FEDERATED LEARNING AND DISTRIBUTED INFERENCE OVER WIRELESS CHANNELS

A DISSERTATION SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Büşra Tegin

November 2023

Federated Learning and Distributed Inference over Wireless Channels
By Büşra Tegin
November 2023

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Tolga Mete Duman (Advisor)

Orhan Arıkan

Ayşe Melda Yüksel Turgut

Sinan Gezici

Stefano Rini

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

FEDERATED LEARNING AND DISTRIBUTED INFERENCE OVER WIRELESS CHANNELS

Büşra Tegin
Ph.D. in Electrical and Electronics Engineering
Advisor: Tolga Mete Duman
November 2023

In an era marked by massive connectivity and a growing number of connected devices, we have gained unprecedented access to a wealth of information, enhancing the reliability and precision of intelligent systems and enabling the development of learning algorithms that are more capable than ever. However, this proliferation of data also introduces new challenges for centralized learning algorithms for the training and inference processes of these intelligent systems due to increased traffic loads and the necessity of substantial computational resources. Consequently, the introduction of federated learning (FL) and distributed inference systems has become essential. Both FL and distributed inference necessitate communication within the network, specifically, the transmission of model updates and intermediate features. This has led to a significant emphasis on their utilization over wireless channels, underscoring the pivotal role of wireless communications in this context.

In pursuit of a practical implementation of federated learning over wireless fading channels, we direct our focus towards cost-effective solutions, accounting for hardware-induced distortions. We consider a blind transmitter scenario, wherein distributed workers operate without access to channel state information (CSI). Meanwhile, the parameter server (PS) employs multiple antennas to align received signals. To mitigate the increased power consumption and hardware cost, we leverage complex-valued, low-resolution digital-to-analog converters (DACs) at the transmitter and analog-to-digital converters (ADCs) at the PS. Through a combination of theoretical analysis and numerical demonstrations, we establish that federated learning systems can effectively operate over fading channels, even in the presence of low-resolution ADCs and DACs. As another aspect of practical implementation, we investigate federated learning with over-the-air aggregation over time-varying wireless channels. In this scenario, workers transmit their local gradients over channels that undergo time variations, stemming from factors such as worker or PS mobility and other transmission medium fluctuations. These channel variations introduce inter-carrier interference (ICI), which can notably degrade the system performance, particularly in cases of rapidly varying channels. We examine the effects of the channel time variations on FL with over-the-air aggregation, and show that the resulting undesired interference terms have only limited destructive effects, which do not prevent the convergence of the distributed learning algorithm.

Focusing on the distributed inference concept, we also consider a multi-sensor wireless inference system. In this configuration, several sensors with constrained computational capacities observe common phenomena and engage in collaborative inference efforts alongside a central device. Given the inherent limitations on the computational capabilities of the sensors, the features extracted from the front part of the network are transmitted to an edge device, which necessitates sensor fusion for the intermediate features. We propose L_p-norm inspired and LogSumExp approximations for the maximum operation as a sensor fusion method, resulting in the acquisition of transformation-invariant features that also enable bandwidth-efficient feature transmission. As a further enhancement of the proposed method, we introduce a learnable sensor fusion technique inspired by the L_p-norm. This technique incorporates a trainable parameter, providing the flexibility to customize the sensor fusion according to the unique network and sensor distribution characteristics. We show that by encompassing a spectrum of behaviors, this approach enhances the adaptability of the system and contributes to its overall performance improvement.

Keywords: Wireless communication, federated learning, over-the-air transmission, convergence, quantization, fading channels, time-varying channels, wireless inference, multi-sensor networks, sensor fusion.

ÖZET

KABLOSUZ KANALLAR ÜZERİNDE FEDERE ÖĞRENME VE DAĞITIK ÇIKARIM

Büşra Tegin
Elektrik ve Elektronik Mühendisliği, Doktora
Tez Danışmanı: Tolga Mete Duman
Kasım 2023

Geniş bağlantı ve artan bağlı cihaz sayısıyla işaretlenen bir dönemde, bilgi zenginliğine benzeri görülmemiş bir erişim elde ettik. Bu sayede, akıllı sistemlerin güvenilirliğini ve hassasiyeti arttı ve daha önce hiç olmadığı kadar yetenekli öğrenme algoritmaları geliştirildi. Ancak, bu veri miktarındaki artış aynı zamanda artan trafik yükleri ve önemli hesaplama kaynaklarının gerekliliği nedeniyle bu akıllı sistemlerin eğitim ve sonuç çıkarma süreçlerini geliştirmede yeni zorlukları da beraberinde getirmektedir. Sonuç olarak, federe öğrenme (FL) ve dağıtık çıkarım sistemlerinin tanıtımı kaçınılmaz hale gelmiştir. Hem FL hem de dağıtık çıkarım, ağ içinde iletişimi gerektirir, özellikle model güncellemelerinin ve ara özelliklerin iletimini gerektirmektedir. Dolayısıyla, bunların kablosuz ağlarda kullanımına önemli bir odak noktası oluşturmuş ve bu bağlamda kablosuz iletişimin kilit rolünü vurgulamıştır.

Kablosuz solma kanallarda federe öğrenmenin pratik uygulanmasına yönelik olarak, donanım kaynaklı bozulmaları hesaba katan maliyet-etkin çözümlere odaklanıyoruz. Dağıtılmış çalışanların kanal durumu bilgisine (CSI) erişim olmadan çalıştığı bir kör verici senaryosunu dikkate alıyoruz. Aynı zamanda, parametre sunucusu (PS), alınan sinyalleri hizalamak için birden fazla anten kullanmaktadır. Güç tüketimini azaltmak ve donanım masraflarını düşürmek için, verici tarafında karmaşık değerli, düşük çözünürlüklü dijital-analog dönüştürücüler (DAC) ve PS tarafında analog-dijital dönüştürücüler (ADC) kullanıyoruz. Teorik analiz ve sayısal gösterimlerin bir kombinasyonuyla, federe öğrenme sistemlerinin az çözünürlüklü ADC ve DAC varlığında bile azalan kanallarda etkili bir şekilde çalışabileceğini belirliyoruz. Pratik uygulamanın başka bir yönü olarak, zaman içinde değişen kablosuz kanallar üzerinde havadan birleştirme ile federe öğrenmeyi araştırıyoruz. Bu senaryoda, işçiler yerel gradyanlarını, işçi veya PS hareketliliği gibi faktörlerden kaynaklanan zaman değişimleri yaşayan kanallar üzerinden iletiyorlar. Bu kanal değişimleri, özellikle hızlı değişen kanalların varlığında sistem performansını kayda değer bir şekilde düşürebilen ara taşıyıcı interferans (ICI) girişi oluşturur. Kanal zaman değişimlerinin FL ile havadan birleştirme yaklaşımı üzerindeki etkilerini inceliyoruz ve ortaya çıkan istenmeyen interferans terimlerinin, dağıtık öğrenme algoritmasının yakınsamasını engellemeyen sınırlı yıkıcı etkilere sahip olduğunu gösteriyoruz.

Dağıtık çıkarım konseptine odaklandığımızda, çoklu sensör kablosuz çıkarım sistemini ele alıyoruz. Bu yapıda, sınırlı hesaplama kapasitesine sahip birkaç sensör, örtüşen bölgeleri gözlemliyor ve merkezi bir cihaz ile birlikte işbirlikçi çıkarım çabalarına katılıyor. Sensörlerin hesaplama yeteneklerinin doğal sınırlamaları göz önüne alındığında, ağın ön bölümünden çıkarılan özellikler, ara özellikler için sensör birleştirmeyi gerektiren bir kenar cihazına iletilir. Maximum işlemi için L_p-norm ilham alınan ve LogSumExp yaklaşımları ile sensör birleştirme yöntemi olarak öneriyoruz, bu da dönüşümle değişmez özelliklerin elde edilmesine ve aynı zamanda bant genişliğini verimli bir şekilde kullanılmasına olanak tanır. Önerdiğimiz yöntemin bir gelişimi olarak, L_p-norm ilham alınan bir öğrenilebilir sensör birleştirme tekniği sunuyoruz. Bu teknik, sensör birleştirme işlevine öğrenilebilir bir parametre ekler ve sensör birleştirmeyi ağın benzersiz özellikleri ve sensör dağılımı karakteristiklerine göre özelleştirmenin esnekliğini sunar. Davranışların bir yelpazesini kapsayarak, bu yaklaşım sistemin uyum sağlama yeteneğini artırır ve genel performans iyileştirmesine katkıda bulunur.

Anahtar sözcükler: Kablosuz iletişim, federe öğrenme, havadan iletim, yakınsama, kuantizasyon, sönme kanalları, zamanla değişen kanallar, kablosuz çıkarım, çoklu sensör ağları, sensör birleştirme.

Acknowledgement

I want to express my deepest gratitude to my advisor, Prof. Tolga M. Duman. Without his unwavering guidance and support, this journey would have been an insurmountable challenge. His immense knowledge, dedicated help, and infinite patience have made the impossible not only possible but also a rewarding experience.

I am also immensely thankful to my esteemed examiners: Prof. Orhan Arıkan, Prof. Ayşe Melda Yüksel Turgut, Prof. Sinan Gezici, and Prof. Stefano Rini for their perceptive comments and feedback that have significantly improved the quality of my work.

The success of my research would not have been possible without the collaborative and inspiring atmosphere of the Bilkent Communication Theory and Application Research (CTAR) Lab. I extend my sincere appreciation to all the lab members, Mert Özateş, Mohammad Javad Ahmadi, Muhammad Atif Ali, Furkan Bağcı, E. Uras Kargı, Dr. Mücahit Gümüş, Dr. Mohammad Kazemi, Dr. Reza Asvadi, and Dr. Ruslan Morozov. I also would like to acknowledge Eduin Hernandez as my valued research collaborator. Furthermore, I wish to thank my colleagues at Huawei for their support and enlightening discussions, particularly Dr. Mert Kalfa, Sadık Yağız Yetim, Mehmetcan Gök, and Arda Atalık.

I cannot express but only try how much I am thankful to my family for their endless and unconditional support throughout this journey.

Additionally, I want to extend my heartfelt thanks to Büşra Çankaya Akoğlu and Beyza Yazıcı, my closest friends, who were a constant source of laughter and strength during the highs and lows of this process.

This journey would not have been possible without the collective support and encouragement of all these individuals. I am truly grateful for their contributions to my academic and personal growth.

Contents

List of Figures

List of Tables

List of Acronyms

A-DSGD	Analog distributed stochastic gradient descent
ADC	Analog-to-digital converter
AI	Artificial intelligence
AQNM	Additive quantization noise model
AWGN	Additive white Gaussian noise
CARLA	Car Learning to Act
CLT	Central limit theorem
CNN	Convolutional neural network
CP	Cyclic prefix
CS	Compressed sensing
CSI	Channel state information
CSIR	CSI at the receiver
D-DSGD	Digital distributed stochastic gradient descent
DAC	Digital-to-analog converter
DeepJSCC	Deep joint source-channel coding
DFT	Discrete Fourier Transform
DNN	Deep neural network
FL	Federated learning
FLOP	Floating-point operation
GNN	Graph neural network
HOTAFL	Hierarchical over-the-air federated learning
IB	Information bottleneck
ICI	Inter-carrier interference
IDFT	Inverse discrete Fourier Transform
IoT	Internet of Things

i.i.d.	Independent and identically distributed
LFL	Lossy federated learning
LSE	LogSumExp
MAC	Multiple access channel
MIMO	Multiple-input multiple-output
ML	Machine learning
MVCNN	Multi-view convolutional neural network
NN	Neural network
non-i.i.d.	Non-independent and identically distributed
OFDM	Orthogonal frequency-division multiplexing
OTA	Over-the-air
PS	Parameter server
QNN	Quantized neural network
ReLU	Rectified linear unit
SGD	Stochastic gradient descent
SNR	Signal-to-noise ratio
SQNR	Signal to quantization noise ratio
TI	Transformation-invariant
WSS	Wide-sense stationary

Chapter 1

Introduction

Artificial intelligence (AI) and machine learning (ML) systems have been a significant focus of research in both academia and industry for many decades. Despite the introduction of the first trainable network, *The Perceptron* [4], in the 1950s, the development of neural networks was gradual and hindered by constraints such as insufficient data and computational power. Nevertheless, these early developments in neural networks laid the foundation for deep learning, a representation-learning method designed to learn complex functions [5], with the aid of the introduction of larger datasets and increased computational power. With these advancements, the modern industry has started to direct its attention towards deep learning which is used in various areas such as maintenance, manufacturing, or automatic systems for practical applications. Due to this interest, the demand for accurate models has increased, and deep neural networks have become an essential part of many applications due to their unprecedented success in capturing the nonlinear input-output mapping and learning data patterns [6].

Training precise algorithms for deep learning requires a substantial amount of training data and computing power. However, the surge in data traffic, resulting from increased dataset sizes and the excessive computational load of the training phase, has made centralized training unfeasible. To address these challenges and enable effective network training, an alternative to centralized processing called federated learning (FL) has gained significant attention in recent years, which decentralizes the training of the network by distributing the computations. In FL, each connected device computes the necessary local updates based on its local dataset and subsequently transmits these local updates to a central processor without any raw data transmission. The central processor then aggregates the updates by averaging and adjusts the global parameters. With the introduction of this concept, recent literature has delved into various issues associated with federated learning, e.g., the impact of energy constraints, resource allocation, privacy concerns, compression techniques for local computations, and performance across diverse channel models. Furthermore, learning-related challenges are also explored, such as non-independent and identically distributed (non-i.i.d) data distributions, network optimization, and the convergence analysis of learning algorithms.

With the increasing interest in FL, its efficient implementation has also attracted attention. Despite the absence of explicit raw data transmission, there is still a communication overhead in FL due to local update transmission. Therefore, as an efficient solution to address this concern, over-the-air transmission has been introduced. With this approach, all participating users transmit their local updates simultaneously over a multiple access channel (MAC). The central processor directly receives the superposition of these transmitted signals by exploiting the waveform superposition property of the MAC. It is important to emphasize that this process does not require raw data sharing, and therefore, it inherently provides higher privacy compared to centralized training.

As previously mentioned, federated learning primarily focuses on the distributed training phase to benefit from local datasets without the need for data transmission from participating users. While the training phase is typically seen as the most demanding computational bottleneck for an intelligent system, equal attention must also be given to the real-time inference phase, where DNNs are utilized due to their superior performance. However, a typical network model for a DNN can comprise tens of millions of parameters, resulting in terabyte-scale floating-point operations per second [7]. Therefore, it may be challenging to perform the computational tasks required for the inference phase of an intelligent system on resource-constrained devices commonly used today.

To address the excessive resource requirements of DNNs, several solutions have been proposed in the literature. Traditionally, one can transmit the sensed raw data to the cloud or another computationally powerful device, which then performs the DNN inference. However, as a drawback, this solution increases the amount of data traffic and significantly raises the latency, which is undesirable for real-time applications. As a low-latency and energy-efficient DNN inference solution, one can partition the network into two components: the front-end at the sensor side and the back-end at the cloud or helper device. Initially, sensor data is collected and processed at the front-end to generate an intermediate feature, which is then transmitted to a helper processing device for completing the inference task. Note that intermediate feature transmission is mostly performed over wireless channels, which introduces impairments on the transmitted signal for a distributed inference system. This setup is referred to as *wireless inference*.

With the development of multiple types of sensors and the increasing availability of cheap devices, the amount of available data has significantly increased, leading to the growing importance of wireless inference. This data amplification generated by multi-sensor networks creates a need for data fusion from different sources to extract and utilize the most relevant and useful information. Drawing inspiration from these advancements, one can enhance the wireless inference performance by deploying multiple sensors to gather data about a common phenomenon, exploiting data augmentation to produce reliable inference results. In the context of multi-sensor wireless inference setups, similar to the single-sensor wireless inference, each sensor performs the front-end network operations independently, and the resulting intermediate features are transmitted to the central device. Inherently, this process requires sensor fusion to produce combined intermediate features from multiple sensors, which are then further processed to obtain an inference result at the receiver side. It is important to note that this sensor fusion step significantly influences not only the accuracy of the inference process but also the associated transmission and computational costs, paving the way for new research areas.

The success of intelligent systems, the growing interest in their deployment across various applications, and the surge in their usage have led to the emergence of numerous research areas at the intersection of wireless communications and machine learning. Specifically, federated learning over wireless channels and inference represent active areas of research. This thesis systematically addresses their practical implementation in real-world scenarios, exploring the detailed impact of hardware impairments, complexities of wireless channels, and the pursuit of transmission-efficient solutions.

1.1 Contributions

We explore a cost-effective implementation of federated learning using low-resolution digital-to-analog converters (DACs) at the user side and analog-to-digital converters (ADCs) at the parameter server. This approach significantly reduces the implementation cost and overall energy consumption of the system while also employing over-the-air local update transmission for communication efficiency. We demonstrate that the learning performance is only slightly degraded despite the use of low-resolution DACs and ADCs. Next, we extend our study to over-the-air federated learning over time-varying channels to model the channel variations encountered in realistic environments. These time variations result in distortion in the model updates, which can have a detrimental effect on learning performance. However, through convergence rate analysis and numerical examples, we illustrate that federated learning over time-varying channels can effectively provide sufficient accuracy.

With an emphasis on the inference phase of learning approaches, we investigate multi-sensor wireless networks for data sensing and front-end feature processing. In this configuration, data fusion is inherently performed in an over-the-air manner, reducing data traffic and yielding transformation-invariant features by employing approximations for sensor fusion using LogSumExp and L_p-norm inspired functions for maximum operation. To expand our research, we further introduce an L_p-norm inspired flexible sensor fusion method, incorporating a trainable parameter to create an adaptable network while the transmission cost is significantly reduced with the help of over-the-air transmission.

