m = 70$).

6.4.1 Training Data Collection

In order to learn the optimal mapping function g^* between the received signal and support, sufficient amount of training data is required. In Fig. 6.8, we plot the validation loss $J_v(\Theta)$ as a function of the training iteration for various sizes of training dataset. We see that when the number of training dataset is not enough, the deep neural network does not converge, causing a failure in the D-AUD training. In acquiring the dataset, one can naturally consider using the *real* received signals. Unfortunately, this requires too many training data transmissions. For example, when collecting one million received signals in LTE systems, it will take around 30 minutes (1 ms subframe consisting of 14 symbols). This time will further increase in proportion to the number of ensemble networks. Therefore, this type of data collection strategy is by no means practical in terms of energy consumption, latency, and resource utilization efficiency.

In order to reduce the overhead associated with the training data transmission, we

use synthetically generated signals as the training dataset at BS. One might concern that the synthetically generated signal is different from the actual transmitted signal since the channel depends heavily on the environmental factors such as frequency band, mobility, and geometric objects. Fortunately, we can circumvent this issue since the AUD process is essentially the same as the support identification and all channel components are contained in an input sparse vector $\mathbf{x}$, not the system matrix $\mathbf{\Phi}$ (see (6.5)). Thus, the D-AUD scheme only needs to learn the codebook matrix $\mathbf{\Phi}$ (which is known a priori), not the individual channel state, which will ease the training process significantly. Indeed, what we need to do in the training phase is to artificially generate the received vector in (6.5). In doing so, time and effort to collect huge training data can be saved and at the same time the training process can be done offline.

Since the training operation of D-AUD is performed offline using the synthetically generated data, we train multiple D-AUD networks for various settings in terms of the number of total users and the number of active users. From this process, we can obtain the internal parameters (e.g., weight and bias) for each scenario. When applying the D-AUD to the actual transmission, we thus use the pre-trained network corresponding to the system environment. Even though the system environment will vary, as long as we have a trained model matching to the environment, re-training process is unnecessary. For example, we train two D-AUD schemes for two different number of total users ($N = 50$, and 100). In the test phase (real operation phase), when N changes from 30 to 80, what we need to do is to change the trained model for $N = 50$ to that for $N = 100$.

6.4.2 Sparsity Estimation

In the grant-free transmission, devices can transmit the data without the granting process so that the BS needs to be aware of the sparsity to perform the AUD. Since the sparsity is used as the number of iterations in many sparse recovery algorithms, incorrect sparsity leads to either miss detection (early termination) or false alarm (late

Algorithm 1. Sparsity estimation in the proposed D-AUD scheme

Input: the received signal $\hat{\mathbf{y}} \in \mathbb{R}^{2m\times 1}$, the trained threshold $\tau \in \mathbb{R}$, the maximal sparsity $U \in \mathbb{R}$

Output: the estimated sparsity $\hat{k}$, the estimated support $\hat{\Omega}$

Initialization: $l = 0, \Gamma = \{1, \cdots, N\}$

1: **while** $l \leq U$ and $l \neq |\Gamma|$ **do**

2: $\quad l = l + 1$

3: $\quad$ Obtain $\hat{\mathbf{p}}^{(l)}$ by passing $\hat{\mathbf{y}}$ into the D-AUD network trained for sparsity level l

4: $\quad \hat{p}_{\max}^{(l)} = \max_i \hat{p}_i^{(l)}$

5: $\quad \Gamma = \left\{ i \in \{1, \cdots, N\} \;\middle|\; \frac{\hat{p}_i^{(l)}}{\hat{p}_{\max}^{(l)}} \geq \tau \right\}$

6: **end while**

7: $\hat{k} = l$

8: $\hat{\Omega} = \Gamma$

Return: $\hat{k}$, $\hat{\Omega}$

termination). In the former case, some of active devices cannot be identified while inactive devices can be chosen as active devices for the latter case. Therefore, the sparsity estimation error degrades the support identification quality substantially[13].

In the proposed D-AUD scheme, instead of using an iterative support identification, k support elements are chosen from the softmax output (see (6.17)). Thus, in contrast to the conventional sparse recovery algorithms, a separate sparsity estimation process is unnecessary. One simple option to choose the support is to take the indices of the softmax output values being larger than the threshold τ. Benefit of this approach is that τ can be readily chosen in the training phase since the support Ω and sparsity

[13] As a sparsity estimation strategy, the residual-based stopping criterion is widely used [2]. In this scheme, basically, an algorithm is terminated when the residual power $\|\mathbf{r}\|_2$ is smaller than the pre-specified threshold ϵ (i.e., $\|\mathbf{r}\|_2 < \epsilon$) and the iteration number at the termination point is set to the sparsity level. However, since the residual magnitude decreases monotonically and the rate of decay depends on the system parameters, it might not be easy to figure out an accurate terminating point.

k of training data are already available. When determining τ, we use both softmax output $\hat{\mathbf{p}}$ and sparsity k of training data. Specifically, in the training phase, we obtain the softmax values $\hat{p}_{\omega_1}, \cdots, \hat{p}_{\omega_k}$ for $\omega_i \in \Omega$. Note that these values would be close to $\frac{1}{k}$ since the D-AUD is trained to generate the true probability $p_{\omega_i} = \frac{1}{k}$. In order to remove the effect of k (meaning that τ is set to be independent of k), we scale $\hat{p}_{\omega_1}, \cdots, \hat{p}_{\omega_k}$ by k and then set the minimum value to τ (i.e., $\tau = \min_i k\hat{p}_{\omega_i}$). In doing so, in the test phase, we can identify the support without the knowledge of the sparsity k. To be specific, by using multiple D-AUD networks for U distinct sparsity levels, we obtain U softmax output vectors $\hat{\mathbf{p}}^{(l)} = [\hat{p}_1^{(l)}, \cdots, \hat{p}_N^{(l)}]$ for $l = 1, \cdots, U$. Then, we take indices satisfying $\frac{\hat{p}_i^{(l)}}{\hat{p}_{\max}^{(l)}} \geq \tau$ ($\hat{p}_{\max}^{(l)} = \max_i \hat{p}_i^{(l)}$ is the maximal value of $\hat{\mathbf{p}}^{(l)}$). If the number of the chosen indices is l, then we set the sparsity to l (i.e., $\hat{k} = l$) and then obtain estimated support $\hat{\Omega}$. The proposed sparsity estimation in the D-AUD scheme is summarized in Algorithm 1.

6.5 Simulations and Discussions

6.5.1 Simulation Setup

In this section, we investigate the performance of the proposed D-AUD scheme. Our simulation setup is based on the grant-free NOMA transmission in the orthogonal frequency division multiplexing (OFDM) systems. Specifically, we use 100 users ($N = 100$) and 70 subcarriers ($m = 70$) in each transmission so that the overloading factor is 143%. As a channel model, the pathloss component γ_i between the i-th device and the BS is modeled as $\gamma_i = 128.1 + 37.6 \log_{10}(d_i)$ [dB] where d_i is the distance (in km) between the i-th device and the BS [85] and independent Rayleigh fading coefficient is used for each device [86]. The noise spectral density and transmission bandwidth are set to -170 dBm/Hz and 1 MHz, respectively. For comparison, we examine the performance of the conventional LS-BOMP [2], MMSE-BOMP [87], and approximate message passing (AMP) algorithm [88]. When generating nonzero values in the LDS

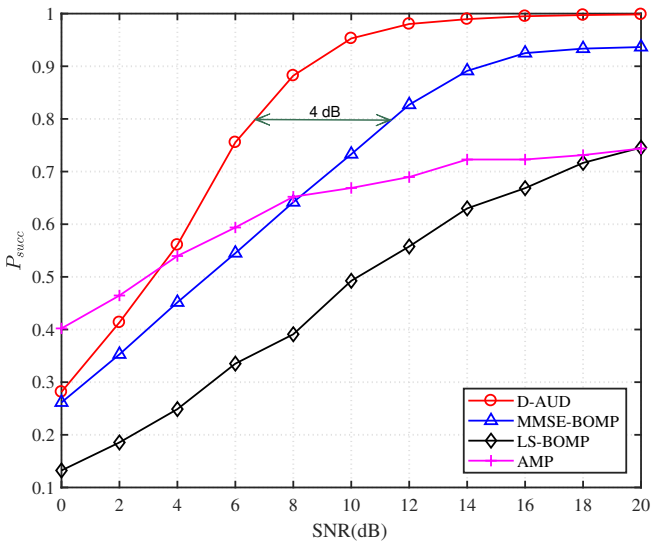


Figure 6.9: P_{succ} as a function of SNR ($N = 100, k = 4, N_d = 7, m = 70$).

codebook, we use an i.i.i. Gaussian random variable[14]. Length of the LDS codeword S is set to 10 ($S = 10$).