The key contributions can be summarized as follows:

Blind Federated Learning at the Wireless Edge with Low-Resolution ADC and DAC: We explore application of FL in realistic wireless environments, addressing practical implementation challenges and wireless channel effects. We model the communication link as a frequency-selective fading channel and utilize orthogonal frequency division multiplexing (OFDM) for transmitting local gradients. In our setup, we assume that there is no transmitter-side channel state information (CSI), hence multiple antennas are employed at the receiver side to align the received signals. Additionally, to reduce hardware complexity and power consumption, we implement low-resolution DACs at the transmitter side and ADCs at the parameter server (PS) to study the effects of practical low-cost DACs and ADCs on the learning performance. Our theoretical analysis shows that the impairments caused by low-resolution DACs and ADCs, including those of one-bit DACs and ADCs, do not prevent the convergence of the federated learning algorithms, and the multipath channel effects vanish when a sufficient number of antennas are used at the PS.

Our research on this topic has been published in [8, 9].

Federated Learning with Over-the-Air Aggregation over Time-Varying Channels: We consider over-the-air aggregation empowered federated learning over time-varying wireless channels. Workers independently compute their local gradients based on their respective datasets and transmit them to a PS via a time-varying multipath fading multiple access channel utilizing OFDM. These wireless channel variations introduce inter-carrier interference (ICI), particularly in rapidly changing channels, posing a challenge to OFDM systems. We explore the impact of channel variations on FL convergence with over-the-air aggregation and demonstrate that these interference effects do not hinder the convergence of the learning algorithm, especially under slow to moderate variations. Furthermore, we validate our results through extensive simulations, which align with the theoretical expectations.

Our results for this line of investigations have been published in [10, 11].

Transformation-Invariant Over-the-Air Combining for Multi-Sensor Wireless Inference: In this line of work, we propose a multi-sensor wireless inference system where an edge device combines features sensed by different sensors. Due to the limited computational capabilities of sensors, the features obtained through the front part of the network are transmitted to the edge device, which uses L_p-norm inspired and LogSumExp (LSE) approximations for the maximum operation to obtain transformation-invariant features. These features can be transmitted in an over-the-air manner, ensuring bandwidth-efficient transmission. We also consider multi-modal network branches for sensors based on their computational capabilities, improving the overall performance by using data obtained from both computationally limited and powerful devices enhancing the usefulness of the overall sensed data.

Our results on over-the-air combining for multi-sensor wireless inference have been accepted for presentation in [12].

Learnable Sensor Fusion for Multi-Sensor Wireless Inference Networks: In the final part of the thesis, we explore the concept of learnable sensor fusion in the context of a multi-sensor wireless inference system. Given the limited computational capacity of individual sensors, in our setup, the sensors exclusively employ front-end networks to extract intermediate features. These features are subsequently transmitted to a central processing device through a multiple access channel to facilitate the inference process. The use of multiple sensors inherently entails the collection of more data, and introduces the necessity for sensor fusion. Specifically, we introduce an over-the-air learnable sensor fusion method inspired by the L_p-norm through a trainable parameter in the sensor fusion function. The proposed approach allows for the customization of sensor fusion to match the specific characteristics of the network and sensor distribution by capturing a range of behaviors, and it enhances the adaptability of the system and its overall performance.

1.2 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we provide the fundamentals of federated learning and wireless inference, along with an overview of the existing literature on these topics. Chapter 3 focuses on blind federated learning at the wireless edge with low-resolution ADC and DAC. Our study on FL with over-the-air aggregation over time-varying channels is presented in Chapter 4. In Chapter 5, we discuss transformation-invariant over-the-air combining for multi-sensor wireless inference, addressing transmission-efficient sensor fusion. Chapter 6 is dedicated to learnable sensor fusion for multi-sensor wireless inference networks, which provides flexibility in the sensor fusion function. Our conclusions and future research directions are presented in Chapter 7.

Chapter 2

Preliminaries and Literature Review

2.1 Neural Networks and Deep Learning

Neural networks (NNs) are tools that are capable of emulating the cognitive understanding of the human brain. The origins of NN studies trace back to 1943 when McCulloch and Pitts analytically modeled the biological workings of brain neurons to imitate logical functions [13]. Inspired by this initial work, Rosenblatt introduced the Perceptron featuring a single layer of neurons capable of classifying images with just a few hundred pixels [4]. The Perceptron is often considered as the ancestor of neural networks.

In recent years, neural networks have become widely used to simulate the adaptable nature of the human brain to respond effectively to changing inputs and give optimal results. These neural networks typically consist of three layers: the input layer, the hidden layer, and the output layer, with each layer composed of multiple nodes known as neurons, as illustrated in Fig. 2.1. The input layer receives and forwards the input data to the next layer. The hidden layer, which is between the input and output layers, carries out operations to uncover hidden

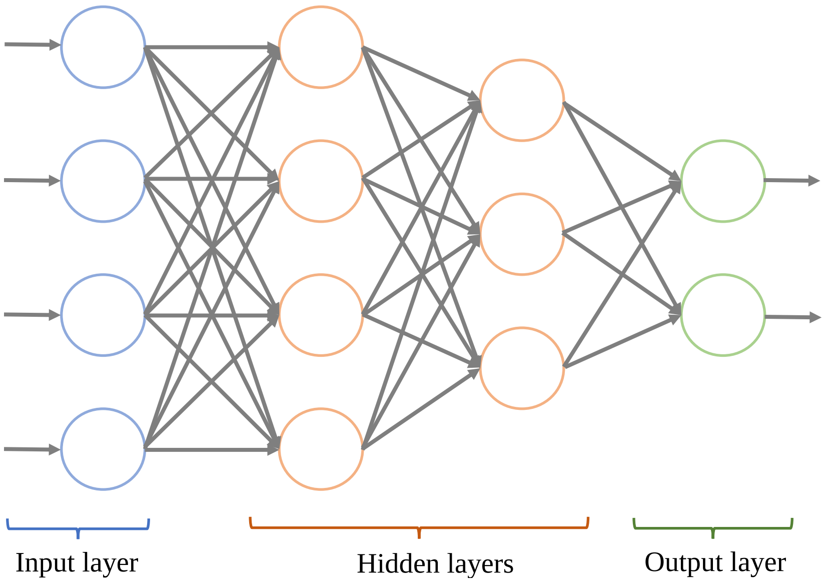


Figure 2.1: A simple neural network representation.

features and patterns of the input. Once the input has been processed through the hidden layers, the final result is obtained from the output layer.

Each neuron in these layers (represented by nodes in Fig. 2.1) receives inputs from connected layers, which are then multiplied by corresponding weights. The resulting weighted inputs are combined with a bias term, and this summation is passed through a nonlinear activation function to generate the neuron output. These operations are depicted in Fig. 2.2 and commonly referred as forward pass of a network. It is important to note that during the training process, the weight and bias terms are adjusted and optimized to learn the input-output relationship.

After the forward pass, the predictions are compared to the ground truth labels in the dataset using a loss function. For machine learning approaches, the empirical loss function, denoted by $F(\mathbf{w})$ during iteration t, can be written as

$$F(\mathbf{w}_t) = \frac{1}{B} \sum_{d \in \mathcal{B}} f(\mathbf{w}_t, d), \tag{2.1}$$

where $\mathbf{w_t}$ is the model parameters to be optimized, $\mathcal{B}$ is the dataset with size B and $f(\cdot)$ is the loss function. Most commonly used loss functions for neural

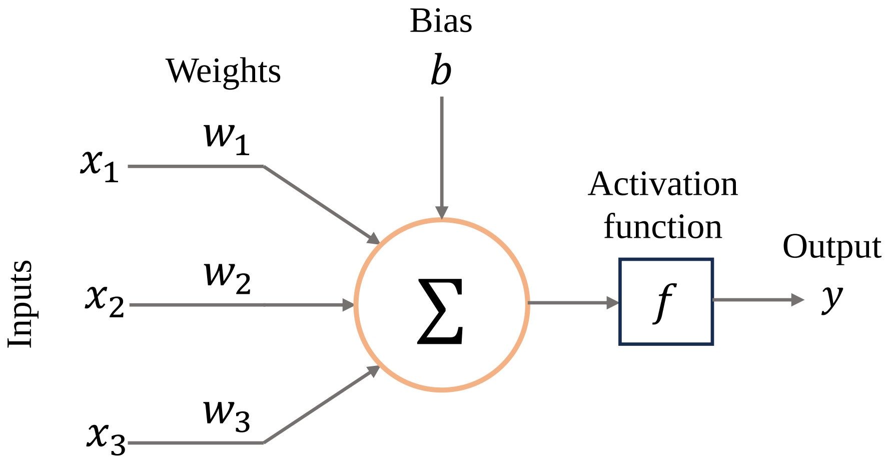


Figure 2.2: A simple neural network representation.

networks include cross-entropy and mean squared error.

To minimize the loss function, the gradient descent can be used and the network parameters can be optimized through the update rule, which is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t) = \mathbf{w}_t - \eta \frac{1}{B} \sum_{d \in \mathcal{B}} \nabla f(\mathbf{w}_t, d), \tag{2.2}$$

during iteration t where η is the learning rate. The gradient calculation and update process provide insight into how much each network parameter should be adjusted to reduce the current loss.

Iteratively, forward pass and backpropagation are performed until a predetermined number of iterations or the desired level of accuracy is achieved. This process results in the final network weights, which are utilized in machine learning applications for prediction including shallow and deep neural networks.

Neural networks are effective tools in capturing the nonlinear structure of input data through hidden layers, making them a fundamental component of intelligent systems. However, the issue of vanishing gradients during backpropagation prevents the effective training of lower layers, often leading to convergence at local minima or saddle points, as discussed in [14, 15]. Despite various proposed solutions to mitigate these challenges, DNNs have gained attention due to their remarkable performance in numerous applications compared to NNs. Their multiple hidden layers enable the extraction of more information related to input-output relationships, significantly enhancing performance [16]. Moreover, their ability to generalize data reduces the occurrence of overfitting problems when employing them [17].

Regardless of the remarkable success of DNNs, which makes them an integral part of a wide range of applications, their multilayered structure introduces challenges related to their complexity. For instance, to perform a forward pass of the ResNet-152 model [18], consisting of 152 layers, 11×10^9 floating-point operations (FLOPs) are required for a 224×224 input image in a single iteration. This may not be feasible on simple devices, as discussed in [19], and the use of distributed learning approaches has gained significant attention over the years. As shown with the update rule given in (2.2), in gradient descent, all the samples in the training set are used during each iteration to update the model parameters, which can be computationally demanding for deeper networks. Alternatively, it is possible to use a random subset of training samples as a training set during a particular iteration for the update, which is called stochastic gradient descent. This approach allows for the parallelization of the learning process by distributing the dataset and computation across several participants.

With the distributed SGD, the model update rule during iteration t can be written as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{1}{M} \sum_{m=1}^{M} g_m(\mathbf{w}_t), \tag{2.3}$$

where M is the number of participants, and

$$g_m(\mathbf{w}_t) = \frac{1}{B_m} \sum_{d \in \mathcal{B}_m} \nabla f(\mathbf{w}_t, d), \tag{2.4}$$

is the stochastic gradient for the m-th participant for its local dataset $\mathcal{B}_m$ with size B_m. This approach enables efficient training for DNNs by accelerating the training phase and distributing the computational loads among multiple participants. Hence, more complex and accurate machine learning algorithms can be trained in a collaborative way.

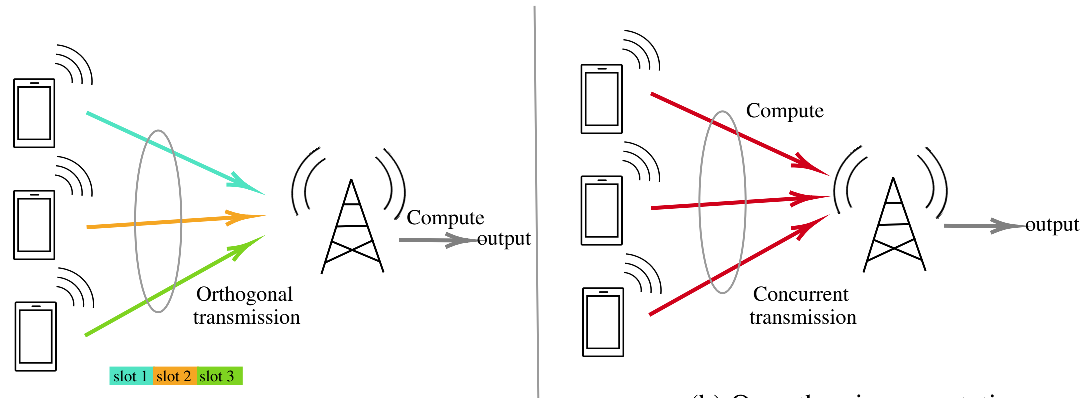
2.2 Over-the-Air Computation

The rapid advancements in communication, data collection methods, and device technologies have given rise to the emergence of a new phenomenon: big data. While big data brings with it the potential for robust optimization and learning techniques, it also presents challenges in the form of substantial computational loads, high bandwidth demands, and data traffic congestion. To address these challenges, the concept of over-the-air (OTA) computation can be considered as an efficient approach to data fusion. Over-the-air computing allows us to obtain the average of transmitted signals directly without allocating different resources for the transmitters. This approach can be applied in various domains, including deep learning training using SGD, as discussed in the previous section with the update rule provided in (2.3) and (2.4), which utilizes the average of the local gradients from the participants during the model update.

Consider a system with multiple signal sources aiming to transmit their signals to a common receiver. The goal of the receiver is to calculate a superposition signal from all the transmitters. Traditionally, the simplest solution is the *transmit-then-compute* method, in which signal sources transmit their signals during a resource slot reserved only for them. The receiver then combines these signals to obtain the superposition signal after recovering each transmitted ones.

In contrast, an alternative approach, where error-free transmission of data from M sources over a multiple access channel simultaneously is achieved with the superposition property, can be explored. Let the communication model be given as

$$y = \sum_{m=1}^{M} \psi_m(u_m), \tag{2.5}$$



(a) Transmit-then-compute

(b) Over-the-air computation

Figure 2.3: Traditional method vs over-the-air computing [1].

where u_m is the transmitted signal by the m-th data source, ψ_m pre-processing function of the m-the data source and y is the received signal. As discussed in [20, 21], every real-valued multivariate function is representable in its nomographic form as a function of a finite sum of univariate functions, there always exists a set of pre-processing functions ψ_m and a post-processing function φ such that

$$r = \phi(u_1, \cdots, u_M) = \varphi\left(\sum_{m=1}^{M} \psi_m(u_m)\right). \tag{2.6}$$

As indicated in (2.6), each data source m simultaneously transmits their pre-processed data over the multiple access channel, which is then received and processed by the receiver using the function $\varphi(\cdot)$ to extract the desired signal. This approach streamlines the communication process, allowing various operations such as summation and averaging to be performed within a single time slot, as illustrated in Fig. 2.3.

Over-the-air computation finds application in various scenarios. It is particularly valuable in cases where consensus is required. For instance, in a network featuring multiple robots, each robot can transmit their sensed data to a central node concurrently, and the central node can take action based on the average of the data from all robots. More recently, over-the-air computation has gained prominence in the domains of distributed machine learning and federated learning, where the communication of local gradients is efficiently executed through over-the-air methods.

Sufficient Statistics and Over-the-Air Computation

Statistics are functions of data $\mathbf{X}$ denoted by $T(\mathbf{X})$. For any statistics with a parameter θ describing the underlying data distribution, we observe the following relationship, expressed as a Markov chain,

$$\theta \rightarrow \mathbf{X} \rightarrow T(\mathbf{X}). \tag{2.7}$$

According to the data processing inequality, (2.7) implies that

$$I(\theta, T(\mathbf{X})) \leq I(\theta, \mathbf{X}), \tag{2.8}$$

where $I(\cdot, \cdot)$ represents the mutual information function. Consider statistics satisfying the following Markov chain, in addition to (2.7):

$$\theta \rightarrow T(\mathbf{X}) \rightarrow \mathbf{X}, \tag{2.9}$$

which means that knowing $T(\mathbf{X})$ removes the randomness in $\mathbf{X}$ related to the parameter θ. These are referred to as *sufficient statistics*, resulting in

$$I(\theta, T(\mathbf{X})) = I(\theta, \mathbf{X}). \tag{2.10}$$

If $T(\mathbf{X})$ is a sufficient statistic, it inherently contains all the information in $\mathbf{X}$ necessary to estimate the unknown parameter θ. Sufficient statistics can be identified using various methods, such as the Fisher-Neyman Factorization Theorem [22].

Connecting (2.6) and (2.5) with sufficient statistics, one can infer that over-the-air computation can be viewed as parameter estimation for samples $\{u_1, u_2, \cdots, u_M\}$ using $\psi_m(\cdot)$ and $\varphi(\cdot)$ functions (or equivalently function $\phi(\cdot)$) as a sufficient statistic $T(u_1, u_2, \ldots, u_M)$ for the samples. For instance, consider a random sample $u_1, u_2, \ldots, u_M$ from a normal distribution with an unknown mean μ and known variance σ^2. Using the Factorization theorem, one can demonstrate that $\frac{u_1+u_2+\cdots+u_M}{M}$ serves as a sufficient statistic for estimating the mean μ. Consequently, one may infer that if the underlying statistical model for the transmitted signals from M different transmitters is known, with an unknown parameter to estimate, the situation bears a resemblance to the over-the-air approach.