In order to guarantee the model stability of the D-AUD scheme, we use K-fold cross validation in the training phase. In the K-fold cross validation, total samples are randomly partitioned into K equal-sized sets. Among K partitioned sets, a single set is used for the model testing, and the remaining $K - 1$ sets are used for the D-AUD training. Then, this process is repeated $K - 1$ times for the remaining sets. In our simulations, we generate 10^7 samples and set $K = 10$. When selecting the hyperparameters, we use the cross-validation technique (see Fig. 12). In our simulations, we use an Adam optimizer, well-known optimization tool to guarantee the robustness of learning process [89]. As an activation function in hidden layers, we used ReLU function. Also, we set $L = 6$ (the number of hidden layers), $\alpha = 1000$ (the width of hidden layer), $P_{drop} = 0.1$ (dropout probability), $\eta = 5 \times 10^{-4}$ (learning rate),

[14]When the different set of spreading sequences is used, we need to re-train the proposed D-AUD scheme using the new training data (generated from the new spreading sequences).

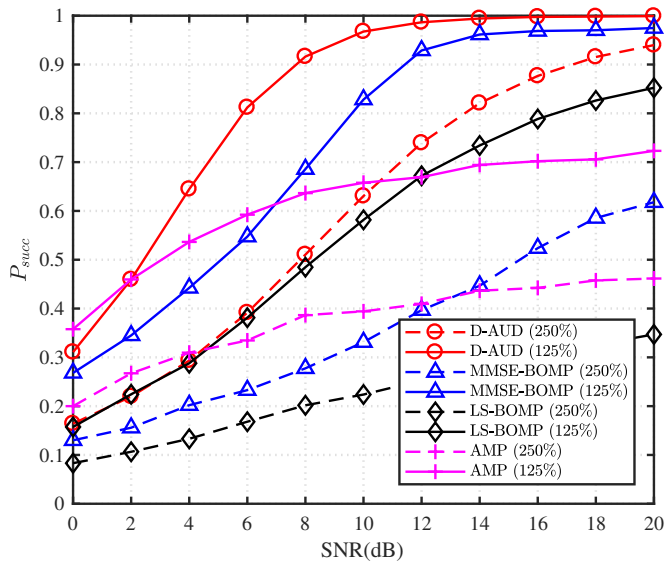


Figure 6.10: P_{succ} as a function of SNR with various overloading factor ($N = 100, k = 4$).

and $E = 3$ (the number of ensemble networks). As a performance metric, we use the success probability P_{succ} which corresponds to the percentage of the detected users among all active users.

6.5.2 Simulation Results

In Fig. 6.9, we evaluate P_{succ} of the proposed D-AUD scheme and competing AUD schemes as a function of SNR. We observe that D-AUD outperforms the conventional schemes for all SNR regime. Since D-AUD learns the mapping between the received signal $\tilde{\mathbf{y}}$ and the support Ω, an estimate of support $\hat{\Omega}$ can be determined only by the input data $\tilde{\mathbf{y}}$. This means that the whole AUD process can be handled by a simple end-to-end mapping in D-AUD. For example, we observe that D-AUD achieves around 6 dB gain over the MMSE-BOMP at $P_{succ} = 0.9$.

In Fig. 6.10, we investigate P_{succ} for various overloading factors. We can clearly

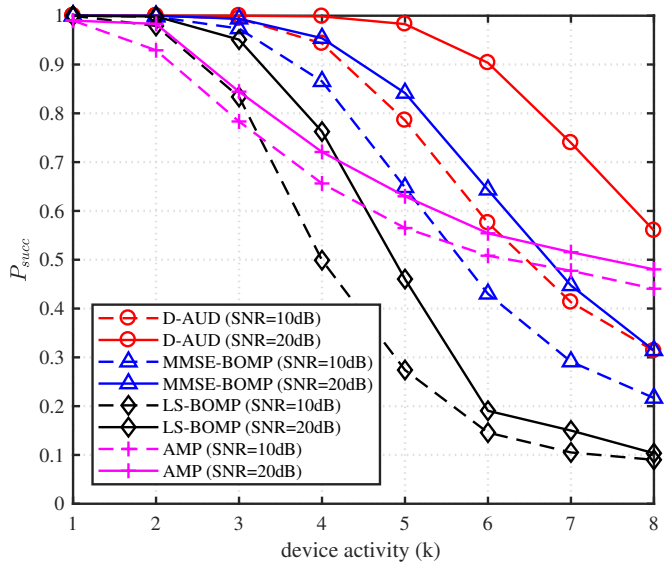


Figure 6.11: P_{succ} as a function of k with 2 different SNR ($N = 100, N_d = 7, m = 70$).

see that D-AUD outperforms the conventional AUD approaches by a large margin. For example, in case of 125% overloading, D-AUD achieves around 4 dB gain over the MMSE-BOMP at $P_{succ} = 0.9$. We also observe that the AUD performance of D-AUD is robust to the overloading factor due to the decoupling of the correlated activation patterns (see Section III). For instance, in case of 250% overloading, D-AUD achieves $P_{succ} = 0.9$ at SNR $= 17.5$ dB. Since there is no such mechanism for the conventional sparse recovery algorithms, performance of conventional schemes is not appealing when the overloading factor is high.

In Fig. 6.11, we plot the P_{succ} as a function of the number of active devices k. We observe that D-AUD outperforms conventional schemes across the board. For example, when the number of active devices is 6 (i.e., $k = 6$) and SNR $= 20$ dB, P_{succ} of D-AUD is 0.9 while those of the MMSE-BOMP and AMP are 0.65 and 0.55, respectively. Also, we can see that the D-AUD maintains its robustness even when k increases. When k increases, the mutual correlation associated with the active devices increases,

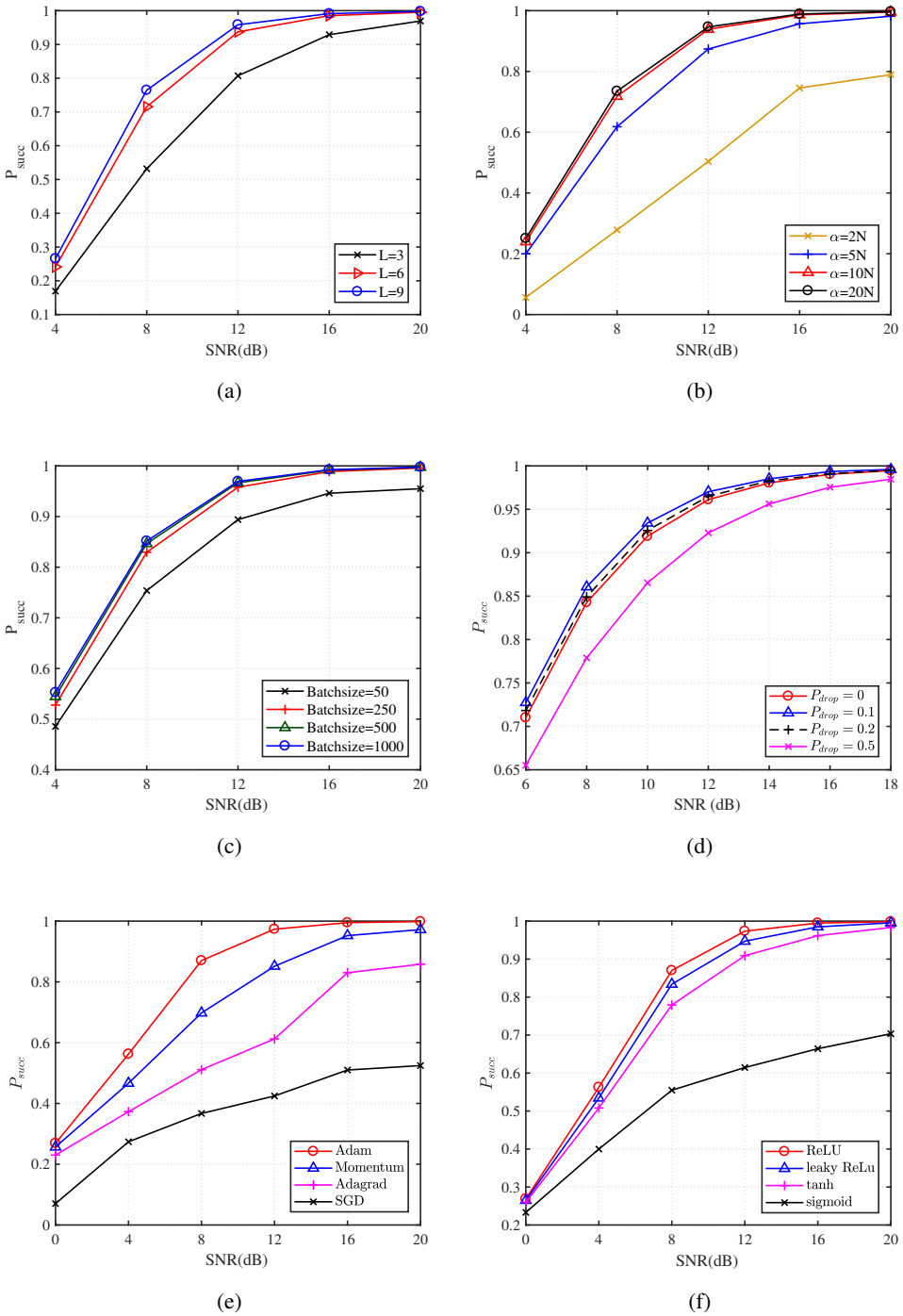


Figure 6.12: An example of hyperparameter tuning process: (a) depth of hidden layers, (b) width of hidden layers, (c) batch size, (d) dropout probability, (e) optimizer, and (f) activation function

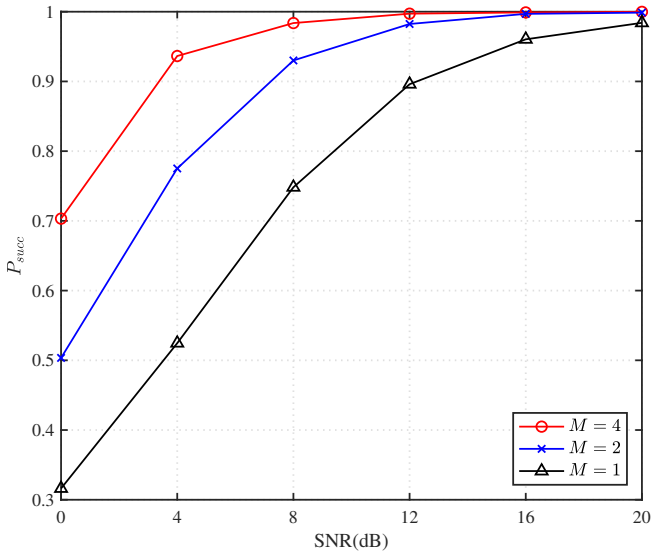


Figure 6.13: P_{succ} as a function of SNR in the multi-antenna scenario ($N = 100, k = 4$).

causing a severe degradation of the AUD performance (see Section II). Since the DNN already learned the correlation feature from the training dataset, D-AUD can better discriminate the correlated supports in the test phase. For example, when k increases from 3 to 6, P_{succ} of the D-AUD decreases marginally from 0.99 to 0.91. However, P_{succ} of the MMSE-BOMP and LS-BOMP decrease sharply from 0.98 to 0.65 and from 0.95 to 0.19, respectively.

In Fig. 6.12, we evaluate P_{succ} for various hyperparameters such as depth and with of hidden layers, batch size, dropout probability, activation function, and optimizer. From these results, we can observe the effect of each hyperparameter on the AUD performance. For example, if the width of the hidden layer is small (e.g., $\alpha = 2N$ case), we expect that the performance of D-AUD will degrade considerably. Whereas, if the width is larger than $10N$, the D-AUD performance will not be improved further. From this offline tuning process, we can obtain the right hyperparameters of D-AUD.

Finally, in order to test the performance of D-AUD scheme in multiple-antenna

scenarios[15], we consider the three distinct cases (i.e., number of received antennas is $M = 1, 2$, and 4). As shown in Fig. 13, we observe that the performance of D-AUD improves with M. For example, when $M = 4$, we observe 8.4 dB gain over the single received antenna scenario ($M = 1$) at $P_{succ} = 0.9$. Since the active users are detected blindly (without the channel information), the BS cannot achieve the gain proportional to the number of antennas. Nevertheless, the multi-antenna gain proportional to the number of antennas (around 2.1 dB gain per antenna) can be achieved.

6.6 Summary

In this paper, we proposed a DNN-based AUD scheme called D-AUD for the mMTC uplink scenario. Our work is motivated by the observation that CS-based AUD cannot support the massive number of devices and high device activity scenario in the grant-free NOMA systems. By feeding the training data to the properly designed DNN, the proposed D-AUD scheme learns the nonlinear mapping between the received signal and support. As long as we train the deeply stacked hidden layers using a proper loss function and the backpropagation mechanism, we can detect active devices in the test phase. We demonstrated from numerical evaluations that the proposed D-AUD scheme is very effective in the highly-overloaded mMTC scenarios. In this paper, we restricted our attention to the AUD but we believe that there are many interesting applications of the proposed approaches such as DoA estimation, mmWave channel estimation, and MIMO detection.

[15]When using M antenna at the BS, the input of the D-AUD scheme will become multiple measurement vectors $\tilde{\mathbf{y}}_{(1)}, \cdots, \tilde{\mathbf{y}}_{(M)}$, not a single measurement vector $\tilde{\mathbf{y}}$. Accordingly, the AUD problem, originally modeled as single measurement vector (SMV) problem, will be also converted to the multiple measurement vector (MMV) problem. From the equation (5), we can obtain the MMV model $\tilde{\mathbf{Y}} = \mathbf{\Phi X}$ where $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_{(1)} \ \cdots \ \tilde{\mathbf{y}}_{(M)}]$ and $\mathbf{X} = [\mathbf{x}_{(1)} \ \cdots \ \mathbf{x}_{(M)}]$. Since the supports of $\mathbf{x}_{(1)}, \cdots, \mathbf{x}_{(M)}$ are common, we can exploit the correlation among them in the recovery process.

Chapter 7

Conclusion

In this dissertation, sparse-aware wireless communications to employ the compressed sensing and the deep learning techniques have been extensively studied. Although the sparse system model has been popularly used in the mmWave channel estimation and the spectrum sensing, I focused on various applications including channel-aware sparse transmission in TDD systems, partial sample transmission in V2X systems, and active user detection in mMTC. These new extensions enables to transmit data efficiently, reduce the physical-layer latency, and transmit highly accurate information, which provide guidelines for designing future wireless systems. Specifically, I make the following contributions:

- In Chapter 2, we have introduced new type of short packet transmission scheme referred to as sparse vector transmission (SVT). Key idea of SVT is to transmit the short-sized information after the sparse vector transformation. Using the principle of compressed sensing (CS), we decode the packet using a small number of resources. SVT has a number of advantages over the conventional transmission strategies; it is simple to implement, reduces the transmission latency as well as the encoding/decoding complexity. When the position of a sparse vector is used to encode the information exclusively, decoding can be done without the channel knowledge, saving the pilot transmission overhead and the channel esti-

mation effort. Further, SVT can inherently improve the user identification quality and security. In a nutshell, SVT is a viable solution for massive machine-type communication (mMTC) and URLLC scenarios having many advantages over the conventional packet transmission mechanism.

- In Chapter 3, we proposed the ultra low latency access scheme based on the CAST for URLLC. The key idea behind the proposed CAST scheme is to transform a URLLC grant information into the sparse vector and to exploit the sparse recovery algorithm in decoding process. As long as the number of subcarriers is small enough and the measurements contain enough information to figure out the support and decode the grant information, accurate decoding of the CAST scheme can be guaranteed. We demonstrated from the numerical evaluations that the proposed CAST scheme is very effective in TDD-based URLLC. In this paper, we restricted our attention to the URLLC scenario but we believe that there are many interesting extensions worth investigating, such as the diversity support, machine learning-based CAST, and CAST for the FDD systems.

- In Chapter 4, we proposed a novel low-latency transmission scheme suitable for the URLLC-based V2X scenarios. The key idea behind the proposed PST scheme is to encode the mission-critical sidelink information in a form of the sparse symbol vector and then decode the information with a partially-buffered samples using deep learning-based decoder. When the number of subcarriers is small enough and the measurements contain enough information to figure out the transmit information, accurate decoding of the PST-encoded packet can be guaranteed. In the decoding process, we exploit the DNN architecture to learn the nonlinear mapping between the received signal vector and the support of input sparse signal. As long as we train the deeply stacked network using a properly designed loss function along with the backpropagation weight update, we can identify the accurate support in the test phase. We demonstrated from the

numerical evaluations that the proposed D-PST is very effective in terms of both the reliability and latency. In this paper, we restricted our attention to the V2X sidelink transmission but we believe that there are many interesting applications of the proposed approaches in mission-critical machine-type communications.

- In Chapter 5, we have discussed two major challenges in the design of DL-based wireless communication system, mainly related to the dataset collection and neural network architecture. For each issue, we have provided the learning-based solutions which can be easily implemented in practice. First, in order to collect the sufficient training data, we basically have three options: collection from the actual received signals, synthetic data generation using the analytic system model, and real-like training set generation using generative adversarial network (GAN). Second, when designing the DNN architecture, we need to consider the input characteristics, wireless environments, and system configurations. Other than the solutions we have discussed, there are many interesting issues worth exploring such as distributed learning and meta learning. We believe that future researches will be more extended for the problems we have discussed and the other intriguing problems.