2.3 Federated Learning

With the emergence of new technologies, both the quality and quantity of data collection devices have increased, enabling the collection of an unprecedented amount of data for intelligent machine learning algorithms. The increase in the amount of data results in higher classification accuracy of the ML algorithms; however, training with a single machine requires an excessive amount of computations, memory requirements, and greatly increased training time and energy consumption [23]. In addition, it also raises privacy concerns [24]. To address these challenges, a novel decentralized learning approach known as *federated learning* is proposed in [25]. This method facilitates distributed learning, ensuring that the local data remains on the respective data owner's device while the aggregated model is obtained by incorporating the local updates from all participating devices. Unlike traditional distributed learning methods, participating users do not share their local data directly with each other. Instead, the local model updates are shared with a central device, typically a central server or aggregator. With this approach, the need for sharing raw data with other devices (other participants and the central server) is removed, significantly reducing the risk of privacy infringements [26].

2.3.1 System Model for Federated Learning

We consider an FL system with M users, each possessing its own local dataset $\mathcal{B}_m$ with size B_m, where $m \in [M]$. The total amount of data in the overall system

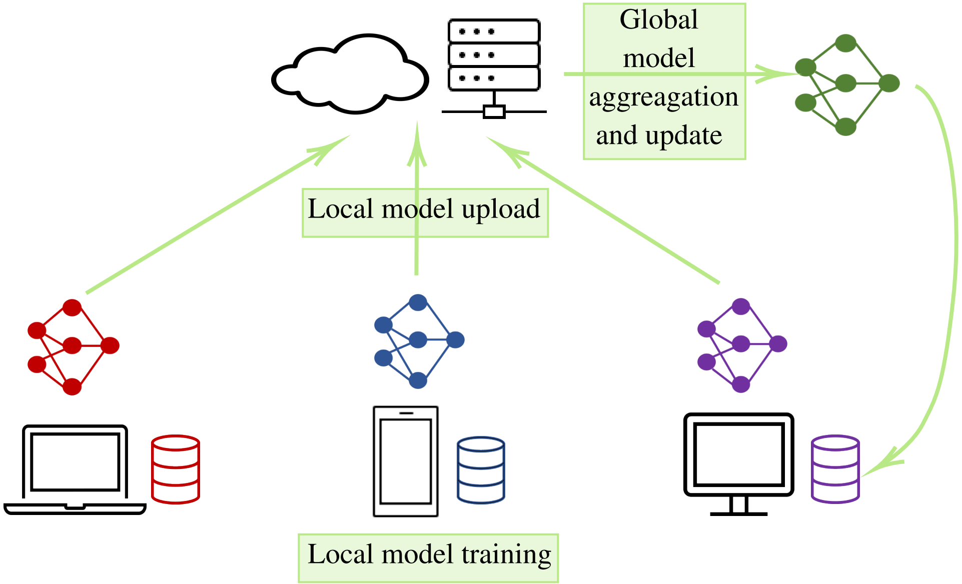


Figure 2.4: System model for general federated learning.

is denoted as B, such that $B \triangleq \sum_{m=1}^{M} B_m$. For the learning process of FL, SGD is employed. The local gradients calculated by each user are sent to a central server, commonly referred to as the PS, as illustrated in Fig. 2.4.

The overall loss function of the considered learning model is defined as

$$F\left(\boldsymbol{\theta}\right) = \sum_{m=1}^{M} \frac{B_m}{B} F_m\left(\boldsymbol{\theta}\right), \tag{2.11}$$

where $F_m\left(\boldsymbol{\theta}\right)$ is the empirical loss function of the m-th user for $m \in [M]$.

At the beginning of each global iteration t, the parameter server transmits the global parameter $\boldsymbol{\theta}(t)$ to all the users. Upon receiving the global parameter, the m-th user performs ϵ local iterations, following the update rule

$$\boldsymbol{\theta}_m^{p+1}(t) = \boldsymbol{\theta}_m^{p}(t) - \eta_m^p(t)\nabla F_m\left(\boldsymbol{\theta}_m^p(t), \xi_m^p(t)\right), \tag{2.12}$$

where $p \in [\epsilon]$, $\eta_m^p(t)$ is the learning rate for the m-th user during the p-th local iteration of the t-th global iteration. $\nabla F_m\left(\boldsymbol{\theta}_m^p(t), \xi_m^p(t)\right)$ is the stochastic gradient estimate in which $\xi_m^p(t)$ is the local mini-batch sample. The local iterations are initialized as $\boldsymbol{\theta}_m^1(t) = \boldsymbol{\theta}(t)$.

At the end of each iteration t, after performing ϵ local updates, each user m transmits its update to the PS, i.e.,

$$\Delta\boldsymbol{\theta}_m(t) = \boldsymbol{\theta}_m^{\epsilon+1}(t) - \boldsymbol{\theta}(t), \tag{2.13}$$

where $m \in [M]$. In an ideal scenario without any noise or interference, the update at the parameter server is given by

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \Delta\boldsymbol{\theta}(t), \tag{2.14}$$

where $\Delta\boldsymbol{\theta}(t)$ is the average of all the local updates which is

$$\Delta\boldsymbol{\theta}(t) = \frac{1}{M}\sum_{m=1}^{M} \Delta\boldsymbol{\theta}_m(t). \tag{2.15}$$

Then, this updated global parameter is shared with all the users, and the next iteration starts. Training continues iteratively until a predetermined number of iterations is reached or until a desired level of accuracy or loss is achieved.

2.3.2 FL with Over-the-Air Aggregation

Federated learning is a collaborative learning approach in which several users can participate without sharing their local data with other participants or the parameter server. While this approach allows for the distribution of the computational load to the participants, it also enhances privacy and security, since there is no explicit raw data sharing.

In FL, as given in (2.15), the global parameter update is performed by averaging the local updates with the help of SGD, and individual local updates are not directly used. Hence, the process of locally averaging gradients can be significantly enhanced through the utilization of over-the-air computation, a concept introduced in the previous section. With this approach, the workers are not required to be allocated to different resource slots; instead, they can utilize the same slot for their simultaneous transmissions, resulting in a significant reduction in transmission costs. Consequently, the gradient averaging method can be carried out over-the-air, promoting transmission efficiency in federated learning.

2.3.3 Advances in Federated Learning

The rapid growth of data sensing and collection capabilities of computation devices facilitates the use of massive datasets enabling performing the ML task in a distributed manner with the help of federated learning, has recently drawn significant attention [23, 27]. In federated learning, each device connected to the central processor performs the required gradient computation based on its local dataset, and sends it to the central processor. The global parameter update is performed at the central processor using the local computations of the connected devices.

While federated learning can be considered as a combination of two broadly studied areas: statistical learning and communications, it also opens up new research avenues. With this motivation, different problems related to federated learning are studied in the recent literature. These include studies on the effects of energy constraints, resource allocation, privacy, compression of local computations, convergence analysis of the learning algorithms, and performance over different channel models. In particular, in [28], digital and analog distributed stochastic gradient descent (D-DSGD and A-DSGD) algorithms over a Gaussian MAC are proposed. The authors use the superposition property of the MAC to recover the mean of the local gradients computed at remote workers. In D-DSGD, workers digitally compress their locally computed gradients into a finite number of bits, while in A-DSGD, workers use an analog compression similar to what is done in compressed sensing (CS) to obey the bandwidth limitations. In [29] and [30], the channel between the PS and the workers is modeled as a fading MAC. Ref. [29] performs power allocation among the gradients to schedule workers according to their CSI. The authors show that the latency reduction of the proposed method scales linearly with the device population. Ref. [30] proposes a gradient sparsification method which is followed by a CS algorithm to reduce the dimensions of a large parameter vector. By reducing the dimensionality of the gradients and designing a power allocation scheme, the authors obtain significant performance improvements compared to the existing benchmarks. In [31], the authors concentrate on achieving communication efficiency in FL over a multiple-input multiple-output (MIMO) MMAC. In the uplink transmission phase, where local updates are sent from workers to the PS server, the workers employ a technique known as block sparsification to map a high-dimensional local gradient to multiple lower-dimensional gradient vectors. On the PS side, the process of obtaining the average of local updates involves performing joint MIMO detection and sparse local-gradient recovery iteratively, drawing inspiration from the principles of turbo decoding.

In addition to the studies that decrease the communication load, [32] considers transmission energy, and formulates an optimization problem for the joint learning and communication process. The goal is to minimize the total energy consumption for local computations and wireless transmission under latency constraints. In [33], the authors focus on the minimization of the convergence time of a federated learning system by jointly considering user selection and resource allocation. The aim of the PS is to include as many workers as possible in the learning process for convergence to the global model with limited resources. There are also several studies on data exchange rate reduction via quantization [34, 35, 36, 37]. Specifically, in [37], the authors introduce a lossy federated learning (LFL) system, which directly quantizes both the global and the local model parameters to reduce the communication loss. They show that the convergence of the learning algorithm is guaranteed despite the quantization process. When the training data is randomly split among the workers, LFL with a small number of quantization levels performs as well as a system with unquantized parameters. In another line of research, [38] considers a federated learning system for which there is no CSI at the workers; hence the PS employs multiple antennas to align the received signals. In [39], this study is extended further, and a convergence analysis for the blind federated learning with both perfect and imperfect CSI is performed.

Ref. [40] proposes a robust distributed computing scheme where the destructive effects of slower workers are mitigated, while [41] studies Byzantine resilient distributed learning systems where failures may occur due to outside attacks, software bugs and synchronization problems. In [42], a Byzantine-resilient federated learning framework is explored specifically for datasets with heavy-tailed distributions. To ensure robustness against both Byzantine worker nodes and the challenges posed by heavy-tailed data, the authors employ a combination of convex and non-convex optimization techniques. Additionally, they implement gradient compression methods to alleviate the communication overhead associated with this resilient federated learning setup. In [43], a Byzantine-robust secure aggregation scheme for federated learning is introduced. This scheme is designed to withstand various challenges, including user dropouts, collusion, and Byzantine adversarial attacks. It accomplishes this robustness through the strategic utilization of ramp secret sharing and coded computing techniques. In [44], the authors leverage the additive structure of the Weiszfeld algorithm to craft a Byzantine-robust over-the-air federated learning model. This approach enhances the communication efficiency of the federated learning system. Byzantine-robust FL systems are further explored in [45, 46, 47, 48, 49].

Despite the absence of direct local data uploads in FL, the privacy of participants can still be at risk due to reverse attacks, wherein user data can be analyzed through the examination of their uploaded updates. In [50], the authors devise an efficient privacy-preserving data aggregation method based secret sharing. Utilizing homomorphisms of secret sharing, the PS can efficiently obtain the sum of local updates from the participating users without the need to access or learn each user's individual data. This method serves the dual purpose of averting local model leakages and resisting reverse attacks, improving the privacy of participants in FL. In [51], the authors introduce a privacy-preserving defense strategy that relies on two-trapdoor homomorphic encryption. This strategy is designed to prevent model poisoning attacks in FL. Notably, many existing defense mechanisms against data poisoning fall short when attackers employ encrypted poisonous gradients that cannot be easily identified. However, the proposed approach incorporates a secure cosine similarity method, which allows for the measurement of the distance between two encrypted gradients. This technique aids in the identification of encrypted malicious gradients, enhancing the security and resilience of FL systems against model poisoning attacks. In [52], the authors employ an approach that combines adaptive gradient descent with differential privacy. Adaptive gradient descent is utilized to dynamically adjust the learning rate during training, helping to prevent issues like overfitting and fluctuations, thereby enhancing both efficiency and performance. Concurrently, differential privacy is applied to imrpve the resistance against background knowledge attacks, further enhancing the privacy and security of the FL process. Privacy and security in FL are further investigated in [53, 54, 55, 56].

With the focus on downlink transmission from the central server to the workers, [57] considers both digital and analog approaches. In the digital approach, the PS quantizes the global parameter and utilizes a capacity achieving channel code, while the analog one transmits the global parameter without any coding. In [58], the authors focus on the federated learning in conjunction with quantization and user scheduling. Initially, an upper bound on the loss function is established, which subsequently undergoes minimization while adhering to latency constraints. This optimization considers several factors, including parameter quantization, user scheduling, channel bandwidth, and transmit power, to achieve efficient federated learning. In [59], a novel quantization approach for FL known as doubly-adaptive quantization is introduced. This approach involves the dynamic adjustment of quantization levels, which can vary not only over time but also among individual clients, enhancing the adaptability and efficiency of the FL system. From a different standpoint, in [60], the authors employ an adaptive approach to adjust quantization levels based on local gradient updates. More precisely, gradients with a larger updates are quantized and transmitted using a greater number of bits, whereas gradients with lower updates are quantized with fewer bits. This approach effectively leverages the heterogeneity in local data distributions and concurrently reduces the overall transmission cost. In [61], quantization is integrated into both the training process of FL and the uplink transmission phase. To achieve this, the authors employ fixed-precision format quantized neural networks (QNNs) within the FL framework. The primary objective of this approach is to minimize energy consumption and reduce the number of communication rounds. This optimization is achieved by formulating a multi-objective problem that considers various factors, including the number of iterations, device selection, and quantization levels, all contributing to the overall efficiency of the FL system. In [62], the authors propose a novel approach to gradient compression for training DNNs within the FL framework. This approach draws inspiration from rate-distortion principles and focuses on scalar-quantizer design. Specifically, two rate-distortion principles are considered: the *M-magnitude weighted L_2 distortion measure,* which yields higher fidelity for larger gradients, and the 2 *degrees of freedom distribution fitting,* which involves fitting the gradient distribution using a distribution characterized by two degrees of freedom. Subsequently, a scalar quantizer is designed to minimize the expected distortion in gradient reconstruction, enhancing the efficiency of the FL system. FL with quantization is further investigated in [63, 64, 65, 66].

In [67], FL with energy harvesting is considered where the energy arrivals are heterogeneous. The authors propose weighted averaging with respect to the latest energy arrivals and data cardinalities to prevent bias and show that the performance of the proposed approach is similar to that of full participation. Ref. [68] studies hierarchical over-the-air federated learning (HOTAFL) to investigate the limitation introduced by mobile users, and its convergence analysis is performed. In [69], a hierarchical edge-based FL system exploiting the similarity between the local and global model parameters to avoid uploading unnecessary local updates is proposed. The effect of the distance of the cell-edge users to the base station [70], hierarchical clustering for non-i.i.d data [71], and an evolutionary game theory approach for worker cluster selection decisions [72] in hierarchical FL are also studied. In another line of research, the authors in [73] propose a flexible communication and compression scheme that balances the energy consumption of local computations and wireless communication costs. In [74], the authors focus on an FL system with heterogeneous user data and optimize the resource allocation by considering the trade-off between the convergence time and energy consumption. In [75], client scheduling for FL is studied where the links between the users and the PS are resource-constrained and unreliable. The authors derive the convergence rate for the proposed system by taking into account both scheduling and inter-cell interference. Resource allocation for users where both the channel conditions and significance of the local model updates are important factors is considered in [76]. Different aspects of user/client selection [77, 78, 79, 80], user scheduling [81] and resource allocation [82, 83, 84] are also studied.

2.4 Wireless Inference

DNNs are one of the most promising and widely used machine learning methods which allow intelligent mobile applications to perform a very complicated task with highly accurate and reliable results. With the advancement in this area, DNNs are successfully applied in many domains, e.g., computer vision, smart driving, and natural language processing. However, due to their high computational complexities, DNN-based applications cannot be fully employed in simple edge devices like the Internet of Things (IoT) devices due to their limited computational capabilities, latency, and energy consumption requirements of the given application.

2.4.1 Solutions and Strategies for Wireless Inference

As a straightforward solution to address the challenges introduced by the complexity of the DNNs, the IoT device may prefer to upload the sensed data to another helping server which can easily deploy a DNN with any complexity to perform the inference. However, this raw data upload will excessively increase transmission offload, which is an undesired outcome. Note that this raw data may include many irrelevant data with the task, resulting in an unnecessary communication load. Additionally, uploading raw data raises potential privacy concerns, as sensitive information could be unintentionally disclosed. The alternative solution is to employ the whole network at the sensor side; however, it may not be possible to perform all the required calculations for the inference of a complex network with necessary reliability. Therefore, for practical applications, [85] removes unimportant weights of the network, which is achieved by using second-derivative information for the trade-off between the training error and complexity. This method, widely referred to as pruning, results in better generalization and faster learning/inference processes. Ref. [86] consider a similar approach for general error measures to achieve a reduced computational and storage implementation. A more detailed and timely literature search on pruning to accelerate the inference phase can be found in [87].

The previously mentioned methods, i.e., uploading the raw data without any processing and fully employing the network on the sensor side, are two opposite perspectives, each with its advantages and disadvantages. Hence, as a straightforward solution, the studies in the literature focus on splitting the network between the IoT device and another helper edge server. With this proposed approach, the IoT device pre-processes the raw data (i.e., performs the forward propagation up to a cut layer) by only using its limited computational capabilities and energy limitations. The resulting feature vector is transmitted to the edge server, which completes the forward propagation and obtains an inference result for its given task. Note that the edge server can be a very high-capacity device that can easily perform complicated computation tasks. Here the main focus is to minimize the number of computations at the IoT device while maintaining reasonable inference accuracy under the latency limitations of the given task which is commonly referred as *wireless inference* or *device-edge co-inference*. A simple system model for wireless inference with a single sensor setup is illustrated in Fig. 2.5.

It is essential to highlight that wireless inference primarily focuses on the online inference phase, whereas federated learning is primarily concerned with the distributed training phase. Therefore, in wireless inference, the training can be carried out in an online manner for the given setup beforehand, without taking into account the cost of training. The main objective is to optimize and enhance

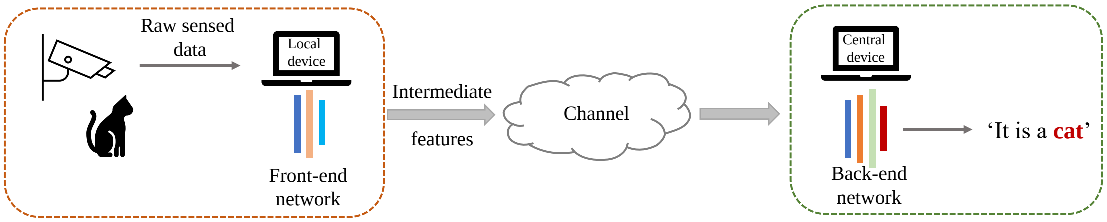


Figure 2.5: System model for general wireless inference.

the real-time inference process without explicit consideration of the training overhead.