- In Chapter 6, we proposed a DNN-based AUD scheme called D-AUD for the mMTC uplink scenario. By feeding the training data to the properly designed DNN, the proposed D-AUD scheme learns the nonlinear mapping between the received signal and support. As long as we train the deeply stacked hidden layers using a proper loss function and the backpropagation mechanism, we can detect active devices in the test phase. We demonstrated from numerical evaluations that the proposed D-AUD scheme is very effective in the highly-overloaded mMTC scenarios. In this paper, we restricted our attention to the AUD but we believe that there are many interesting applications of the proposed approaches such as DoA estimation, mmWave channel estimation, and MIMO detection.

Chapter A

Proof of (3.14)

Before we proceed to the main results, we provide the useful properties of the column correlation of $\mathbf{A}$ in (3.9).

Lemma 5 *Recall that* $f(|\omega_p - \omega_q|) = \left|\langle \mathbf{a}_{\omega_p}, \mathbf{a}_{\omega_q} \rangle\right| = \frac{1}{m}\left|\frac{\sin\frac{\pi m(\omega_p-\omega_q)}{N}}{\sin\frac{\pi(\omega_p-\omega_q)}{N}}\right|$ *is the column correlation between* $\mathbf{a}_{\omega_p}$ *and* $\mathbf{a}_{\omega_q}$ *(see* (3.9)*). Then the following statements hold true:*

(i) If $|\omega_p - \omega_q| = \frac{N}{m}, \frac{2N}{m}, \cdots, \frac{(m-1)N}{m}$*, then* $f(|\omega_p - \omega_q|) = 0$.

(ii) $f(|\omega_p - \omega_q|) \leq \frac{1}{m\left|\sin\frac{\pi(2i+1)}{2m}\right|}$ *for some integer* $i \geq 0$ *satisfying* $\max\left\{\frac{N}{2m}, \frac{iN}{m}\right\} \leq |\omega_p - \omega_q| \leq \frac{(i+1)N}{m}$.

Proof: In order to prove this proposition, we express the success probability $\mathrm{P}(S^1)$ in terms of the column correlation of $\mathbf{A}$. Specifically, let $\omega^* = \arg\max_{1\leq\omega\leq N} |\langle \mathbf{a}_\omega, \tilde{\mathbf{y}} \rangle|$ be the index chosen in the first iteration. Then, the first iteration would be successful if there exists only one $\omega \in \Omega = \{\omega_1, \cdots, \omega_k\}$ satisfying $|\omega^* - \omega| < \tau$ (see Fig. 3.5). Thus, we have

$$\mathrm{P}(S^1) = \mathrm{P}(|\omega^* - \omega| < \tau, \text{ for some } \omega \in \Omega).$$

Since the distance between two adjacent support elements is $\frac{N}{m}$ from Lemma 5(i), one can notice that τ should satisfy $\tau \leq \frac{N}{2m}$. For analytic simplicity, we set $\tau = \frac{N}{2m}$ in our

work. Then we have

$$\begin{aligned}
\mathrm{P}(S^1) &= \mathrm{P}\left(|\omega^* - \omega| < \frac{N}{2m}, \text{ for some } \omega \in \Omega\right) \\
&= 1 - \mathrm{P}\left(|\omega^* - \omega_i| \geq \frac{N}{2m}, \text{ for all } \omega_i \in \Omega\right) \\
&= 1 - \mathrm{P}\left(|\omega^* - \omega_1| \geq \frac{N}{2m}, \cdots, |\omega^* - \omega_k| \geq \frac{N}{2m}\right). \qquad \text{(A.1)}
\end{aligned}$$

First, we will find an upper bound of $\mathrm{P}\left(|\omega^* - \omega_1| \geq \frac{N}{2m}, \cdots, |\omega^* - \omega_k| \geq \frac{N}{2m}\right)$. Let $\delta_1 = \left[\frac{N}{2m}, \frac{N}{m}\right]$ and $\delta_i = \left(\frac{(i-1)N}{m}, \frac{iN}{m}\right]$ for $i = 2, 3, \cdots$, then $\Delta = \{\delta_1, \delta_2, \cdots\}$ is a partition of the interval $\left[\frac{N}{2m}, \infty\right)$. In this setting, it is clear that $|\omega^* - \omega_i|$ belongs to one interval in Δ. In other words, $|\omega^* - \omega_1| \in \delta_{\omega_1}, \cdots, |\omega^* - \omega_k| \in \delta_{\omega_k}$ where $\delta_{\omega_p} = \left(\max\left\{\frac{N}{2m}, \frac{i_{\omega_p}N}{m}\right\}, \frac{(i_{\omega_p}+1)N}{m}\right]$ for some $i_{\omega_p} \geq 0$ (see Fig. A). Therefore,

$$\begin{aligned}
&\mathrm{P}\left(|\omega^* - \omega_1| \geq \frac{N}{2m}, \cdots, |\omega^* - \omega_k| \geq \frac{N}{2m}\right) \\
&= \mathrm{P}\Big(\max\left\{\frac{N}{2m}, \frac{i_{\omega_1}N}{m}\right\} \leq |\omega^* - \omega_1| \leq \frac{(i_{\omega_1}+1)N}{m}, \\
&\qquad \text{for some } i_{\omega_1}, \cdots, \\
&\qquad \max\left\{\frac{N}{2m}, \frac{i_{\omega_k}N}{m}\right\} \leq |\omega^* - \omega_k| \leq \frac{(i_{\omega_k}+1)N}{m}, \\
&\qquad \text{for some } i_{\omega_k}\Big) \\
&\overset{(a)}{\leq} \mathrm{P}\Big(f(|\omega^* - w_1|) \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_1}+1)}{2m}\right|}, \cdots, \\
&\qquad f(|\omega^* - w_k|) \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_k}+1)}{2m}\right|}\Big) \\
&= \mathrm{P}\Big(|\langle \mathbf{a}_{\omega^*}, \mathbf{a}_{w_1}\rangle| \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_1}+1)}{2m}\right|}, \cdots, \\
&\qquad |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_{w_k}\rangle| \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_k}+1)}{2m}\right|}\Big) \\
&\leq \mathrm{P}\left(\sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_{w}\rangle| \leq \sum_{p=1}^{k} \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_p}+1)}{2m}\right|}\right) \qquad \text{(A.2)}
\end{aligned}$$

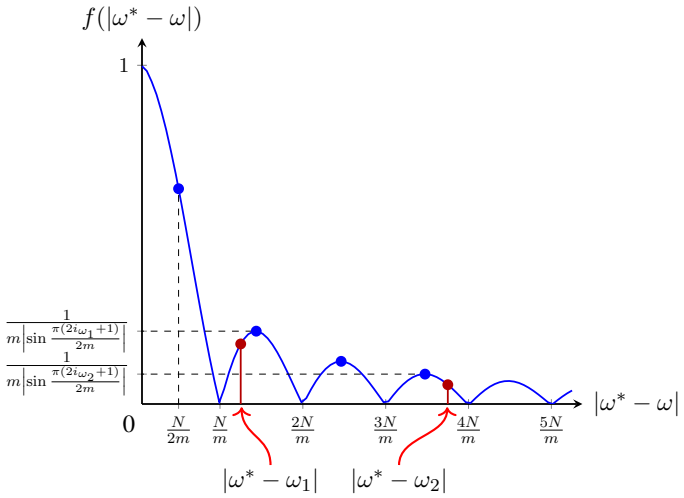


Figure A.1: If $|\omega^* - \omega_p| \geq \frac{N}{m}$, there exists a local maximum of $f(|\omega^* - \omega_p|)$ such that $f(|\omega^* - \omega_p|) \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_p}+1)}{2m}\right|}$. For example, if $\frac{N}{m} \leq |\omega^* - \omega_1| \leq \frac{2N}{m}$, then $f(|\omega^* - \omega_1|) \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_1}+1)}{2m}\right|}$. In a similar way, if $\frac{3N}{m} \leq |\omega^* - \omega_2| \leq \frac{4N}{m}$, then $f(|\omega^* - \omega_2|) \leq \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_2}+1)}{2m}\right|}$.

where (a) is from Lemma 5(ii). From (A.1) and (A.2), we have

$$\begin{aligned} \mathrm{P}(S^1) &\geq \mathrm{P}\left(\sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_w\rangle| \geq \sum_{p=1}^{k} \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_p}+1)}{2m}\right|}\right) \\ &= \mathrm{P}\left(\sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_w\rangle| \geq \rho\right) \end{aligned}$$

where $\rho = \sum_{p=1}^{k} \frac{1}{m\left|\sin\frac{\pi(2i_{\omega_p}+1)}{2m}\right|}$. Note that $\tilde{\mathbf{y}} = \sum_{\omega\in\Omega} \mathbf{a}_\omega x_\omega + \tilde{\mathbf{v}} = \sum_{\omega\in\Omega} \mathbf{a}_\omega h_\omega s_\omega + \tilde{\mathbf{v}} = \sum_{\omega\in\Omega} \mathbf{a}_\omega \beta h_\omega \check{s}_\omega + \tilde{\mathbf{v}}$ where $\beta = \sqrt{\frac{2m\alpha}{k}}$ and $\check{s}_\omega$ is the normalized symbol. Let $|h_{\omega_l}| =$

$\max_{\omega\in\Omega} |h_\omega|$, then we have

$$\begin{aligned}
\mathrm{P}(S^1) &\geq \mathrm{P}\left(\beta\,|h_{\omega_l}| \sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_w\rangle| \geq \beta\,|h_{\omega_l}|\,\rho\right) && \text{(A.3)}\\
&= \mathrm{P}\Big(\beta\,|h_{\omega_l}| \sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_\omega\rangle| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle| \\
&\qquad \geq \beta\,|h_{\omega_l}|\,\rho + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\Big) && \text{(A.4)}\\
&\geq \mathrm{P}\Big(\beta \sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_\omega\rangle|\,|h_\omega| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle| \\
&\qquad \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\Big) && \text{(A.5)}\\
&= \mathrm{P}\Big(\sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_\omega\rangle|\,|x_\omega| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle| \\
&\qquad \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\Big) && \text{(A.6)}\\
&= \mathrm{P}\Big(\sum_{\omega\in\Omega} |\langle \mathbf{a}_{\omega^*}, \mathbf{a}_\omega\rangle x_\omega| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle| \\
&\qquad \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\Big) && \text{(A.7)}\\
&\geq \mathrm{P}\Big(\left|\sum_{\omega\in\Omega} \langle \mathbf{a}_{\omega^*}, \mathbf{a}_\omega\rangle x_\omega + \langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle\right| \\
&\qquad \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\Big) && \text{(A.8)}\\
&= \mathrm{P}\left(\left|\langle \mathbf{a}_{\omega^*}, \sum_{\omega\in\Omega} \mathbf{a}_\omega x_\omega + \tilde{\mathbf{v}}\rangle\right| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\right) && \text{(A.9)}\\
&= \mathrm{P}\left(|\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{y}}\rangle| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}}\rangle|\right) && \text{(A.10)}
\end{aligned}$$

where (A.6) is because $|x_\omega| = \beta\,|h_\omega|$ and (A.8) is from the triangular inequality.

Since $|\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{y}} \rangle| \geq |\langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{y}} \rangle|$, we further have

$$\begin{aligned}
\mathrm{P}(S^1) &\geq \mathrm{P}\left(|\langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{y}} \rangle| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle|\right) && \text{(A.11)} \\
&= \mathrm{P}\Big(\Big|\langle \mathbf{a}_{\omega_l}, \sum_{\omega\in\Omega} \mathbf{a}_\omega x_\omega + \tilde{\mathbf{v}} \rangle\Big| \\
&\qquad \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle|\Big) && \text{(A.12)} \\
&= \mathrm{P}\left(|x_{\omega_l} + \langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{v}} \rangle| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle|\right) && \text{(A.13)} \\
&\geq \mathrm{P}\left(|x_{\omega_l}| - |\langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{v}} \rangle| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle|\right) && \text{(A.14)} \\
&= \mathrm{P}\left(\beta\,|h_{\omega_l}| - |\langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{v}} \rangle| \geq \beta\rho\,|h_{\omega_l}| + |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle|\right) && \text{(A.15)} \\
&= \mathrm{P}\left(\beta\,|h_{\omega_l}| - |\langle \mathbf{a}_{\omega_l}, \tilde{\mathbf{v}} \rangle| - |\langle \mathbf{a}_{\omega^*}, \tilde{\mathbf{v}} \rangle| \geq \beta\rho\,|h_{\omega_l}|\right) && \text{(A.16)} \\
&\geq \mathrm{P}\left(\beta\,|h_{\omega_l}| - 2\|\tilde{\mathbf{v}}\|_2 \geq \beta\rho\,|h_{\omega_l}|\right) && \text{(A.17)} \\
&= \mathrm{P}\left(\sqrt{\frac{\alpha m}{2k}}\,(1-\rho)\,|h_{\omega_l}| \geq \|\tilde{\mathbf{v}}\|_2\right) && \text{(A.18)} \\
&= \mathrm{P}\left(\sqrt{\frac{\alpha m}{2k}}\,(1-\rho)\,\|\mathbf{h}\|_\infty \geq \|\tilde{\mathbf{v}}\|_2\right), && \text{(A.19)}
\end{aligned}$$

where (A.13) is because $|\langle \mathbf{a}_{\omega_l}, \mathbf{a}_{\omega_l} \rangle| = 1$ and $|\langle \mathbf{a}_{\omega_l}, \mathbf{a}_{\omega} \rangle| = 0$ for $\omega \in \Omega \setminus \{\omega_l\}$, (A.14) is from the triangular inequality, (A.17) is from the Cauchy-Schwarz inequality (i.e., $|\langle \mathbf{a}_\omega, \tilde{\mathbf{v}} \rangle| \leq \|\mathbf{a}_\omega\|_2 \|\tilde{\mathbf{v}}\|_2 = \|\tilde{\mathbf{v}}\|_2$), and (A.19) is because $\|\mathbf{h}\|_\infty = \max|\mathbf{h}| = |h_{\omega_l}|$. $\square$

Chapter B

Proof of (3.18)

Recall that in the second iteration, the proposed algorithm picks the remaining $k-1$ columns from the set of columns orthogonal to the column chosen in the first iteration[1]. Let Ψ be the index set of the orthogonal columns to $\mathbf{a}_{\omega^*}$. Then, we have

$$\mathrm{P}\left(S^2|S^1\right) = \mathrm{P}\left(\min_{\omega_i\in\Omega\setminus\{\omega^*\}} |\langle \mathbf{a}_{\omega_i}, \tilde{\mathbf{y}}\rangle|^2 > \max_{\omega_j\in\Psi\setminus\Omega} \left|\langle \mathbf{a}_{\omega_j}, \tilde{\mathbf{y}}\rangle\right|^2\right) \tag{B.1}$$

$$= \prod_{\omega_i\in\Omega\setminus\{\omega^*\}} \mathrm{P}\left(|\langle \mathbf{a}_{\omega_i}, \tilde{\mathbf{y}}\rangle|^2 > \max_{\omega_j\in\Psi\setminus\Omega} \left|\langle \mathbf{a}_{\omega_j}, \tilde{\mathbf{y}}\rangle\right|^2\right) \tag{B.2}$$

$$= \prod_{\omega_i\in\Omega\setminus\{\omega^*\}} \prod_{\omega_j\in\Psi\setminus\Omega} \mathrm{P}\left(|\langle \mathbf{a}_{\omega_i}, \tilde{\mathbf{y}}\rangle|^2 > \left|\langle \mathbf{a}_{\omega_j}, \tilde{\mathbf{y}}\rangle\right|^2\right). \tag{B.3}$$

Let $\omega_{i^*} = \arg\min_{\omega_i\in\Omega\setminus\{\omega^*\}} |\langle \mathbf{a}_{\omega_i}, \tilde{\mathbf{y}}\rangle|^2$ and $\omega_{j^*} = \arg\max_{\omega_j\in\Psi\setminus\Omega} \left|\langle \mathbf{a}_{\omega_j}, \tilde{\mathbf{y}}\rangle\right|^2$, then all probability components in (B.3) are lower bounded as $\mathrm{P}\left(\left|\langle \mathbf{a}_{\omega_{i^*}}, \tilde{\mathbf{y}}\rangle\right|^2 > \left|\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}}\rangle\right|^2\right)$. Hence,

$$\mathrm{P}\left(S^2|S^1\right) \geq \left[\mathrm{P}\left(\left|\langle \mathbf{a}_{\omega_{i^*}}, \tilde{\mathbf{y}}\rangle\right|^2 > \left|\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}}\rangle\right|^2\right)\right]^{(k-1)(m-k)} \tag{B.4}$$

$$= \left[\mathrm{P}\left(\left|\frac{\langle \mathbf{a}_{\omega_{i^*}}, \tilde{\mathbf{y}}\rangle}{\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}}\rangle}\right|^2 > 1\right)\right]^{(k-1)(m-k)} \tag{B.5}$$

[1] As mentioned, when $\omega^* \in \{\omega - \frac{N}{2m}, \cdots, \omega, \cdots, \omega + \frac{N}{2m}\}$ for some $\omega \in \Omega$, we can consider ω^* as ω. This is because the mobile device already knows the true support using the channel reciprocity.

where (B.4) is because $|\Omega \setminus \{\omega^*\}| = k-1$ and $|\Psi \setminus \Omega| = m-k$. One can easily show that $\left|\langle \mathbf{a}_{\omega_{i^*}}, \tilde{\mathbf{y}} \rangle\right|^2$ is a non-central Chi-squared random variable with 2 DoF and non-centrality parameter $\zeta = \beta \left|h_{\omega_{i^*}}\right|^2$, and $\left|\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}} \rangle\right|^2$ is a central Chi-squared random variable with 2 DoF. Thus, $\left|\frac{\langle \mathbf{a}_{\omega_{*_i}}, \tilde{\mathbf{y}} \rangle}{\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}} \rangle}\right|^2$ is a non-central F-distribution whose CDF is

$$\mathrm{P}\left(\left|\frac{\langle \mathbf{a}_{\omega_{i^*}}, \tilde{\mathbf{y}} \rangle}{\langle \mathbf{a}_{\omega_{j^*}}, \tilde{\mathbf{y}} \rangle}\right|^2 < x\right) = F\left(x|2,2,\zeta\right) \tag{B.6}$$

$$= \sum_{r=0}^{\infty} \left(\frac{\left(\frac{1}{2}\zeta\right)^r}{r!} e^{\left(-\frac{\zeta}{2}\right)}\right) I\left(\frac{x}{1+x}\middle|1+r,1\right), \tag{B.7}$$

where $I(x|a,b)$ is the regularized incomplete beta function with parameters a and b. From (B.5) and (B.7), we have

$$\mathrm{P}\left(S^2|S^1\right) \geq \left[1 - F\left(1|2,2,\zeta\right)\right]^{(k-1)(m-k)}, \tag{B.8}$$

which is the desired results.