2.4.2 Advances in Wireless Inference

With the rise in the quality and quantity of edge devices, the concept of wireless inference has drawn significant attention recently. In [88], a network splitting approach is proposed where the authors introduce an end-to-end architecture called Bottlenet++ in which the encoder and decoder act as a machine learning-based joint source-channel coder, considering channel impairments as a parameter of the overall network. In [19], the authors study a similar scenario with feature compression and reliable communication via deep joint source-channel coding (DeepJSCC) over additive white Gaussian noise (AWGN) channels and network pruning is employed to handle bandwidth limitations. In [89], wireless image retrieval problem is considered where an edge device/sensor captures an image of an object as raw data, and transmits the corresponding low-dimensional signature with the aim of retrieving similar images belonging to the original object from another dataset.

Most existing works in device-edge co-inference (or wireless inference) focus on DNN-based applications. However, Graph Neural Networks (GNNs) are powerful tools, especially for cloud processing. On the other hand, due to the very severe data amplification effect of GNNs, the existing methods for device-edge co-inference cannot be directly extended to GNN-based inference. In [90], the authors propose a low-latency co-inference framework, called Branchy-GNN, where the computations at the edge device are both controlled by network splitting and early exit mechanism. This early-exit mechanism helps to reduce edge device computations while a learning-based JSCC algorithm is implemented to cope with the wireless channel and reduce the latency. The proposed approach overperforms the standard network splitting, only-edge computation, and only-device computation under several wireless channel scenarios. In [91, 92], the authors study decentralized inference with GNNs over imperfect wireless channels, with a focus on enhancing privacy in [92].

In references [93] and [94], the authors use the information bottleneck (IB) principle [95, 96] to formulate the trade-off between feature informativeness and inference performance for device-edge co-inference in task-oriented applications. As identified in previous studies on different aspects of wireless inference, it can be an effective solution to improve edge device inference capabilities while balancing communication overhead by utilizing computational resources at both the edge devices and the edge server [97, 98, 99, 100].

In [101], a similar computing hierarchy consisting of cloud, edge and end devices where the DNN layers are mapped into hierarchical parts and the whole network is optimized to maximize the usefulness of the features, but the effect of wireless channels and efficiency of transmission is ignored. In [102], deep over-the-air computation is introduced for transmission efficient distributed inference over wireless channels using multiple sensor devices. The study employs averaging operation as the feature fusion method.

As in FL, privacy can also be a significant concern in device-edge co-inference. In [103], the authors introduce a framework called “Roulette,” which is designed to preserve privacy in the context of collaborative inference with a focus on task-oriented semantic privacy. This framework treats sensitive information, such as the ground truth of the data (e.g., class labels), as private. The proposed approach leverages split learning, where the back-end network remains frozen, and the front-end serves as both a feature extractor and an encryptor. Additionally, the authors provide a guarantee of differential privacy and conduct an analysis of the security regarding ground truth inference attacks.

2.5 Chapter Summary

In this chapter, we first provide a summary of the fundamental concepts in neural networks and deep learning, offering essential background information and laying the groundwork for the subsequent discussions. Next, we explain over-the-air computation, where the superposition signal is achieved through the simultaneous transmission of transmitters in a resource-efficient manner and can be used in the efficient implementation of distributed SGD. We also introduce the basics of federated learning, a distributed learning method, drawing from existing literature on the subject. Consequently, the training phase of federated learning can be carried out in an over-the-air manner, promoting transmission efficiency. Finally, we explore wireless inference, which focuses on the inference phase of a pretrained network, making this approach a counterpart of federated learning, and review recent advances in this domain.

Chapter 3

Blind Federated Learning at the Wireless Edge with Low-Resolution ADC and DAC

In this chapter, our main objective is to study federated learning over wireless channels in realistic settings by considering practical implementation issues as well as the wireless channel effects. We model the communication link as a frequency selective fading channel, and transmit the local gradients using OFDM. We consider the blind transmitter scenario, i.e., there is no CSI at the transmitters, hence multiple (even a massive number of) antennas are employed at the receiver side. Furthermore, to reduce the hardware complexity and power consumption, we employ low-resolution DACs at the transmitter side (at each worker), and ADCs at the receiver side. In fact, this is nothing but the over-the-air machine learning, except that here we are taking into account the effects of the wireless medium as well as the use of low-resolution DACs and ADCs.

The main contributions of this chapter can be summarized as follows:

- Different from previous works regarding federated learning reviewed above ([38, 28, 30, 37, 29, 34, 35, 36, 39]), we consider a realistic wireless channel model where the channel between the workers and PS is modeled as a multipath fading MAC.

- To cope with the realistic channel impairments, we transmit the local gradients using OFDM with a cyclic prefix (CP) to mitigate the ISI caused by the multipath. Thus, different from [37], we consider the transmission and reception of actual OFDM signals as would be necessitated in a practical implementation.

- Since one of our main concerns is a practical implementation of federated learning, we also employ low-resolution DACs and ADCs separately at the workers and the PS side, respectively. Also, we extend our studies to the case of a system which utilizes both low-resolution DACs and ADCs.

- Via both theoretical analysis and extensive simulations, we find that the effects of imperfections due to finite resolution DACs and/or ADCs can be alleviated using a sufficient number of receive antennas at the PS, and the convergence of the distributed learning algorithm is guaranteed even if we employ low-cost (even one-bit) DACs and/or ADCs.

The chapter is organized as follows. Section 3.1 introduces the system model and preliminaries. DSGD with low-resolution DACs is analyzed in Section 3.2, and the effect of low-resolution ADCs at the receiver side is studied in Section 3.3, respectively. Joint utilization of low-resolution DACs and ADCs are considered in Section 3.4. Performance of blind federated learning with realistic channel effects and hardware limitations is studied via simulations in Section 3.5, and the chapter is concluded in Section 3.6.

Note that our initial work on this topic is given in [104].

Notation: Throughout this chapter, the real and imaginary parts of $x \in \mathbb{C}$ are represented by x^R and x^I, respectively. We use the notation $[a\ b]$ to indicate the integer set $\{a, \ldots, b\}$ where $a \leq b$, a and b are positive integers, and $[b] = [1\ b]$. We denote l_2 norm of a vector $\mathbf{x}$ by $||\mathbf{x}||_2$. The entry in the i-th row and j-th column of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}[i, j]$. N-point Discrete Fourier Transform

(DFT) of vector $\mathbf{x} \in \mathbb{C}^N$ is defined as

$$\mathbf{X}[u] = \sum_{n=1}^{N} \mathbf{x}[n] e^{-j2\pi nu/N}. \tag{3.1}$$

while the N-point inverse discrete Fourier Transform (IDFT) of vector $\mathbf{X} \in \mathbb{C}^N$ is given by

$$\mathbf{x}[n] = \frac{1}{N} \sum_{u=1}^{N} \mathbf{X}[u] e^{j2\pi nu/N}. \tag{3.2}$$

3.1 System Model

We consider a distributed ML system where each worker calculates its gradient estimate and sends it to a central PS through a multipath fading MAC using OFDM as illustrated in Fig. 4.1. At the receiver side, OFDM demodulation, signal combining and global model parameter update are performed. The global parameter is broadcast to the workers over an error-free link. We assume that there is no transmit side CSI, and that the PS employs multiple antennas to recover the average of the workers' gradients. With the use of a higher number of workers and many antennas, a significant amount of power at the transmitter and receiver is consumed by the DACs and ADCs [105]. The power consumption of DACs and ADCs increases linearly, and their hardware cost increases exponentially with the number of quantization bits [106]. In order to keep the implementation cost and power consumption low, we consider a distributed learning system where the transmitters and receivers are equipped with low-resolution, even one-bit, DACs and ADCs, respectively.

We jointly train a learning model by using iterative SGD to minimize a loss function $f(\cdot)$. During the t-th iteration, worker $m \in [M]$ calculates the gradient estimate $\mathbf{g}_m^t \in \mathbb{R}^d$ by processing its local dataset $\mathcal{B}_m$ according to $\frac{1}{|\mathcal{B}_m|} \sum_{u \in \mathcal{B}_m} \nabla f(\boldsymbol{\theta}_t, u)$ where $\boldsymbol{\theta}_t \in \mathbb{R}^d$ is the vector of model parameters, d is the number of model parameters, and $g_m^t[n]$ represents the n-th entry of the gradient estimate. We form the baseband frequency domain signal of the local gradient

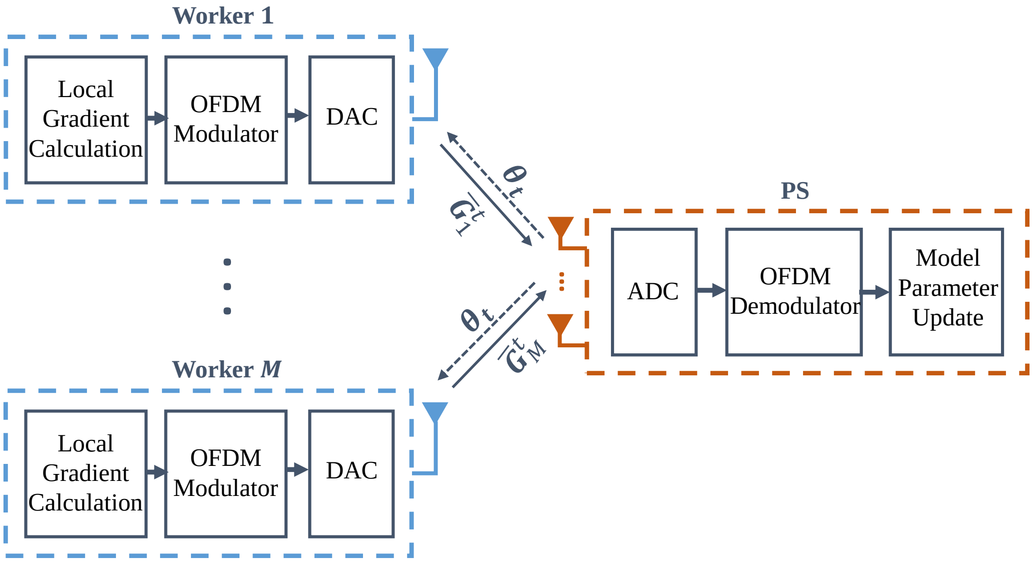


Figure 3.1: System model for distributed machine learning at the wireless edge.

vector as

$$\hat{\mathbf{g}}_m^t = \left[g_m^t[1] + jg_m^t[s+1],\, g_m^t[2] + jg_m^t[s+2], \cdots, g_m^t[s] + jg_m^t[2s]\right], \tag{3.3}$$

where $s = \lceil d/2 \rceil$, $\hat{\mathbf{g}}_m^t \in \mathbb{R}^s$, and $g_m^t[2s]$ is assigned as zero if $d \equiv 1 \pmod 2$. Then, the first step is to form the OFDM signal by taking an N-point IDFT of the gradient vector as

$$G_m^t[u] = \frac{1}{N} \sum_{n=1}^{N} \hat{g}_m^t[n] e^{j2\pi nu/N}, \tag{3.4}$$

for $u \in [N]$. If $s < N$, $\hat{g}_m^t[n] = 0$ for $n > s$, i.e., $\hat{\mathbf{g}}_m^t$ is zero padded.

The channel between the m-th worker and the k-th antenna of the PS is modeled as a (wireless) multipath MAC. We assume that the channel does not change during the transmission of one OFDM word, while it may be different for different OFDM words. The impulse response of the channel is

$$h_{mk}^t[n] = \sum_{l=1}^{L} h_{mkl}^t \delta[n - \tau_{mkl}], \tag{3.5}$$

where $n \in [N + N_{cp}]$, L is the number of channel taps, τ_{mkl} is the time delay and $h^t_{mkl} \in \mathbb{C}$ is the gain of the l-th channel tap from the m-th worker to the k-th antenna of the PS. Note that this is nothing but the machine learning over-the-air framework of [38]. We assume that h^t_{mkl} are zero-mean (circularly symmetric) complex Gaussian with $\mathbb{E}\left[(h^t_{mkl}) \cdot (h^t_{m'k'l'})^*\right] = 0$ for $(m, k, l) \neq (m', k', l')$, and $\mathbb{E}\left[|h^t_{mkl}|^2\right] = \sigma^2_{h,l}$, i.e., all the channel taps experience Rayleigh fading.

To mitigate the ISI caused by the multipath channel, CP addition is performed, i.e.,

$$\bar{\mathbf{G}}^t_m = \left[G^t_m[N - N_{cp} + 1] \dots G^t_m[N] \; G^t_m[1] \dots G^t_m[N]\right], \tag{3.6}$$

where $\bar{\mathbf{G}}^t_m \in \mathbb{C}^{N+N_{cp}}$ is the OFDM word to be transmitted by the m-th worker. The CP length N_{cp} is chosen to be greater than the delay spread of all the channels. The resulting (depending on the setup – quantized or full resolution) OFDM words are transmitted to the PS which are equipped with K receive antennas. The PS uses the received signal to update the model and sends it back to all the receivers over an error-free link.

At the k-th receive chain, after removing the CP, the n-th entry of the received vector at the input of the k-th receive antenna during iteration t is written as

$$Y^t_k[n] = \sum_{m=1}^{M} \sum_{l=1}^{L} h^t_{mkl} G^t_m[n - \tau_{mkl}] + z^t_k[n], \tag{3.7}$$

where the additive noise terms $z^t_k[n]$ are independent and identically distributed (i.i.d.) circularly symmetric zero mean complex Gaussian random variables, i.e., $z^t_k[n] \sim \mathcal{CN}(0, \sigma^2_z)$ for $k \in [K]$.

Ideally, the PS updates the model parameter according to $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mu_t \frac{1}{M} \sum_{m=1}^{M} \mathbf{g}^t_m$, and it is shared with the workers. However, in our setup, the local gradients are not available at the PS, instead the PS uses noisy and distorted version (by low-resolution DACs and/or ADCs) of the local gradients to recover the estimate of the gradient vector as will become apparent in the subsequent sections. In the following, we drop the subscripts referring to iteration index t for ease of exposition.

3.2 DSGD with Low-Resolution DACs at the Workers

In this section, we study the effects of employing low-resolution DACs at the workers on the distributed learning process in an effort to reduce the hardware complexity and power consumption.

After constructing the OFDM word corresponding to the gradient vectors, a complex-valued low-resolution DAC is employed to generate the transmitted signal at each worker. A b-bit complex-valued DAC consists of two parallel real-valued DACs with quantization function $Q_b(\cdot)$. The real and imaginary parts are separately quantized into $\beta = 2^b$ reconstruction levels. The reconstruction levels are denoted by $\hat{\mathbf{a}} = [\hat{a}_1 \ \hat{a}_2 \cdots \hat{a}_\beta] \in \mathbb{R}^\beta$ while the boundaries of the quantization regions are denoted by $\hat{\mathbf{x}} = [\hat{x}_1 \ \hat{x}_2 \cdots \hat{x}_{\beta+1}] \in \mathbb{R}^{\beta+1}$ where $\hat{x}_1 = -\infty$ and $\hat{x}_{\beta+1} = +\infty$ for convenience. Also, we have, $\hat{a}_i < \hat{a}_j$, if $1 \leq i < j \leq \beta$, $\hat{x}_i < \hat{x}_j$ if $1 \leq i < j \leq \beta + 1$, and $\hat{x}_i \leq \hat{a}_j < \hat{x}_k$ if $1 \leq i \leq j < k \leq \beta + 1$. The corresponding real valued quantizer is $Q_b(z) = \hat{a}_i$ for $\hat{x}_i \leq z < \hat{x}_{i+1}$, $i \in [\beta]$, $z \in \mathbb{R}$. The complex-valued DAC operation can be expressed as $Q_b(x) = Q_b(x^R) + jQ_b(x^I)$. We assume that the quantizer output is chosen such that $Q_b(x) = \mathbb{E}[X|Q_b(X)]$, i.e., the reconstruction level is selected to minimize the mean squared error for each quantization region. The corresponding signal to quantization noise ratio (SQNR) of the input vector $\mathbf{x}$ is calculated as

$$\mathrm{SQNR} = \frac{\mathbb{E}\left[|X|^2\right]}{\mathbb{E}\left[|Q_b(X) - X|^2\right]}. \tag{3.8}$$

We model the OFDM words as wide-sense stationary (WSS) Gaussian processes based on an argument similar to the one made in [107]. That is, if the input data which forms the OFDM word is i.i.d. and bounded, the convex envelope of the OFDM word weakly converges to a Gaussian random process as the number of subcarriers goes to infinity through an application of central limit theorem (CLT). Similarly, if we assume that the elements of the gradient vector in the learning process are i.i.d. and bounded, then the real and imaginary parts

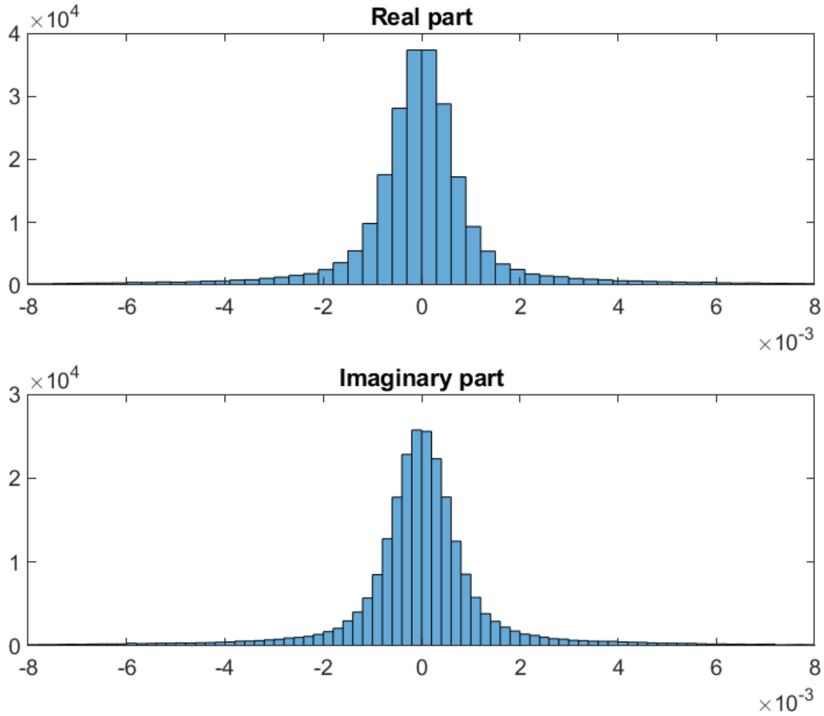


Figure 3.2: Histogram of the real and imaginary parts of an exemplary OFDM word during the learning task with our setup.

of the baseband OFDM word obtained from the gradient vector can be modeled as independent zero-mean stationary Gaussian processes. As a verification, we examine histograms of several OFDM word samples obtained by a certain learning task with our setup. An instance of an exemplary histogram of the OFDM word samples obtained through the 100-th iteration is given in Fig. 3.2 which is consistent with our assumption. Our extensive experiments further confirm that the corresponding OFDM word samples at different time indexes have almost the same variance. Note that, even if the OFDM words are not Gaussian processes, the Bussgang theorem that will be used to model the nonlinear input-output relationship for DACs and ADCs is still a good approximation as illustrated extensively in the literature, see, e.g., [108]-[109].

We denote the autocorrelation matrix of the OFDM words by $\mathbf{C}_{\bar{\mathbf{G}}_m\bar{\mathbf{G}}_m}$ with equal diagonal elements denoted by $\sigma^2_{G_m}$. Using the Bussgang decomposition [110]-[111], we can write the quantized signal in two parts: the desired signal

Table 3.1: Distortion factors with different quantization levels [2, 3].

Number of bits	**Distortion factor** (η)
1	0.3634
2	0.1175
3	0.03454
4	0.009497
5	0.002499

component and the quantization distortion which is uncorrelated with the desired signal, that is,

$$\bar{G}_m^Q[n] = Q(\bar{G}_m[n]) = (1-\eta)\bar{G}_m[n] + q_m[n], \tag{3.9}$$

where $\eta = 1/\text{SQNR}$ is the distortion factor which is the inverse of SQNR, and the variance of the distortion noise is $\sigma_{q_m}^2 = \eta(1-\eta)\sigma_{G_m}^2$. When a unit variance Gaussian input is processed by a non-uniform scalar minimum mean-square-error quantizer, the values of corresponding distortion factors are listed in Table 4.1 [2]-[3].

At the k-th receive chain, after removing the CP, the n-th entry of the received vector is written as

$$\begin{aligned} Y_k[n] &= \sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl} G_m^Q[n-\tau_{mkl}] + z_k[n] && (3.10)\\ &= \sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}\Bigg((1-\eta)\cdot G_m[n-\tau_{mkl}] + q_m[n-\tau_{mkl}]\Bigg) + z_k[n] && (3.11)\\ &= (1-\eta)\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl} G_m[n-\tau_{mkl}] + w_k[n], && (3.12)\end{aligned}$$

where the total non-Gaussian noise term $w_k[n]$ has variance $\sigma_z^2 + \eta(1-\eta)\sigma_{G_m}^2 \sum_{m=1}^{M}\sum_{l=1}^{L} |h_{mkl}|^2$.

To perform the demodulation, we take the DFT of (3.10) which gives

$$r_k[i] = (1-\eta)\sum_{m=1}^{M} H_{mk}[i]g_m[i] + \sum_{m=1}^{M} H_{mk}[i]Q_m[i] + Z_k[i], \tag{3.13}$$

where $Q_m[i]$ is the DFT of the quantization distortion noise and $H_{mk}[i]$'s are the channel gains from the m-th worker to the k-th receive chain for the i-th subcarrier. $H_{mk}[i]$'s are given by

$$\begin{aligned} H_{mk}[i] &= \sum_{n=0}^{N-1} h_{mk}[n]e^{-j2\pi in/N} \\ &= \sum_{n=0}^{N-1}\left(\sum_{l=1}^{L} h_{mkl}\delta[n-\tau_{mkl}]\right)e^{-j2\pi in/N} \\ &= \sum_{l=1}^{L} h_{mkl}e^{-j2\pi i\tau_{mkl}/N}. \end{aligned} \tag{3.14}$$

Since the channel taps are zero mean circularly symmetric complex Gaussian (i.e., Rayleigh fading), $H_{mk}[i]$'s are also zero-mean complex Gaussian random variables with variance $\sigma_H^2 = \sum_{l=1}^{L}\sigma_{h,l}^2$.

Taking the DFT of the channel noise vector, $Z_k[i]$ is evaluated as

$$Z_k[i] = \sum_{n=0}^{N-1} z_k[n]e^{-j2\pi in/N}. \tag{3.15}$$

The noise terms are i.i.d. circularly symmetric complex Gaussian, i.e., $Z_k[n] \sim \mathcal{CN}(0,\, \sigma_{Z_k}^2)$ where $\sigma_{Z_k}^2 = N\sigma_{z_k}^2$.

We assume that the CSI is available at the PS, hence the received signals from the K antennas can be combined to align the gradient vectors using

$$y[i] = \frac{1}{(1-\eta)\cdot K}\sum_{k=1}^{K}\left(\sum_{m=1}^{M}(H_{mk}[i])^*\right)r_k[i], \tag{3.16}$$

as in [38, 39]. By substituting (3.13) into (3.16), we obtain

$$y[i] = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk}[i]|^2 g_m[i]}_{\text{signal term}} \tag{3.17a}$$

$$+ \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}(H_{mk}[i])^* H_{m'k}[i] g_{m'}[i]}_{\text{interference term}} \tag{3.17b}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}(H_{mk}[i])^* H_{m'k}[i] Q_{m'}[i]}_{\text{distortion noise term}} \tag{3.17c}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk}[i]|^2 Q_m[i]}_{\text{second type of distortion noise term}} \tag{3.17d}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\left(\sum_{m=1}^{M}(H_{mk}[i])^*\right) Z_k[i]}_{\text{channel noise term}}. \tag{3.17e}$$

There are five different terms in (3.17): the signal component, interference, distortion noise term, the second type of distortion noise term, and the channel noise.

To analyze the interference term (3.17b), we write it as a summation of M terms

$$\begin{aligned}\frac{1}{K}\Bigg[\Bigg(\sum_{k=1}^{K}\sum_{m=2}^{M}(H_{mk}[i])^* H_{1k}[i]\Bigg) g_1[i] + \cdots \\ + \Bigg(\sum_{k=1}^{K}\sum_{\substack{m=1\\ m\neq j}}^{M}(H_{mk}[i])^* H_{jk}[i]\Bigg) g_j[i] + \cdots \\ + \Bigg(\sum_{k=1}^{K}\sum_{m=1}^{M-1}(H_{mk}[i])^* H_{Mk}[i]\Bigg) g_M[i]\Bigg],\end{aligned} \tag{3.18}$$

and consider the coefficient of each term $g_j[i]$ separately. Let us define

$$\kappa_j[i] = \frac{1}{K} \sum_{k=1}^{K} \sum_{\substack{m=1 \\ m \neq j}}^{M} (H_{mk}[i])^* H_{jk}[i], \tag{3.19}$$

for the coefficient of the j-th interfering gradient $g_j[i]$ in (3.17b) where $i \in [N]$, and $j \in [M]$. Since $H_{mk}[i]$ and $H_{jk}[i]$ are independent for $j \neq m$, the mean and variance of $\kappa_j[i]$ are calculated as

$$\mathbb{E}\left[\kappa_j[i]\right] = 0, \tag{3.20a}$$

$$\mathbb{E}\left[|\kappa_j[i]|^2\right] = \frac{(M-1)\sigma_H^4}{K}. \tag{3.20b}$$

We have M such interference terms in (3.17b) each for a different worker with zero mean, and variance scaling with $\frac{M-1}{K}$. Hence, the total interference term approaches zero as $K \to \infty$.

To analyze the distortion noise term (3.17c), we define the coefficient of each uncorrelated distortion term $Q_j[i]$ separately as in the case of (3.17b) by

$$\delta_{1j}[i] = \frac{1}{(1-\eta)K} \sum_{k=1}^{K} \sum_{\substack{m=1 \\ m \neq j}}^{M} (H_{mk}[i])^* H_{jk}[i], \tag{3.21}$$

where $i \in [N]$, and $j \in [M]$.

Similar to the analysis of $\kappa_j[i]$, the mean and variance of $\delta_{1j}[i]$ are calculated as

$$\mathbb{E}\left[\delta_{1j}[i]\right] = 0, \tag{3.22a}$$

$$\mathbb{E}\left[|\delta_{1j}[i]|^2\right] = \frac{(M-1)\sigma_H^4}{(1-\eta)^2 K}. \tag{3.22b}$$

This implies that each of the M interfering terms in (3.17c) goes to zero if K is large enough. Thus, the detrimental effect of the distortion noise term can also be eliminated by employing a large number of receive antennas.

To analyze the second type of distortion noise term (3.17d), we consider each term $Q_j[i]$ separately for $j \in [M]$, and define the coefficient of the interfering distortion term caused by the j-th one as

$$\delta_{2j}[i] = \frac{1}{(1-\eta)K} \sum_{k=1}^{K} |H_{jk}[i]|^2, \tag{3.23}$$

where $i \in [N]$, and $j \in [M]$. The mean of $\delta_{2j}[i]$ is

$$\mathbb{E}\left[\delta_{2j}[i]\right] = \frac{\sigma_H^2}{(1-\eta)}. \tag{3.24}$$

For the variance of $\delta_{2j}[i]$, we have

$$\mathbb{E}\left[|\delta_{2j}[i]|^2\right] = \frac{1}{(1-\eta)^2 K^2} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathbb{E}\left[|H_{jk_1}[i]|^2 \, |H_{jk_2}[i]|^2\right]. \tag{3.25}$$

- If $k_1 = k_2$ (case 2.1)

$$\mathbb{E}\left[|\delta_{2j}[i]|^2\right]\Big|_{\text{case 2.1}} = \frac{1}{(1-\eta)^2 K^2} \sum_{k=1}^{K} \mathbb{E}\left[|H_{jk}[i]|^4\right] \tag{3.26}$$

$$= \frac{1}{(1-\eta)^2 K} \mathbb{E}\left[|H_{jk}[i]|^4\right]. \tag{3.27}$$

- If $k_1 \neq k_2$ (case 2.2)

$$\mathbb{E}\left[|\delta_{2j}[i]|^2\right]\Big|_{\text{case 2.2}} = \frac{1}{(1-\eta)^2 K^2} \sum_{k_1=1}^{K} \sum_{\substack{k_2=1 \\ k_2 \neq k_1}}^{K} \mathbb{E}\left[|H_{jk_1}[i]|^2\right] \mathbb{E}\left[|H_{jk_2}[i]|^2\right] \tag{3.28}$$

$$= \frac{(K^2 - K)\sigma_H^4}{(1-\eta)^2 K^2} \tag{3.29}$$

$$\approx \frac{\sigma_H^4}{(1-\eta)^2}, \tag{3.30}$$

for $K \gg 1$. Thus, the mean and variance of the second distortion term of the j-th worker is calculated as

$$\mathbb{E}\left[\delta_{2j}[i]\right] = \frac{\sigma_H^2}{(1-\eta)}, \tag{3.31a}$$

$$\mathrm{Var}(\delta_{2j}[i]) \approx \frac{1}{(1-\eta)^2 K}\mathbb{E}\left[|H_{jk}[i]|^4\right]. \tag{3.31b}$$

Note that $\delta_{2j}[i]$ has a finite mean and its variance approaches zero as $K \to \infty$. We know that the mean of the distortion term, $Q_j[i]$ for all $j \in [M]$, is zero. Accordingly, using the law of large numbers, the summation will converge to the mean of $Q_j[i]$, which is zero, for a sufficiently large M.

Using the law of large numbers, as the number of antennas at the PS $K \to \infty$, the signal term can be approximated as

$$y_{\mathrm{sig}}[i] = \sigma_H^2 \sum_{m=1}^{M} g_m[i]. \tag{3.32}$$

Thus, with low-resolution DACs at the workers, the PS can recover the i-th entry of the desired signal using

$$\frac{1}{M}\sum_{m=1}^{M} g_m[i] = \begin{cases} \frac{y^R[i]}{M\sigma_H^2}, & \text{if } 1 \le i \le s, \\ \frac{y^I[i-s]}{M\sigma_H^2}, & \text{if } s < \ i \le 2s. \end{cases} \tag{3.33}$$

This result clearly shows that the destructive effect of low-resolution DACs can be effectively alleviated using a sufficient number of PS antennas. Thus, the convergence of the learning process is guaranteed even if we employ low-cost low-resolution DACs at the workers, which significantly reduces the cost of designing distributed learning systems with a high number of workers. On the other hand, using a very large number of PS antennas will increase both the design cost and energy consumption, hence it may not be efficient. For further assessment, we can consider the coefficients of the distortion terms. For the distortion noise term given in (3.17c), we have M contributing terms each with zero mean and variance $\frac{(M-1)\sigma_H^4}{(1-\eta)^2 K}$. To reduce the effects of these terms on the learning accuracy, it is desired to have this variance close to zero. Clearly, this variance depends on several parameters; hence, to evaluate the overall performance, we should not only consider the number of receive antennas K, but also the channel variance σ_H^2, number of workers M, and distortion factor $\eta \in [0,\ 1]$. For example, if we have a high-resolution DAC, η will be small; hence, using a smaller number of receive antennas may be sufficient to cancel out the resulting impairments. However, when the resolution is very low, e.g., for a one-bit DAC, η will be large, and we will need a higher number of receive antennas due to the $\frac{1}{(1-\eta)^2}$ term. A similar approach can also be used to analyze the second type of distortion noise term given in (3.17d) for which we have M contributing terms each with variance $\frac{1}{(1-\eta)^2 K}\mathbb{E}\left[|H_{jk}[i]|^4\right]$. In other words, there is a trade-off between the DAC resolution and the number of receive antennas, and the overall performance is also affected by the channel statistics.

3.3 DSGD with Low-Resolution ADCs at the PS

In this section, we consider a system where the workers transmit the OFDM words corresponding to the local gradients with full-resolution through a multipath fading channel while the PS employs low-resolution ADCs at each receive antenna, and analyze the convergence of the federated learning algorithm.

At each receive chain, after removing the CP, the n-th entry of the received OFDM word $\mathbf{Y}_k$ is

$$Y_k[n] = \sum_{m=1}^{M} \sum_{l=1}^{L} h_{mkl} G_m[n - \tau_{mkl}] + z_k[n]. \tag{3.34}$$

The $(k,\, k')$-th element of the auto-correlation matrix of $\mathbf{Y}[n] = [Y_1[n] \cdots Y_K[n]]$

received by different antennas can be written as

$$\begin{aligned}\mathbf{C}_{\mathbf{YY}}[k,k'] &= \mathbb{E}\Big[\sum_{m=1}^{M}\sum_{m'=1}^{M}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{mkl}h^*_{m'k'l'}G_m[n-\tau_{mkl}] \\ &\qquad \cdot G^*_{m'}[n-\tau_{m'k'l'}]\Big] + \sigma_z^2 \mathbb{1}_{\{k=k'\}} \qquad (3.35)\\ &= \sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{mkl}h^*_{mk'l'}\mathbb{E}\big[G_m[n-\tau_{mkl}] \\ &\qquad \cdot G_m[n-\tau_{mk'l'}]\big] + \sigma_z^2 \mathbb{1}_{\{k=k'\}}. \qquad (3.36)\end{aligned}$$

The variance of the received signal at the k-th antenna $Y_k[n]$ is given by

$$\begin{aligned}\sigma^2_{Y_k} &= \mathbb{E}\Big[\sum_{m=1}^{M}\sum_{m'=1}^{M}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{mkl}h^*_{m'kl'} \\ &\qquad \cdot G_m[n-\tau_{mkl}]G^*_{m'}[n-\tau_{m'kl'}]\Big] + \sigma_z^2 \qquad (3.37)\\ &= \sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{mkl}h^*_{mkl'} \\ &\qquad \cdot \mathbb{E}\left[G_m[n-\tau_{mkl}]G^*_m[n-\tau_{mkl'}]\right] + \sigma_z^2, \qquad (3.38)\end{aligned}$$

which only depends on k.

A complex-valued low-resolution ADC employed at each receive antenna performs quantization. As in the case with low-resolution DACs described in the previous section, we describe b-bit quantization with quantization function $Q_b(\cdot)$ that independently quantizes the real and imaginary parts into $\beta = 2^b$ reconstruction levels such that the quantizer output is chosen as $Q_b(x) = \mathbb{E}[X|Q_b(X)]$.