Chapter C

Proof of the computational complexities in Table 6.1

In this appendix, we analyze the computational complexities of LS-BOMP and MMSE-BOMP in Table I. We first analyze the complexity of LS-BOMP. In the j-th iteration of LS-BOMP, a submatrix $\mathbf{\Phi}_l$ of $\mathbf{\Phi}$ having the maximum correlation between the residual vector $\mathbf{r}^{j-1}$ is chosen (see (6.7)). The corresponding complexity $\mathcal{C}_{\mathrm{I}}$ is

$$\mathcal{C}_{\mathrm{I}} = \sum_{j=1}^{k} \{(2m-1)mN + (mN-1)\} = 2km^2N - k. \tag{C.1}$$

After identifying a support element, a signal vector $\mathbf{x}^j$ is estimated using the LS estimator (i.e., $\mathbf{x}^j = \left(\mathbf{\Phi}_{\Omega_j}^H \mathbf{\Phi}_{\Omega_j}\right)^{-1} \mathbf{\Phi}_{\Omega_j}^H \mathbf{y}$). Using the Cholesky decomposition [90], the resulting computational complexity $\mathcal{C}_{\mathrm{LS}}$ is approximated as

$$\mathcal{C}_{\mathrm{LS}} \approx \sum_{j=1}^{k} (m + \frac{jm}{3}) j^2 m^2 \tag{C.2}$$

$$= \frac{k^4 + 6k^3 + 7k^2 + 2k}{12} m^3. \tag{C.3}$$

Finally, the residual vector $\mathbf{r}^{j-1}$ is updated as $\mathbf{r}^j = \mathbf{y} - \mathbf{\Phi}_{\Omega^j} \hat{\mathbf{x}}^j$. The corresponding complexity $\mathcal{C}_{\mathrm{U}}$ is

$$\mathcal{C}_{\mathrm{U}} = \sum_{j=1}^{k} \{(2jm-1)m + m\} = k(k+1)m^2. \tag{C.4}$$

From (C.1) to (C.4), the complexity $\mathcal{C}_{\mathrm{LS-BOMP}}$ of LS-BOMP is

$$\mathcal{C}_{\mathrm{LS-BOMP}} = \mathcal{C}_{\mathrm{I}} + \mathcal{C}_{\mathrm{LS}} + \mathcal{C}_{\mathrm{U}} \tag{C.5}$$

$$= 2km^2N - k + \frac{k^4 + 6k^3 + 7k^2 + 2k}{12}m^3 + k(k+1)m^2. \tag{C.6}$$

We next analyze the complexity of MMSE-BOMP. Since the support identification and residual update of MMSE-BOMP are the same as those of LS-BOMP, the corresponding complexities ($\mathcal{C}_{\mathrm{I}}$ and $\mathcal{C}_{\mathrm{U}}$) are also the same as LS-BOMP. When estimating the signal values, the MMSE estimator is used (i.e., $\mathbf{x}^j = \mathbf{\Phi}_{\Omega_j}^H \left(\mathbf{\Phi}_{\Omega_j}\mathbf{\Phi}_{\Omega_j}^H + \frac{\sigma_n^2}{\sigma_x^2}\mathbf{I}\right)^{-1}\mathbf{y}$). By approximating the complexity of the matrix inversion operation [91], the resulting complexity $\mathcal{C}_{\mathrm{MMSE}}$ is

$$\mathcal{C}_{\mathrm{MMSE}} \approx \sum_{j=1}^{k}\left\{2m + j\left(\frac{14}{3}m^3 + m^2 - m\right)\right\} \tag{C.7}$$

$$= 2km + \frac{k(k+1)}{2}\left(\frac{14}{3}m^3 + m^2 - m\right). \tag{C.8}$$

The resulting complexity $\mathcal{C}_{\mathrm{MMSE-BOMP}}$ of MMSE-BOMP is

$$\mathcal{C}_{\mathrm{MMSE-BOMP}} = \mathcal{C}_{\mathrm{I}} + \mathcal{C}_{\mathrm{MMSE}} + \mathcal{C}_{\mathrm{U}} \tag{C.9}$$

$$= 2km^2N - k + 2km + \frac{k(k+1)}{2}\left(\frac{14}{3}m^3 + m^2 - m\right) + k(k+1)m^2 \tag{C.10}$$

Bibliography

[1] Rec. ITU-R M.2083-0, "IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond," Sep. 2015.

[2] J. Choi, B. Shim, Y. Ding, B. Rao and D. Kim, "Compressed sensing for wireless communications: useful tips and tricks," *IEEE Commun. Surveys & Tutorials,* vol. 19, no. 3, pp. 1527-1550, Feb. 2017.

[3] Z. Gao, L. Dai, S. Han, C.-L. I, Z. Wang, and L. Hanzo, "Compressive Sensing Techniques for Next-generation Wireless Communications," *IEEE Wireless Commun.,* pp. 2-11, 2018.

[4] A. Liao, Z. Gao, H. Wang, S. Chen, M. S. Alouini, and H. Yin, "Closed-Loop Sparse Channel Estimation for Wideband Millimeter-Wave Full-Dimensional MIMO Systems," *IEEE Trans. Commun.,* vol. 67, no. 12, pp. 8329-8345, Dec. 2019.

[5] W. Kim, H. Ji, H. Lee, Y. Kim, J. Lee, and B. Shim "Sparse vector transmission: An idea whose time has come," *IEEE Veh. Techn. Mag.,* vol. 15, no. 3, pp. 32-39, Mar. 2020.

[6] T. Taleb and A. Kunz, "Machine Type Communication in 3GPP Networks: Potential, Challenges, and Solutions," *IEEE Commun. Mag.,* vol. 50, no. 3, pp. 178-184, Mar. 2012.

[7] S. Sesia, M. Baker, and I. Toufik, “LTE - the UMTS Long Term Evolution: From Theory to Practice,” *John Wiley & Sons*, 2011.

[8] 3GPP Technical Specifications 38.211, “Technical Specification Group Radio Access Network, NR (Release 15),” v15.0.0, Dec. 2017.

[9] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel Coding Rate in the Finite Blocklength Regime,” IEEE Trans. Inform. Theory, vol. 56, no. 5, pp. 2307-2359, 2010.

[10] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, “Ultra Reliable and Low Latency Communications in 5G:Physical Layer Aspects,” *IEEE Wireless Commun.,* vol. 25, no. 3, pp. 124-130, Jul. 2018.

[11] W. Kim, H. Ji, and B. Shim, “Channel aware sparse signaling for ultra low-latency communication in TDD systems,” *Proc. IEEE 88th Vehic. Tech. Conf. (VTC)*, Aug. 2018.

[12] W. Kim, H. Ji, and B. Shim, “Channel aware sparse signaling for ultra low-latency TDD access,” *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019.

[13] W. Kim, H. Ji, and B. Shim, “Channel aware sparse transmission for ultra low-latency communications in TDD systems,” *IEEE Trans. Commun.,* vol. 68, no. 2, pp. 1175-1186, Feb. 2020.

[14] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Sig. Proc. Mag.,* vol. 25, no. 2, pp. 21-30, March 2008.

[15] 3GPP Technical Report 38.913, “Study on Scenarios and Requirements for Next Generation Access Technologies (Release 14),” v14.2.0, May 2017.

[16] 3GPP Technical Report 38.802, “Study on New Radio Access Technology Physical Layer Aspects (Release 14),” v14.2.0, Sep. 2017.

[17] C.-P. Li, J. Jiang, W. Chen, T. Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017.

[18] B. Lee, S. Park, D. J. Love, H. Ji, and B. Shim, "Packet structure and receiver design for low latency wireless communications with ultra-short packets," *IEEE Trans. Commun.,* vol. 66, no.2, pp. 796-807, Sep. 2018.