With element-wise quantization, we can decompose the quantized signal into two parts as the desired signal component and quantization distortion which is uncorrelated with the desired signal. Analytically, we can write the quantized signal as

$$R_k[n] = (1-\eta_k)\Bigg(\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl} G_m[n-\tau_{mkl}] + z_k[n]\Bigg) + w_q^k[n], \tag{3.39}$$

where η_k is the distortion factor which is the inverse of the SQNR due to quantization of $\mathbf{Y}_k$. To determine η_k, one can use Table 4.1. $w_q^k[n]$ is a non-Gaussian distortion noise at the k-th antenna whose variance is $\sigma^2_{w_q^k} = \eta_k(1-\eta_k)\sigma^2_{Y_k}$.

The receive antennas at the PS are equipped with identical ADCs. As explained in [111], while it may be tempting to think that the quantization noise terms at different ADCs are uncorrelated, this is generally not the case since each antenna receives different (delayed) linear combinations of the same set of OFDM words generated at the workers. On the other hand, as shown in [112], the distortion can be safely approximated as uncorrelated for massive MIMO systems with a sufficient number of users. We have also validated this approximation for our system, and observed that the correlation across the antennas of the PS is near-zero, even for the one-bit ADC case. Therefore, the correlations can be ignored as in the additive quantization noise model (AQNM), leading to a tractable scheme [113]. We further note that there are different studies on low-resolution ADCs which also neglect the distortion correlation among antennas as in our approach [2, 114]-[115]. For zero-mean Gaussian processes, this approach is equivalent to the Bussgang decomposition, except that it ignores the correlation among the elements of the distortion term.

If we define the total effective noise due to the channel and the quantization process as

$$w_k[n] = (1-\eta_k) z_k[n] + w_q^k[n], \tag{3.40}$$

the outputs of the complex ADCs can be written as

$$R_k[n] = (1-\eta_k)\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl} G_m[n-\tau_{mkl}] + w_k[n], \tag{3.41}$$

where $w_k[n]$ is non-Gaussian total noise with variance $\sigma^2_{w_k} = \sigma^2_{w_q^k} + (1-\eta_k)^2\sigma_z^2$, and it is assumed to be uncorrelated across the antennas.

To perform the OFDM demodulation, we take the DFT of (3.41) which results in

$$r_k[i] = (1-\eta_k)\sum_{m=1}^{M} H_{mk}[i]g_m[i] + W_k[i], \tag{3.42}$$

where $H_{mk}[i]$'s are the channel gains from the m-th worker to the k-th receive chain for the i-th subcarrier, given by (3.14), which are zero-mean Gaussian random variables with variance $\sigma_H^2 = \sum_{l=1}^{L}\sigma_{h,l}^2$.

Taking the DFT of the effective noise, $W_k[i]$ is given as

$$W_k[i] = \sum_{n=0}^{N-1} w_k[n]e^{-j2\pi in/N}. \tag{3.43}$$

We know that the channel noise is i.i.d., and we assume that the distortion noise decorrelates sufficiently fast. Hence, $W_k[i]$ converges absolutely to a Gaussian random variable by an application of the CLT [116], i.e., $W_k[n] \sim \mathcal{CN}(0, \sigma_{W_k}^2)$ where $\sigma_{W_k}^2 = N\sigma_{w_k}^2$.

Assuming that the CSI is available at the PS as in the previous section, the received signals from the K antennas can be combined to align the gradient vectors by

$$y[i] = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1-\eta_k}\left(\sum_{m=1}^{M}(H_{mk}[i])^*\right)r_k[i]. \tag{3.44}$$

By substituting (3.42) into (3.44), we obtain

$$y[i] = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk}[i]|^2 g_m[i]}_{\text{signal term}} \tag{3.45a}$$

$$+\underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{m'=1,m'\neq m}^{M}(H_{mk}[i])^* H_{m'k}[i]g_{m'}[i]}_{\text{interference term}} \tag{3.45b}$$

$$\underbrace{+\frac{1}{K}\sum_{k=1}^{K}\frac{1}{1-\eta_k}\left(\sum_{m=1}^{M}(H_{mk}[i])^*\right)W_k[i]}_{\text{noise term}}. \tag{3.45c}$$

There are three different terms in (3.45): the signal component, the interference and the noise. Using the law of large numbers, as the number of antennas at the PS $K \to \infty$, the signal term approaches

$$y_{\text{sig}}[i] = \sigma_H^2 \sum_{m=1}^{M} g_m[i]. \tag{3.46}$$

Thus, the PS can recover the i-th entry of the desired signal

$$\frac{1}{M}\sum_{m=1}^{M} g_m[i] = \frac{y_{\text{sig}}[i]}{M\sigma_H^2}. \tag{3.47}$$

To analyze the interference term (3.45b), we follow the same approach as in the previous section where each of the M interfering terms is analyzed separately. We define the term due to the j-th interfering worker as

$$\kappa_j[i] = \frac{1}{K}\sum_{k=1}^{K}\sum_{\substack{m=1\\ m\neq j}}^{M}(H_{mk}[i])^* H_{jk}[i], \tag{3.48}$$

where $i \in [N]$, and $j \in [M]$. Since $H_{mk}[i]$ and $H_{jk}[i]$ are independent for $j \neq m$, the mean and variance of $\kappa_j[i]$ are calculated as

$$\mathbb{E}\left[\kappa_j[i]\right] = 0, \tag{3.49a}$$

$$\mathbb{E}\left[|\kappa_j[i]|^2\right] = \frac{(M-1)\sigma_H^4}{K}. \tag{3.49b}$$

Accordingly, for fixed gradient values, each of the M interference terms in (3.45b) has zero mean and their variances scale with $\frac{M-1}{K}$. Thus, similar to the ideal case (where the receive chains are equipped with infinite resolution ADCs as considered in [38]), the interference term approaches zero as $K \to \infty$. In other words, using a sufficiently large number of antennas at the PS eliminates the destructive effects of the interference on the learning process, and the estimate for the gradient vector is obtained as

$$\frac{1}{M}\sum_{m=1}^{M} g_m[i] = \begin{cases} \frac{y^R[i]}{M\sigma_H^2}, & \text{if } 1 \leq i \leq s, \\ \frac{y^I[i-s]}{M\sigma_H^2}, & \text{if } s < i \leq 2s, \end{cases} \tag{3.50}$$

for $i \in [d]$. This result clearly shows that the convergence of the learning process is guaranteed even if we employ low-cost low-resolution ADCs at the receiver.

3.4 DSGD with Low-Resolution DACs and ADCs

We now consider a system where the workers and the PS employ low-resolution DACs and ADCs, respectively. Each worker uses a finite resolution DAC to quantize the OFDM words, and transmits them through a multipath fading channel. The PS receives the signal from multiple antennas where finite resolution ADCs are employed at each receive chain. The aim is to obtain an estimate of the gradients using the received signals, which are distorted by ADCs and DACs as well as the multipath fading channel impairments. We analyze the impact of employing finite resolution ADCs and DACs jointly on the convergence of the learning algorithm. We accomplish this by using the Bussgang decomposition and AQNM model for the quantization operation at the workers and the PS, respectively.

Each worker calculates their local gradients and their corresponding OFDM words $\bar{\mathbf{G}}_m \in \mathbb{C}^{N+N_{cp}}$. As in Section 3.2, each worker uses a finite resolution DAC, and quantizes the OFDM words corresponding to the local gradients. The n-th element of the transmitted signal by the m-th worker is

$$\bar{G}_m^Q[n] = Q(\bar{G}_m[n]) = (1-\eta)\bar{G}_m[n] + q_m[n] \tag{3.51}$$

using the Bussgang decomposition. Here $\eta = 1/\text{SQNR}$ due to the quantization of $\bar{G}_m[n]$, and the variance of the distortion noise is $\sigma_{q_m}^2 = \eta(1-\eta)\sigma_{G_m}^2$.

The quantized signals pass through a multipath fading channel whose impulse response is given in (3.5). After removing the CP, the received signal at the input of the finite resolution ADC of the k-th antenna of the PS is

$$U_k[n] = \sum_{m=1}^{M} \sum_{l=1}^{L} h_{mkl} \Big((1-\eta) G_m[n - \tau_{mkl}] + q_m[n - \tau_{mkl}] \Big) + z_k[n]. \tag{3.52}$$

The mean of $U_k[n]$ is zero, and its variance is given by

$$\begin{aligned} \sigma_{U_k}^2 = &\sum_{m=1}^{M} \sum_{l=1}^{L} |h_{mkl}|^2 \left((1-\eta)^2 + \eta(1-\eta) \right) \sigma_{G_m}^2 \\ &+ (1-\eta)^2 \sum_{m=1}^{M} \sum_{l=1}^{L} \sum_{l'=1, l' \neq l}^{L} h_{mkl} h_{mkl'}^* \\ &\cdot \mathbb{E}\Big[G_m[n - \tau_{mkl}] G_m[n - \tau_{mkl'}] \Big] + \sigma_z^2, \end{aligned} \tag{3.53}$$

which only depends on the receive antenna index k.

The PS employs finite resolution ADCs at each receive antenna. The quantization operation of the ADC can be modeled as a linear operation using an AQNM model where the correlation of distortion noise across the antennas is ignored. The corresponding quantized signal at the k-th antenna is written as

$$\begin{aligned} R_k[n] = (1-\eta_k) \Bigg(&\sum_{m=1}^{M} \sum_{l=1}^{L} h_{mkl} (1-\eta) G_m[n - \tau_{mkl}] \\ &+ \sum_{m=1}^{M} \sum_{l=1}^{L} h_{mkl} q_m[n - \tau_{mkl}] + z_k[n] \Bigg) + v_q[n], \end{aligned} \tag{3.54}$$

where η_k is the distortion factor due to quantization of the received signal at the k-th antenna ($\mathbf{U}_k$), and calculated through the SQNR of the corresponding quantization operation as $\eta_k = 1/\mathrm{SQNR}$. $v_q[n]$ is a non-Gaussian distortion noise whose variance is $\sigma_{v_q}^2 = \eta_k(1-\eta_k)\sigma_{U_k}^2$.

The total effective non-Gaussian noise due to the channel and quantization with ADC at the PS is

$$p_k[n] = (1-\eta_k)z_k[n] + v_q[n], \tag{3.55}$$

with variance $\sigma_{p_k}^2 = (1-\eta_k)^2\sigma_z^2 + \sigma_{v_q}^2$, and the output of the complex ADC can be rewritten as

$$\begin{aligned} R_k[n] = (1-\eta_k)(1-\eta)\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}G_m[n-\tau_{mkl}] \\ + (1-\eta_k)\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}q_m[n-\tau_{mkl}] + p_k[n]. \end{aligned} \tag{3.56}$$

For demodulation, we take the DFT of (3.56), which results in

$$\begin{aligned} r_k[i] = (1-\eta_k)(1-\eta)\sum_{m=1}^{M} H_{mk}[i]g_m[i] \\ + (1-\eta_k)\sum_{m=1}^{M} H_{mk}[i]Q_m[i] + P_k[i], \end{aligned} \tag{3.57}$$

where $H_{mk}[i]$'s are as defined in (3.14), and $Q_m[i]$ is the DFT of the quantization distortion noise.

Taking the DFT of the effective noise, $P_k[i]$ is evaluated as $P_k[i] = \sum_{n=0}^{N-1} p_k[n]e^{-j2\pi in/N}$. With a similar approach to the one used in Section 3.3, under some mild assumptions, $P_k[i]$ converges absolutely to a Gaussian random variable by an application of CLT [116].

Since the CSI is only available at the PS as in [38], the received signals can be combined to align the gradient vectors as

$$y[i] = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{(1-\eta)(1-\eta_k)}\left(\sum_{m=1}^{M} H_{mk}[i]\right)^* r_k[i]. \tag{3.58}$$

This quantity can be written as the sum of five different terms as in Section 3.2:

$$y[i] = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk}[i]|^2 g_m[i]}_{\text{signal term}} \tag{3.59a}$$

$$+ \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}(H_{mk}[i])^* H_{m'k}[i] g_{m'}[i]}_{\text{interference term}} \tag{3.59b}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}(H_{mk}[i])^* H_{m'k}[i] Q_{m'}[i]}_{\text{distortion noise term}} \tag{3.59c}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk}[i]|^2 Q_{m'}[i]}_{\text{second type of distortion noise term}} \tag{3.59d}$$

$$+ \underbrace{\frac{1}{(1-\eta)K}\sum_{k=1}^{K}\frac{1}{(1-\eta_k)}\left(\sum_{m=1}^{M}(H_{mk}[i])^*\right)P_k[i],}_{\text{noise term}} \tag{3.59e}$$

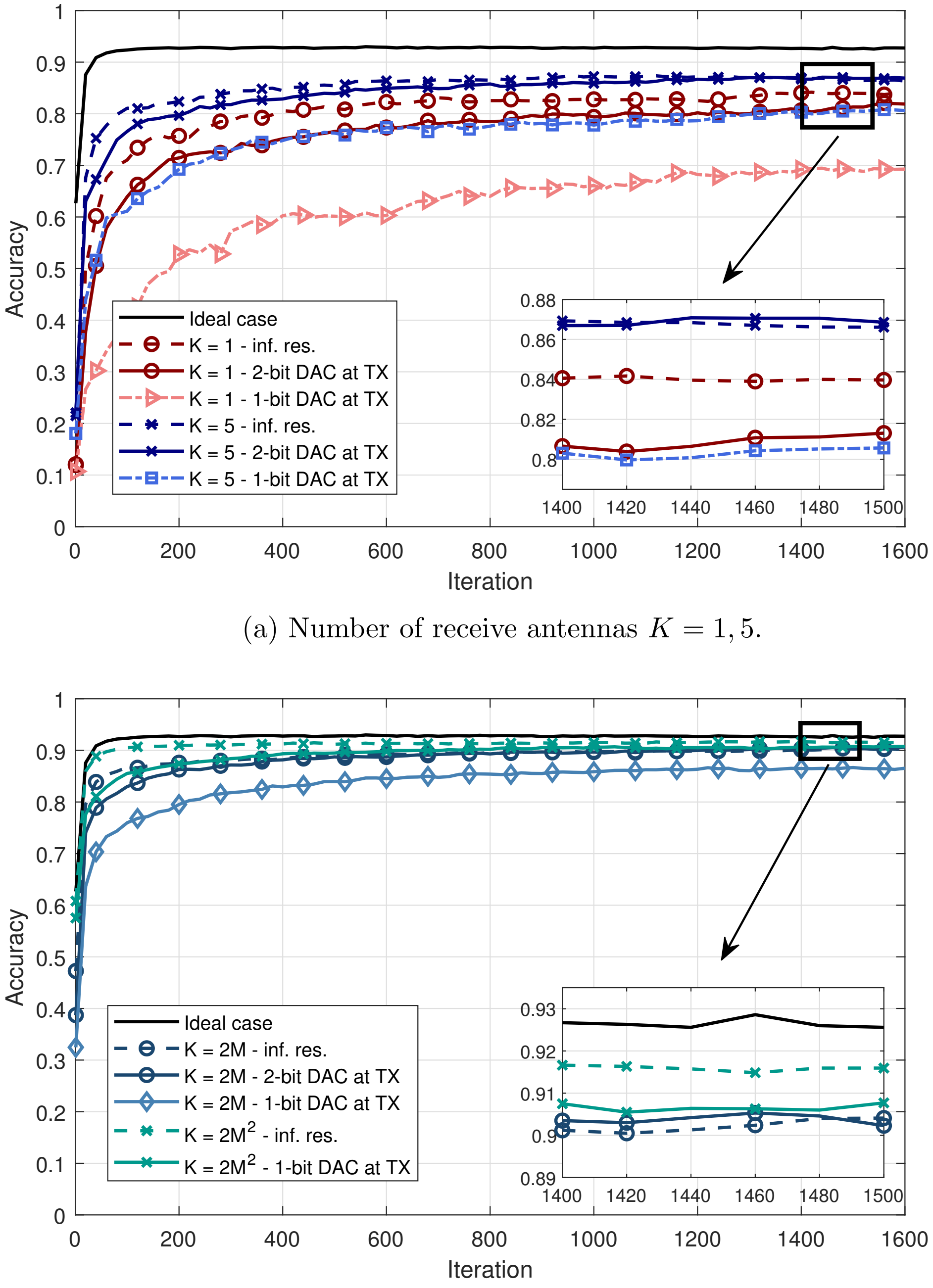
which are the same as the terms given in (3.17) except for the last noise term. As in Section 3.3, the noise term, $P_k[i]$, includes both the channel noise and the quantization noise due to ADCs, and it is with zero mean and finite variance. The analyses of the interference term (3.59b), distortion noise term (3.59c), and the second type of distortion noise term (3.59d) are the same as those of (3.17b), (3.17c), and (3.17d), respectively. Hence, similar arguments on the convergence of the learning algorithm with finite resolution DACs are also valid for the combined effects of DACs and ADCs. In other words, using a sufficiently large number of antennas at the PS, the gradients can be recovered via (3.33). The main conclusion is that we can design a federated learning system with a large number of workers and receive antennas, and still have extremely low hardware cost and energy consumption. This is remarkable since it shows the practicality of the federated learning over realistic wireless channels with very low-cost hardware.

3.5 Numerical Examples

We now evaluate the performance of blind federated learning with realistic channel effects and hardware limitations via simulations. Our main objective is to verify the theoretical expectations on the low-cost federated learning systems over wireless channels via simulations. We use the MNIST dataset [117] with 60000 training and 10000 test samples to train a single layer neural network using the Adam optimizer [118]. At the beginning of the training process, each worker caches $B = 1000$ training samples randomly. The number of parameters is $d = 7850$.