[19] H. Sun, M. Wildemeersch, M. Sheng, and T. Q. Quek, "D2D Enhanced heterogeneous cellular networks with dynamic TDD," *IEEE Trans. Wireless Commun.,* vol. 14, no. 8, pp. 4204-4218, Aug. 2015.

[20] Z. Gao, L. Dai, D. Mi, Z. Wang, M. A. Imran, and M. Z. Shakir, "mmWave massive MIMO based wireless backhaul for 5G ultra-dense network," *IEEE Wireless Commun. Mag.,* vol. 22, no. 5, pp. 13-21, Oct. 2015.

[21] C. She, C. Yang, and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Mag.,* vol. 55, no. 6, pp. 72-78, Jun. 2017.

[22] H. Ji, Y. Kim, S. Choi, J. Cho, and J. Lee, "Dynamic resource adaptation in beyond LTE-A TDD heterogeneous networks," *Proc. IEEE Int. Conf. Commun. (ICC) Workshop,* pp. 133-137, Jun. 2013.

[23] 3GPP Technical Report 36.881, "Evolved Universal Terrestrial Radio Access (E-UTRA); Study on Latency Reduction Techniques for LTE (Release 13)," v0.6.0, March 2016.

[24] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Sig. Proc.,* vol. 60, no. 12, pp. 6202-6216, Dec. 2012.

[25] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Sig. Proc.,* vol. 66, no. 7, pp. 1890-1904, Apr. 2018.

[26] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proc. Asilomar Conf. Signals, Systems, and Computers,* Nov. 1993.

[27] X. Ge, "Ultra-reliable low-latency communications in autonomous vehicular networks," *IEEE Trans. Veh. Technol.,* vol. 68, no. 5, pp. 5005-5016, May 2019.

[28] W. Kim and B. Shim, "Ultra-mini slot transmission for 5G+ and 6G URLLC network," *Proc. IEEE Veh. Technol. Conf. (VTC),* Nov. 2020.

[29] W. Kim and B. Shim, "Partial sample transmission and deep neural decoding for URLLC-based V2X systems," *Proc. IEEE Int. Conf. Commun. (ICC),* Jun. 2021.

[30] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Cambridge: MIT press,* 2016.

[31] W. Kim, Y. Ahn, and B. Shim, "Deep neural network-based active user detection for grant-free NOMA systems," *IEEE Trans. Commun.,* vol. 68, no. 4, pp. 2143-2155, Apr. 2020.

[32] X. Wei, C. Hu, and L. Dai, "Deep learning for beamspace channel estimation in millimeter-wave massive MIMO systems," *IEEE Trans. Commun.,* vol. 69, no. 1, pp. 182-193, Jan. 2021.

[33] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Commun. Mag.,* vol. 54, no. 12, pp. 160-167, Dec. 2016.

[34] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2X access technologies: regulation, research, and remaining challenges," *IEEE Commun. Surveys & Tutorials,* vol. 20, no. 3, pp. 1858-1877, 2018.

[35] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Commun. Mag.,* vol. 54, no. 5, pp. 36-42, May 2016.

[36] 3GPP Technical Report 38.885, "Study on NR Vehicle-to-Everything (V2X) (Release 16)," *v16.0.0*, Mar. 2019.

[37] 3GPP Technical Report 38.912, "Study on New Radio (NR) access technology (Release 15)," *v15.0.0*, Sep. 2018.

[38] L. Liu and W. Yu, "A D2D-based protocol for ultra-reliable wireless communications for industrial automation," *IEEE Trans. Wireless Commun.,* vol. 17, no. 8, pp. 5045-5058, Aug. 2018.

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

[40] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Commun. of the ACM,* vol. 53, no. 12, pp. 93-100, Dec. 2010.

[41] S. Kwon, J. Wang, and B. Shim, "Multipath matching pursuit," *IEEE Trans. Sig. Process.,* vol. 60, no. 12, pp. 6202-6216, Dec. 2012.

[42] S. Loffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 [cs], Mar. 2015.

[43] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo "Deep learning for wireless communications: An emerging interdisciplinary paradigm," *IEEE Wireless Commun.,* vol. 27, no. 4, pp. 133-139, Aug. 2020.

[44] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 807-814, 2010.

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

[46] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[47] D. Silver et al., “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484-489, Jan. 2016.

[48] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” In *Proc. Adv. Neural Inf. Process. Syst. (NIPS),* 2012.

[49] A. Graves, A. Mohamed, G. Hinton, “Speech recognition with deep recurrent neural networks,” In *Proc. Int. Conf. Acoustics, Speech, and Signal Process. (ICASSP),* 2013.

[50] N. Samuel, T. Diskin, and A. Wissel, “Learning to detect,” *IEEE Trans. Signal. Process.*, vol. 67, no. 10, pp. 2554-2564, 2019.

[51] W. Kim, Y. Ahn, and B. Shim, “Deep neural network-based active user detection for grant-free NOMA systems,” *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2143-2155, 2020.

[52] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, “Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549-8560, 2018.

[53] X. Wei, C. Hu, and L. Dai “Deep learning for beamspace channel estimation in millimeter-wave massive MIMO systems,” *IEEE. Trans. Commun.,* vol. 69, no. 1, Jan. 2021.

[54] M. Alenezi, K. K. Chai, A. S. Alam, Y. Chen, and S. Jimaa, "Unsupervised Learning Clustering and Dynamic Transmission Scheduling for Efficient Dense LoRaWAN Networks," *IEEE Access*, 2020.

[55] J. Guo, C. K. Wen, and S. Jin, "Deep Learning-Based CSI Feedback for Beamforming in Single-and Multi-cell Massive MIMO Systems," *IEEE Journal. Sel. Areas. Commun.*, 2020.

[56] H. Ju, S. Kim, Y. Kim, H. Lee, and B. Shim, "Energy-Efficient Ultra-Dense Network via Deep Reinforcement Learning," in *Proc. IEEE Workshop Sig. Proc. Adv. Wireless Commun. (SPAWC),* 2020.

[57] 3GPP TS 36.104. "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) Radio Transmission and Reception," 3rd Generation Partnership Project; Technical Specification Group Radio Access Network.

[58] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu,D. Warde-Farley, S. Ozair, A. Courville, and Y. Ben-gio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS),* 2014.

[59] N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitiing," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[60] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531,* 2015.

[61] W. Kim, G. Lim, Y. Ahn, and B. Shim, "Active user detection of machine-type communications via dimension spreading neural network," *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019.

[62] W. Kim, Y. Ahn, and B. Shim, "Deep neural network-based active user detection for grant-free NOMA systems," *IEEE Trans. Commun.,* vol. 68, no. 4, pp. 2143-2155, Apr. 2020.

[63] I. Sutskever, O. Vinyals, and Q. VV. Le, "Sequence to sequence learning with neural networks," In *Proc. Adv. Neural Inf. Process. Syst (NIPS),* 2014.

[64] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248-1261, June 2019.

[65] M. B. Shahab, R. Abbas, M. Shir- vanimoghaddam, and S. J. Johnson, "Grant-free Non-orthogonal Multiple Access for IoT: A Survey," arXiv e-prints, p. arXiv:1910.06529, Oct. 2019.

[66] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Tech.,* vol. 67, no. 9, pp. 8440-8450, Sep. 2018.

[67] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

[68] 3GPP Technical Report 36.523, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Packet Core (EPC); User Equipment (UE) conformance specification; Part 1: Protocol conformance specification," *v13.5.0*, 2017.

[69] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59-65, 2016.

[70] T. Taleb and A. Kunz, "Machine type communications in 3GPP networks: Potential, challenges, and solutions," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 178-184, Mar 2012.

[71] 3GPP Technical Report 38.802, "Study on New Radio Access Technology Physical Layer Aspects (Release 14)," *v14.1.0,*, 2017.

[72] L. Dai, B. Wang, Y. Yuan, S. Han, C-Lin I and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74-81, Sep. 2015.

[73] D. L. Donoho, A. Maleki and A. Montanari, , "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914-18919, 2009.

[74] R. Xin, Z. Ni, L. Kuang, H. Jia, and P. Wang, "Joint Active User and Data Detection in Uplink Grant-Free NOMA by Message-Passing Algorithm," *Proc. IEEE Int. Wireless Commun. & Mobile Computing Conf. (IWCMC),* June 2019.

[75] Y. Du, B. Dong, W. Zhu, P. Gao, Z. Chen, X. Wang, and J. Fang, "Joint Channel Estimation and Multiuser Detection for Uplink Grant-free NOMA," *IEEE Wireless Commun. Letters*, vol. 7, no. 4, pp. 682-685, Feb. 2018.

[76] J. Ahn, B. Shim, and K. B. Lee, "EP-based Joint Active User Detection and Channel Estimation for Massive Machine-Type Communications," *IEEE Trans. Commun.,* vol. 67, no. 7, pp. 5178-5189, Jul. 2019.

[77] R. Hoshyar, F. P. Wathan, and R. Tafazolli, "Novel low-density signature for synchronous CDMA systems over AWGN channel," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1616-1626, Apr. 2008.

[78] R.G. Baraniuk, V. Cevher, M.F. Duarte, C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982-2001, Apr. 2010.

[79] 3GPP Technical Report 38.812, "Study on Non-Orthogonal Multiple Access (NOMA) for NR," v15.0.0, Dec. 2018.

[80] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4700-4708, 2017.

[81] S. Loffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 [cs], Mar. 2015.

[82] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770-778, 2016.

[83] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 807-814, 2010.

[84] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 231-238, 1995.

[85] 3GPP Technical Report 36.931, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) requirements for LTE Pico Node B," v9.0.0, May 2011.

[86] S. Sesia, M. Baker, and I. Toufik, "LTE-the UMTS Long Term Evolution: from Theory to Practice," John Wiley & Sons., 2012.

[87] S. Park, H. Seo, H. Ji and B. Shim, "Joint active user detection and channel estimation for massive machine-type communications," *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2017.

[88] S. Lyu, and C. Ling, (2018). "Hybrid vector perturbation precoding: The blessing of approximate message passing," *IEEE Trans. Sig. Proc.,* vol. 67, no. 1, pp. 178-193, Oct. 2018.

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

[90] S. Boyd and L. Vandenberghe, “Convex Optimization,” Cambridge University Press, 2004.

[91] R. W. Farebrother, “Linear least squares computations,” Marcel Dekkler, Inc., 1988.

초 록

차세대 5G+ 및 6G 무선통신 시스템의 새로운 기술 혁신은 무인 차량 및 항공기, 스마트 도시 및 공장, 원격 의료 진단 및 수술, 인공 지능 기반 맞춤형 지원과 같은 전례없는 서비스 및 응용프로그램으로 부상하고 있다. 이러한 새로운 애플리케이션 및 서비스와 관련된 통신 방식은 대기 시간, 에너지 효율성, 신뢰성, 유연성 및 연결 밀도 측면에서 다양한 응용 요구성능이 보장되어야 한다. 현재의 무선 액세스 방식을 비롯한 종래의 접근법은 이러한 요구 사항을 만족할 수 없기 때문에 최근 무선 통신 시스템에 새로운 신호처리 기법을 활용하는 연구가 진행되고 있다. 주목할만한 예로는 희소신호처리(sparse signal processing)를 위한 압축센싱(compressed sensing, CS) 기법을 통해 채널 추정, 간섭 제거, 각도 추정, 및 스펙트럼 감지가 있다. 현재까지 연구는 주어진 신호가 가지고 있는 본래의 희소성에 주목하였으나 본 논문에서는 기존의 접근 방법과 달리 인위적으로 설계된 희소성을 이용하여 통신 시스템의 성능을 향상시키는 방법을 제안한다. 또한, 자율주행, 자동화, 알파고 등을 통해 큰 관심을 받고 있는 딥러닝(deep learning) 기법을 활용하여 무선 통신 시스템을 설계함으로써, 기존 데이터 처리량 향상을 위해 고안된 시스템의 한계점을 극복하고, 통신 시스템의 응용 요구조건을 만족시키는 방법을 제안한다.

우선 본 논문은 저지연 통신을 위해 데이터를 희소 신호에 매핑하여 전송하는 희소 벡터 전송(sparse vector coding, SVT) 기술을 제안한다. SVT 기법은 직교 주파수 분할 다중 방식(orthogonal frequency division multiplexing, OFDM) 심볼에서 부반송파 (subcarrier)의 위치와 심볼에 데이터 정보를 매핑하여 전송하고, 수신단에서는 전송된 희소벡터의 0이 아닌 위치를 식별한 뒤, 매핑된 심볼 검출을 통해 원신호를

복원한다. 희소벡터 복원에는 전송 신호의 차원보다 적은 차원의 관측치를 활용하기 때문에, 전체 신호를 모두 관측할 필요가 없으므로 물리계층 지연시간을 대폭 줄일 수 있다. 본 논문은 SVT 기술을 다음의 두가지 영역으로 확장하였다. 첫째로, 채널 상호성(channel reciprocity)을 가지는 시분할 이중통신(time division duplex, TDD) 시스템에 SVT를 활용하여 고신뢰저지연 통신(ultra-reliable low-latency communications, URLLC)을 지원하는 방법을 제안한다. 제안하는 기법인 채널인지 희소전송(channel-aware sparse transmission, CAST)은 정보 전달을 위한 부반송파 선택에 기지국과 단말 사이의 상향 링크, 하향 링크 채널 및 이산 푸리에 변환(inverse discrete Fourier transform, IDFT) 행렬을 활용하여 페이딩 채널에 강인한 성능을 제공하고 짧은 물리계층 지연시간을 제공한다. 둘째로, SVT 기술이 적은 수의 관측치로 복호화(decoding)가 가능하다는 사실을 이용하여 차량-사물통신(vehicle-to-everything, V2X) 시스템에서의 저지연 sidelink 전송 기법을 제안한다. 특히, 제안하는 기법인 부분 샘플 전송(partial sample transmission, PST) 기법에서 심층인공신경망(deep neural network, DNN) 기반 복호화를 통해 IDFT 행렬의 높은 열 상관도에 강인한 성능을 제공한다.

마지막으로, 대규모 사물통신 환경에서 딥러닝 기술 기반 활성단말검출(active user detection, AUD) 방법을 제안한다. 제안하는 기법인 심층인공신경망 기반 활성단말검출(deep neural network-based AUD, D-AUD) 기법은 훈련과정에서 충분한 양의 수신 신호와 활성 단말 인덱스를 데이터로 활용하여, 두 데이터 간 비선형적 매핑 함수를 학습한다. 단말할당 코드를 사전에 인지하고 있지 않은 기존의 방식과 달리, 제안된 기술에서는 코드 간의 상관도, 단말 수 등의 유용한 특징(feature)을 훈련과정을 통해 학습하기 때문에 검출 성능을 상당히 증가한다. 이를 통해 고과부화된(highly-overloaded) 환경 및 활성단말 밀도가 높은 상황에서도 정확한 활성단말검출이 가능하다.

주요어: 희소인지, 압축센싱, 딥러닝, 희소신호전송, 채널인지, 활성단말검출
학번: 2016-20876

ACKNOWLEGEMENT

박사 학위를 시작할 무렵, 노트 첫 장에 '이 과정이 정말 고마워졌으면 좋겠다' 라고 썼던 기억이 납니다. 마치고 보니, 많은 분들의 도움 없이는 쉬이 이룰 수 없는 과정이었기에 본 글을 통해 감사의 마음을 전하고자 합니다.

우선 박사 학위동안 열성적으로 지도해주신 심병효 교수님께 감사의 말씀을 전합니다. 항상 연구에 집중할 수 있는 환경을 물심양면으로 만들어 주시고, 밤낮으로 부족한 제자의 결과물을 갈고 닦아 주신 점은 평생 잊지 않겠습니다. 학위 논문 심사 과정에서 위원장을 맡아 주시고, 심사해주신 최완 교수님께 감사드립니다. 아울러 바쁘신 와중에도 심사위원을 맡아 주시고 소중한 조언을 해주신 이경한 교수님, 연세대학교 김성륜 교수님, 한양대학교 최준원 교수님께도 깊은 감사를 드립니다.

대학원 생활 동안 함께 지낸 소중한 연구실 친구들에게도 감사드립니다. 특별히, 초창기 서툴렀던 연구에 대해 곁에서 조언해 주신 지형주 선배님께 감사의 말씀을 전합니다. 아울러 301동 및 뉴미디어 연구소에서 오랜 기간 함께 해준 상태, 진홍, 구영, 준한, 승년, 용준, 현규, 루옹, 쟈오에게도 항상 고마웠습니다. 또한 짧은 시간이었지만 같이 생활했던 지훈, 지섭, 선우, 현수, 코아, 윤성, 지아지에, 동훈, 정재, 안호에게도 앞으로 좋은 일만 가득한 박사 과정이 되기를 바랍니다.

더불어, 행복한 시간을 함께 보냈던 소중한 친구 우준, 건모, 동윤, 경원이에게도 고맙다는 말을 전합니다. 특별히, 항상 곁에서 지지해주고 함께 해준 여자친구 이슬에게 고마움과 사랑을 전합니다.

마지막으로 사랑하는 가족들에게 깊은 감사를 전합니다. 항상 묵묵히 아들의 과정을 지켜봐주시고 믿어주신 아버지 김영규, 곁에서 항상 잘하고 있다고 격려해 주셨던 어머니 장만수, 그리고 자신의 일처럼 응원해 준 수진이에게 말로 다 표현할 수 없는 감사함과 고마움을 전합니다.