Our system consists of $M = 20$ workers connected to a PS through a multipath fading channel with $L = 3$ taps and $\sigma_{h,l}^2 = 1/L$, hence we have a normalized uniform multipath delay profile where each tap experiences Rayleigh fading. We consider an OFDM setup with $f_c = 3$ GHz carrier frequency, and the number of subcarriers is $N_{\text{cp}} = 4096$ where the subcarrier spacing is $\Delta f = 80$ kHz. We take the sampling period as $T_s = T_w/N$ where $T_w = \frac{1}{\Delta f} = 12.5$ µs is the OFDM word duration without the CP. As given in [119], the maximum delay spread of a typical urban area is 3.5 µs. Consider a wireless network in an urban area where the delay spread is 3.05 µs which is approximately $1000T_s$. We assume that the first tap has no delay and coherence time corresponds to $1000T_s$. Also, time delays are uniformly spaced, i.e., $\tau_{mk1} = 0$, $\tau_{mk2} = 500T_s$, $\tau_{mk3} = 1000T_s$ for $\forall m, k$[1]. The cyclic prefix length is set to $N_{\text{cp}} = 1024$, which is enough to remove the intersymbol interference caused by the multipath. The average transmit power of the OFDM word transmitted by the m-th worker is calculated as $P_T = \frac{1}{T}\sum_{t=1}^{T} \left|\left|\bar{\mathbf{G}}_m^t\right|\right|_2^2$, which gives $P_T = 1.3267\times 10^{-4}$ for this setup, where T is the total iteration count. In our theoretical analysis, we model the OFDM words with the autocorrelation matrix $\mathbf{C}_{\bar{\mathbf{G}}_m\bar{\mathbf{G}}_m}$ with equal nonzero diagonal elements denoted by σ_G^2, and zero off-diagonal elements. In our simulations, we do not make any assumption on the statistics of the gradients; we simply use the estimates through the realistic channel simulations.

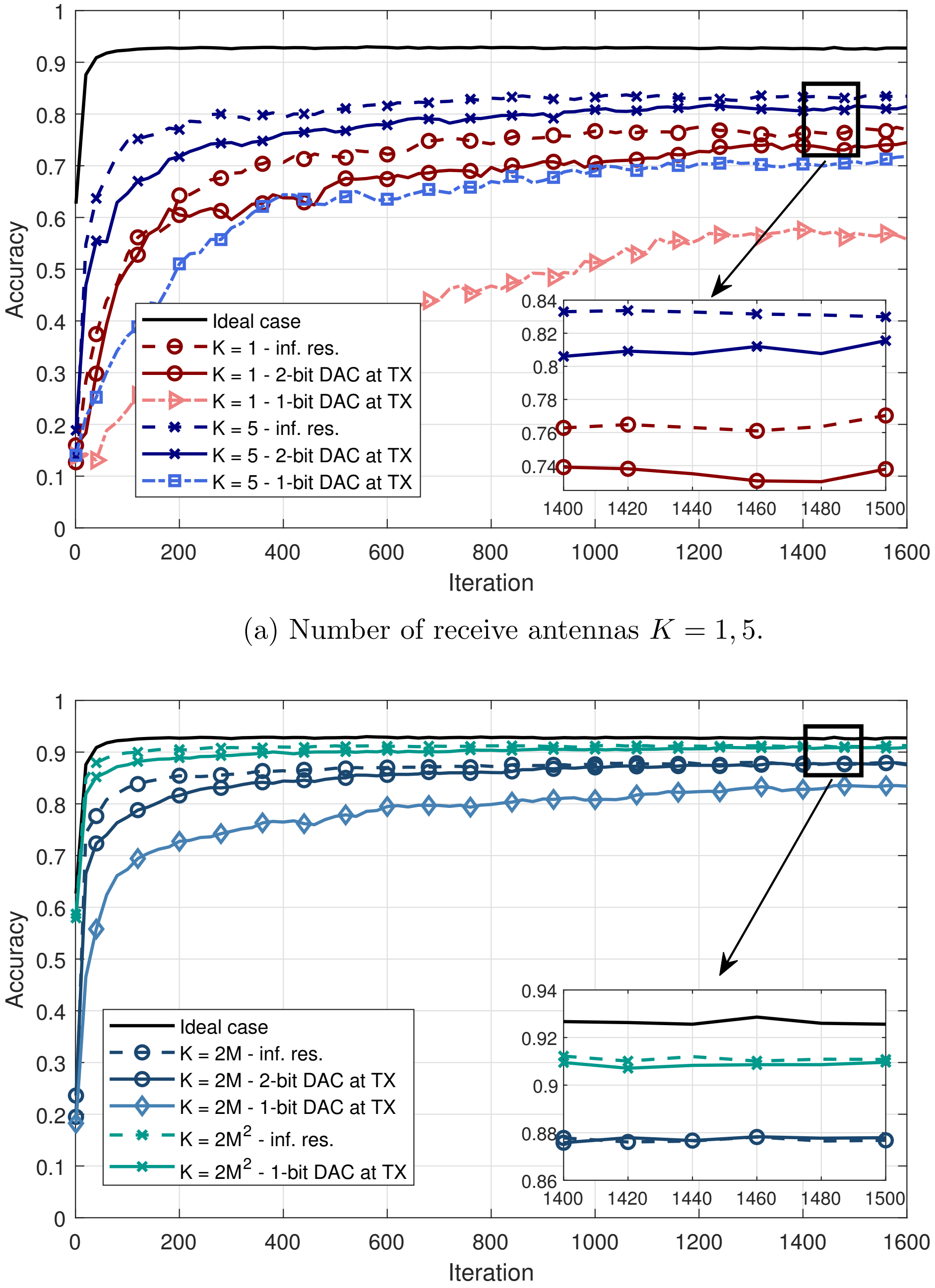
[1] We select this multipath delay profile for ease of illustration. More realistic multipath delay profiles, e.g., exponential delay profiles, can be selected, but doing so will not change our main conclusions.



(a) Number of receive antennas $K = 1, 5$.

(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.3: Test accuracy of the system with low-resolution DACs for channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$.



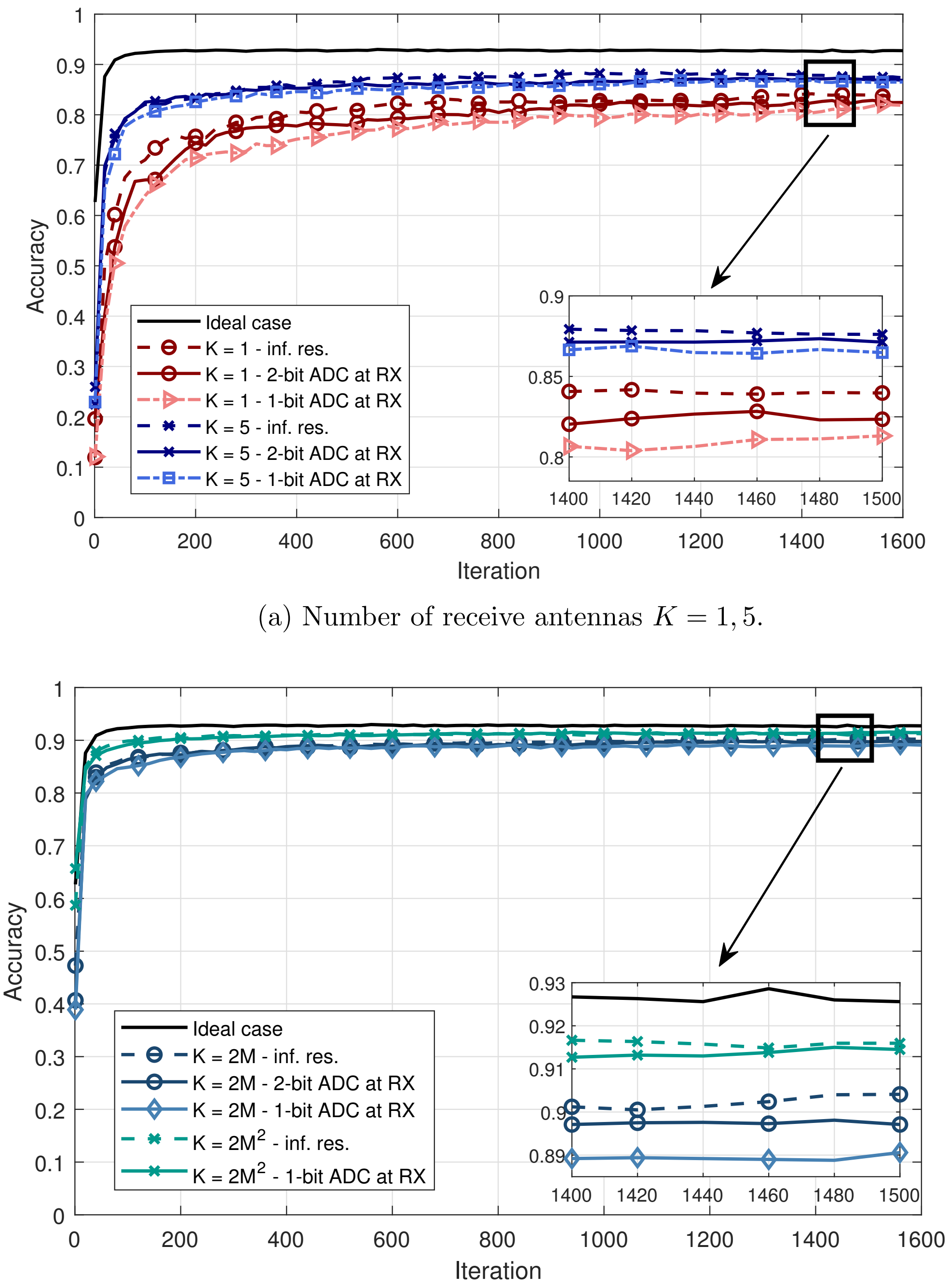
(a) Number of receive antennas $K = 1, 5$.

(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.4: Test accuracy of the system with low-resolution DACs for channel noise variance $\sigma_z^2 = 4 \times 10^{-3}$.

In Figs. 3.3a and 3.3b, the test accuracy for a system where each worker is equipped with a low-resolution DAC and different number of antennas $K \in \{1,\ 5,\ M,\ 2M^2\}$ at the receiver side is illustrated for $\sigma_z^2 = 8\times 10^{-4}$. As the number of receive antennas increases, the test accuracy approaches that of the infinite resolution case since the variance of the distortion noise and interference decrease. At iteration $T = 1600$, the accuracy loss with one-bit DAC compared to infinite resolution case is 17.62%, 6.62%, 4.07%, and 0.37% for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively. Furthermore, the low complexity system achieves almost the same accuracy with the infinite resolution case when two-bit DACs are employed (except for $K = 1$ which has an accuracy loss of 2.64%). In Figs. 3.4a and 3.4b, we increase the channel noise variance to $\sigma_z^2 = 4 \times 10^{-3}$, i.e., there is a 14 dB signal-to-noise ratio (SNR) reduction. Since the effect of the noise term is increased as expected, the performance of the learning algorithm deteriorates. However, as shown in Figs. 3.4a and 3.4b, the convergence is still achieved, and the accuracy loss of the one-bit DAC case compared to infinite resolution case is 27.54%, 13.95%, 4.71%, and 0.8% for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively. With two-bit DACs, the accuracy loss decreases to 3.26% and 2.40% for $K = 1$ and $K = 5$, respectively, while it gives almost the same performance when the number of PS antennas is $K = 2M$ and $K = 2M^2$. These results clearly illustrate that when a moderate number of receive antennas are employed, low-resolution, even two-bit, DACs can achieve a learning performance comparable with the infinite resolution case.

In Figs. 3.5a and 3.5b, the test accuracy for different number of PS antennas $K \in \{1,\ 5,\ M,\ 2M^2\}$ each equipped with a low-resolution ADC is illustrated for a system with $\sigma_z^2 = 8 \times 10^{-4}$, and compared with the error-free shared link case. As expected, using higher number of receive antennas results in an improved learning accuracy. Indeed the results are very close to those of the infinite resolution case, especially with two-bit ADCs, while there is a minor drop on accuracy with one-bit ADCs. For instance, after the 1600-th iteration, using one-bit ADCs causes only 2.64%, 0.95%, and 0.13%, accuracy loss compared to infinite resolution case for $K = 1$, $K = 5$, and $K = 2M$, respectively.



(a) Number of receive antennas $K = 1, 5$.

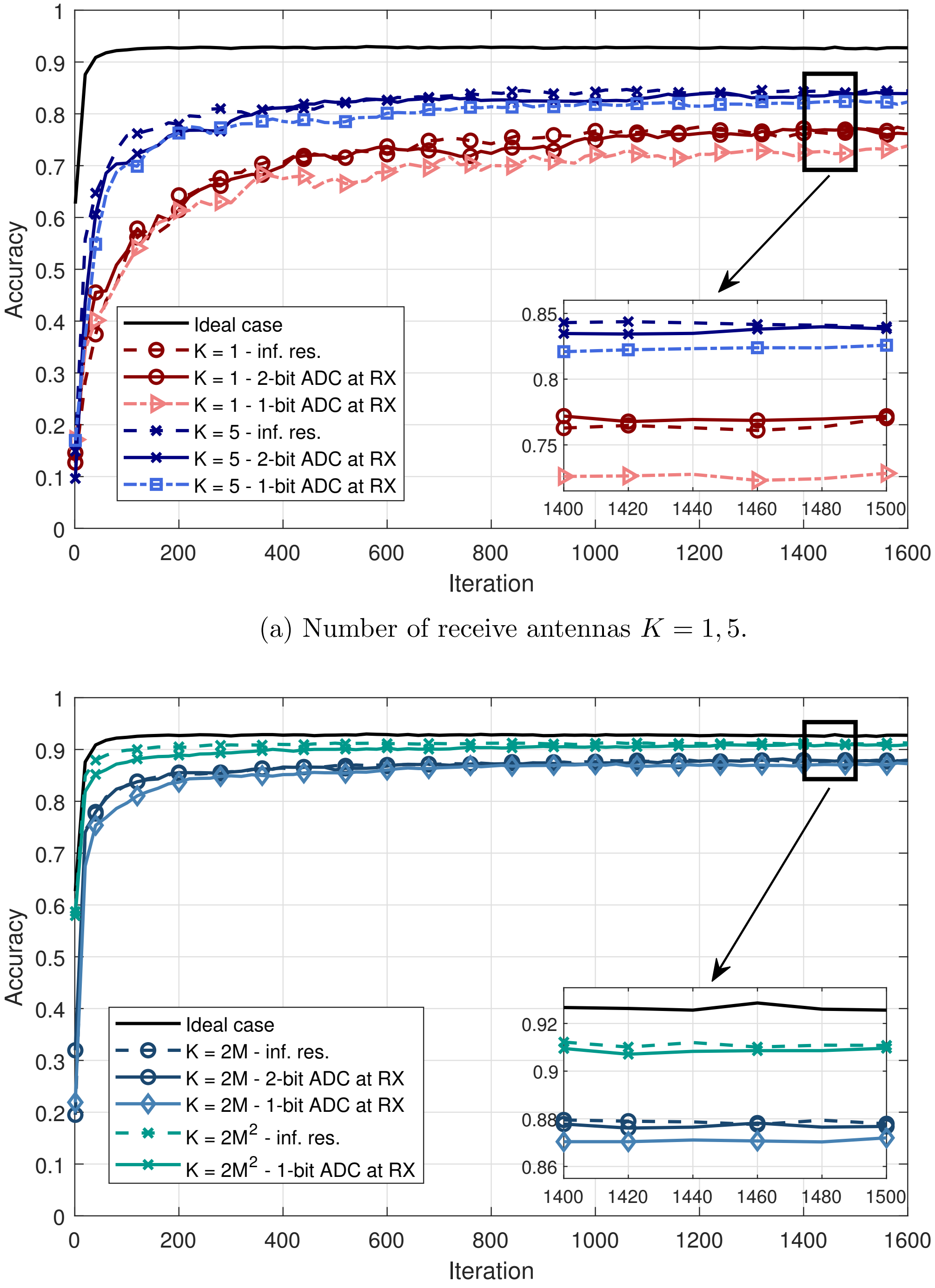
(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.5: Test accuracy of the system with low-resolution ADCs for channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$.

Furthermore, the system achieves the performance of the infinite resolution scenario with $K = 2M^2$ PS antennas. These results are due to the fact that increasing the number of antennas reduces the interference dramatically which makes the combined signal a very good estimate of the gradient vector, even with low-resolution ADCs.

Without changing any other parameters of the setup described above, we increase the noise variance to $\sigma_z^2 = 4 \times 10^{-3}$ in Figs. 3.6a and 3.6b. As in the previous case, for the two-bit ADC case, the performance of the proposed scheme is very close to the error-free case for a large number of receive antennas. When the number of antennas is decreased, with the detrimental effects of the channel noise and interference caused by the multipath fading channel, the accuracy decreases. However, even for this high level of channel noise, using one-bit ADCs causes only 4.09%, 2.55%, 0.37%, and 0.32% accuracy loss compared to the infinite resolution case for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively, after the 1600-th iteration.

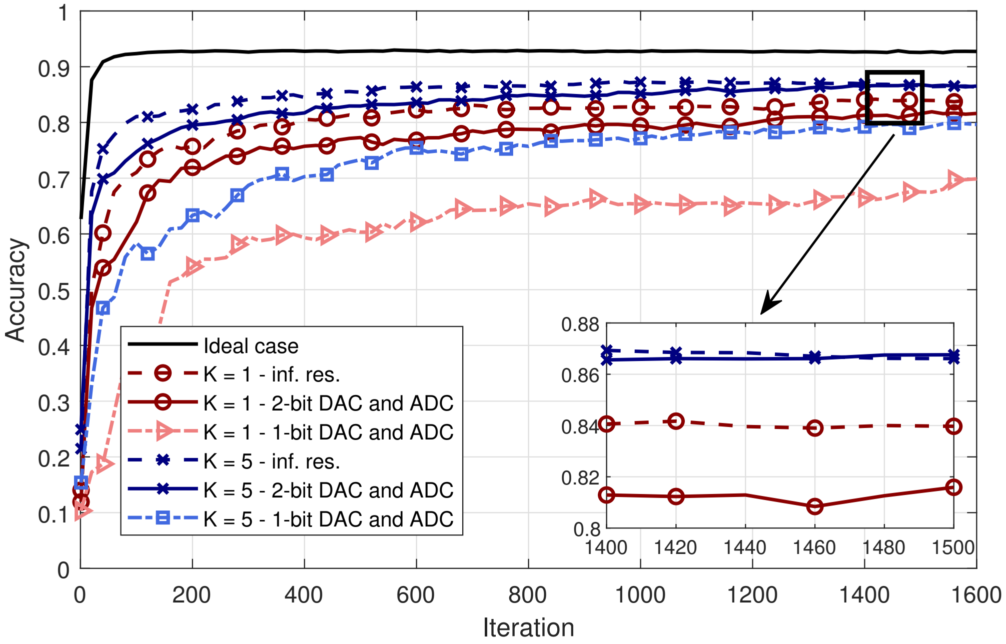
In Figs. 3.7a and 3.7b, we consider a system which employs both low-resolution DACs at the workers and one-bit ADCs at the PS antennas with channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$. As expected, using low-resolution DAC and ADC at the same time increases the amount of interference in the gradient estimates at the PS, which decreases the learning accuracy of the distributed system. However, the combined effect of the interference terms is still negligible, especially for sufficiently large number of receive antennas. After the 1600-th iteration, the use of one-bit DACs and ADCs simultaneously causes only 17.91%, 7.76%, 4.18%, and 0.39% accuracy loss compared to the infinite resolution case for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively. When $K = 1$, using two-bit DACs and ADCs results in a 2.95% accuracy loss while the performance is almost the same as that of the infinite resolution case when the number of PS antennas is higher. In the same system, we increase the channel noise variance to $\sigma_z^2 = 4 \times 10^{-3}$, and show the corresponding results in Figs. 3.8a and 3.8b. We observe that increasing the noise level causes 28.56%, 15.66%, 6.87%, and 1.91% accuracy loss compared to the infinite resolution case for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively after the 1600-th iteration (with one-bit DACs and ADCs).



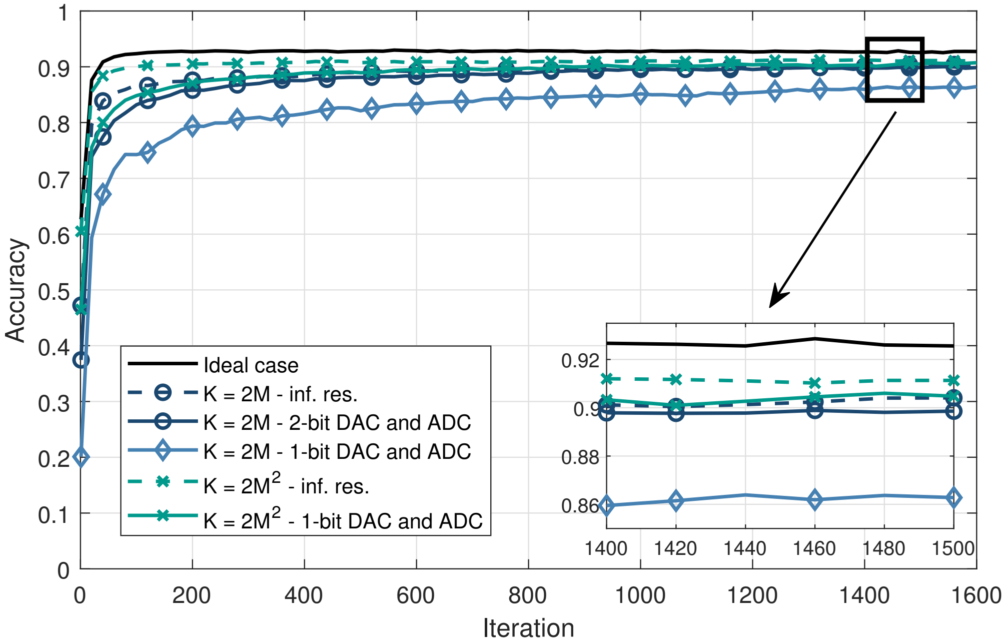
(a) Number of receive antennas $K = 1, 5$.

(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.6: Test accuracy of the system with low-resolution ADCs for channel noise variance $\sigma_z^2 = 4 \times 10^{-3}$.

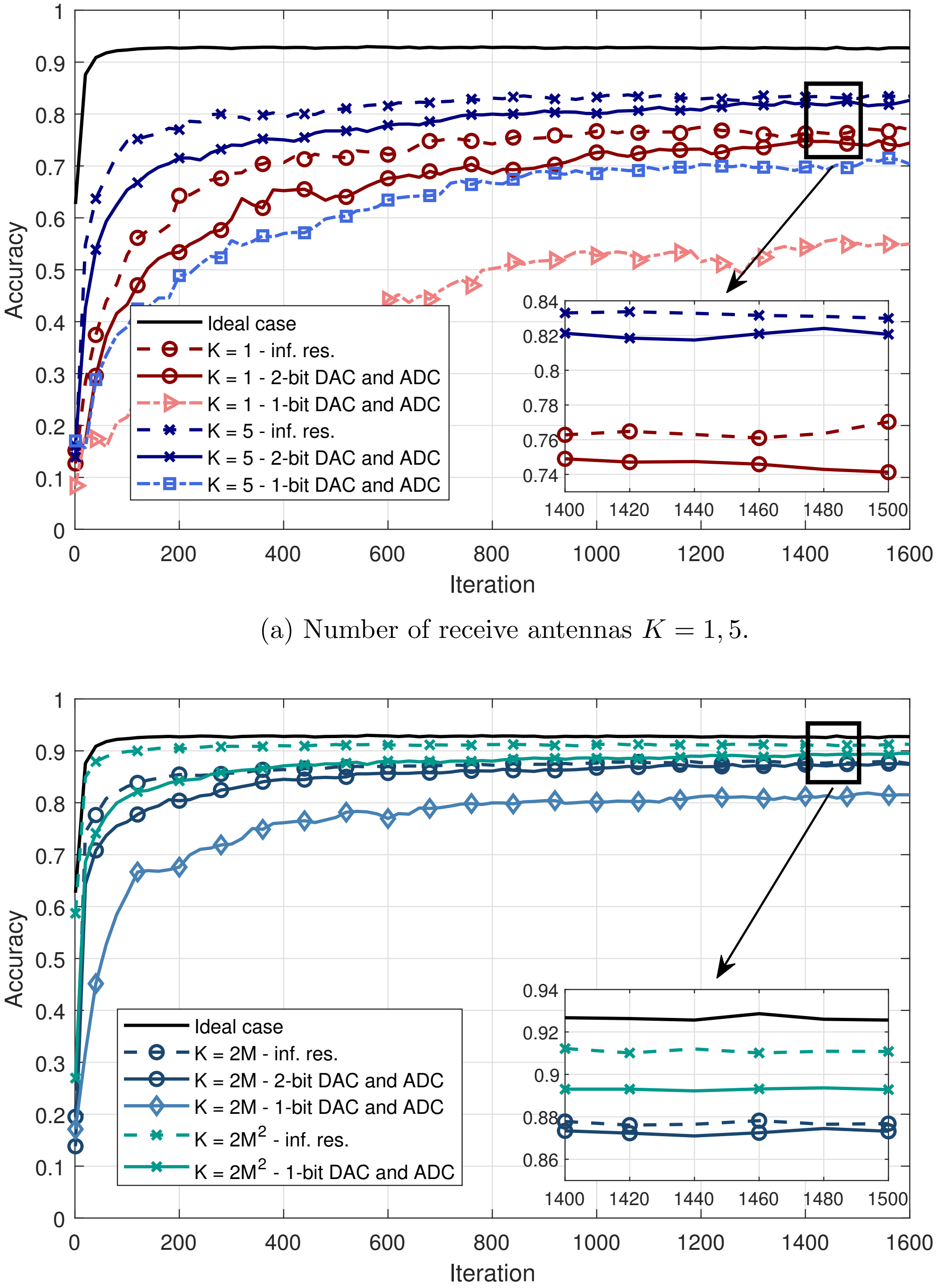


(a) Number of receive antennas $K = 1, 5$.



(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.7: Test accuracy of the system with low-resolution DACs and ADCs for channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$.



(a) Number of receive antennas $K = 1, 5$.

(b) Number of receive antennas $K = 2M, 2M^2$.

Figure 3.8: Test accuracy of the system with low-resolution DACs and ADCs for channel noise variance $\sigma_z^2 = 4 \times 10^{-3}$.

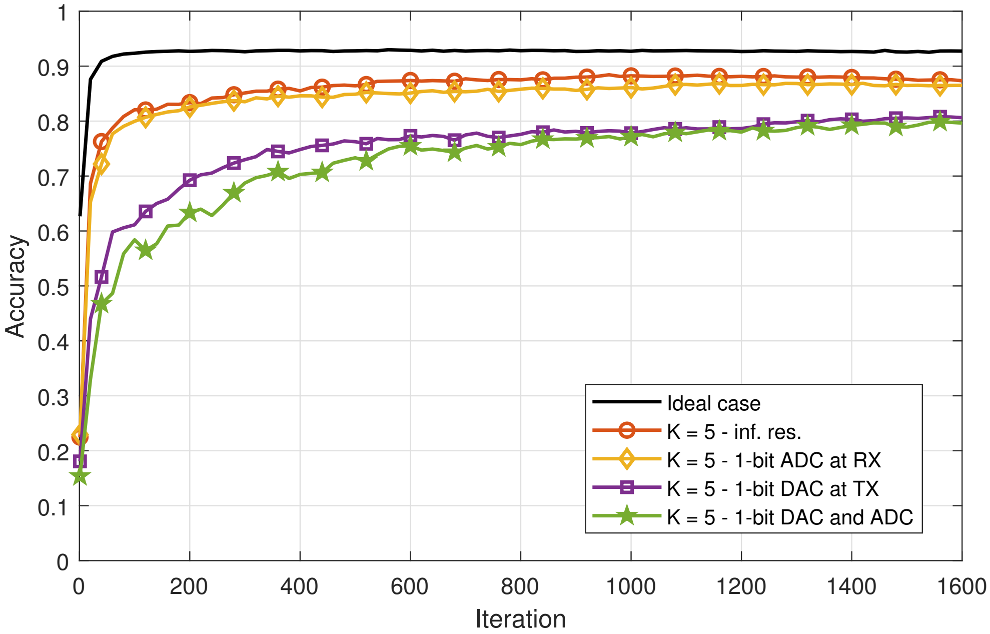


Figure 3.9: Test accuracy of the system with seperate one-bit DACs at the workers, one-bit ADCs at the PS antennas, and joint DACs and ADCs where the channel noise variance is $\sigma_z^2 = 8 \times 10^{-4}$, and $K = 5$.

Finally, in Fig. 3.9, we compare the effect of one-bit quantization on the transmitter and receiver sides, both separately and jointly, with a fixed number of receive antennas $K = 5$. As expected, the test accuracy of the system with one-bit DAC workers and infinite resolution ADCs at the PS is lower than that for the case of infinite resolution DACs at the workers and one-bit ADCs at the PS. This is because, using DACs at the workers results in higher interference than using ADCs at the PS, and the performance is deteriorated. However, the convergence of the learning algorithm is preserved. Another important implication of our results is that even though our analysis is based on a certain assumption on the statistics of the gradients, the simulation results (which are obtained without using the Gaussian assumption on the OFDM words) are consistent with our theoretical expectations. Hence, with a slight sacrifice on the accuracy rate of the learning algorithm, power and hardware efficient systems (at both transmitter and receiver sides) can be designed and implemented for distributed learning at the wireless edge over realistic wireless channels.

3.6 Chapter Summary

We have investigated blind federated learning at the wireless edge with OFDM based transmission and low-resolution, even one-bit, DACs and ADCs at the transmitter and receiver sides, respectively, for a practical and inexpensive system design, and reduced power consumption. Our analytical results illustrate that with low-resolution DACs at the transmitter and ADCs at the receiver, the convergence of the distributed learning algorithms based on SGD is guaranteed when the number of receive antennas is increased as in the ideal case of infinite resolution DACs and ADCs. Moreover, the convergence is still attained with the joint use of DACs and ADCs which reduces the implementation costs further. The results are also valid for the extreme case of one-bit DACs and ADCs. Through extensive numerical examples, it is also illustrated that using a moderate number of antennas with low-resolution DACs and ADCs, e.g., using 5 antennas at the PS, can closely approach the performance of the infinite resolution case. It is also observed that, in case of low channel noise, the learning performance is decreased only slightly even for the extreme case of one-bit ADCs and DACs.

Chapter 4

Federated Learning with Over-the-Air Aggregation over Time-Varying Channels

In this chapter, we investigate the effects of the mobility of workers and/or the PS on the performance of FL systems over wireless channels with OTA aggregation. We model the channel between the workers and the PS as a time-varying multipath MAC, and assume that the local gradients are transmitted via OFDM to mitigate the frequency selectivity of the channel. The relative motion of the workers and/or the PS as well as the other variations in the environment induce time variations in the channel resulting in Doppler shifts/spreads. These variations destroy the orthogonality among the subcarriers causing ICI, which deteriorates the system performance. In our context, these time variations result in interference among the elements of the gradient vector received at the PS. Our primary focus is the analysis of the interference terms in the received signal due to multiple user transmissions and channel variations, and the investigation of the convergence of the FL algorithm.

Our main contributions are as follows:

- Different from the existing studies on FL, we model the wireless channel between the workers and PS as a time-varying multipath fading channel and employ a practical OFDM-based transmission scheme to address the effects of time variations on FL performance. This modeling is crucial for our setup because using time-varying channels enables us to assess the limitations of federated learning when there is worker or PS mobility, as well as when there are other variations in the transmission medium. Our results show that FL systems can still be used under these practical constraints, especially, when the time variations are not excessive.

- We perform a convergence analysis for the FL with OTA aggregation over time-varying channels. Even though there are studies in the current literature on the convergence of FL with OTA aggregation, they mainly focus on user selection, resource allocation, blind workers, hierarchical learning, etc., see [39, 76, 68, 33, 74]. There are no results which consider the effects of time-varying links on federated learning. With this motivation, we explicitly study the effects of the amount of time variations and Doppler spread on the convergence rate of the FL with OTA aggregation, and illustrate that the convergence is preserved, especially for small to moderate amount of time variations, which are typical for wireless communication systems.

- We further validate our theoretical expectations on FL over time-varying channels through extensive simulations with MNIST and CIFAR-10 datasets with both i.i.d. and non-i.i.d. data.

Table 4.1: Summary of variables.

Variable	**Value**
M	Number of workers, m and m_i are used as the worker index
K	Number of PS antennas, k and k_i are used for the antenna index
L_{tap}	Number of taps in multipath channel, l and l_i are used for tap index
$\mathcal{B}_m$, B_m, B	Local dataset of the m-th worker, its size, overall data size, respectively
t, ϵ	Global iteration index and number of local iterations, respectively
$\eta_m^p(t)$	Learning rate of the m-th worker during the p-th local iteration of the t-th global iteration
$h_{mk,n}^{e,\lambda}(t)$	Time-varying impulse response for the t-th global iteration and the e-th OFDM word where n is the time index, λ is the delay variable, m and k are the worker and antenna indices, respectively
$h_{mkl,n}^{e}(t)$	Gain of the l-th lap of the time varying channel
σ_h^2	Variance of $h_{mkl,0}^{e}(t)$
α	Parameter of the auto-regressive time-varying channel model and represents the dependence between the consecutive samples
$\Delta\boldsymbol{\theta}(t)$	Average of all local updates
$\Delta\boldsymbol{\theta}_m^e(t)$	The e-th part of the m-th worker's local update
$\Delta\boldsymbol{\theta}_m(t)$	Local update of the m-th worker
$\Delta\boldsymbol{\vartheta}_m^e(t)$	The e-th OFDM word for the m-th worker's local update
σ_z^2	Variance of the channel noise where the variance of its DFT is σ_Z^2
$z_{k,i}^e(t)$	The i-th entry of the channel noise vector for the k-th PS antenna and e-th OFDM word during iteration t whose DFT is denoted by $Z_{k,u}^e(t)$
N_{sc}	Number of OFDM subcarriers
$H_{mk,uv}^e(t)$	The frequency response of the channel from the m-th worker to the k-th PS antenna

The chapter is organized as follows. Section 4.1 introduces the system model and preliminaries. DSGD over time varying-channels is analyzed in Section 4.2. A convergence analysis of FL over time-varying channels is provided in Section 4.3, while its performance is studied via simulations in Section 4.4. The chapter is concluded in Section 4.5.

Notation: Throughout this chapter, we use the notation $[a\ b]$ to indicate the integer set $\{a, \ldots, b\}$ where $a \leq b$, a and b are positive integers, and $[b] = [1\ b]$. We denote l_2 norm of a vector $\mathbf{x}$ by $\|\mathbf{x}\|_2$. A summary of the notation used throughout the manuscript is provided in Table 4.1.

4.1 System Model and Preliminaries

We consider a federated learning system with M workers, each with its own local dataset $\mathcal{B}_m$ with size B_m, $\forall m \in [M]$. We denote the amount of data in the overall system by B, i.e., $B \triangleq \sum_{m=1}^{M} B_m$. For the learning process, SGD is implemented where the local gradients calculated by each worker are sent to a PS with OTA aggregation through a time-varying multipath fading MAC with OFDM as shown in Fig. 4.1. Since this is an over-the-air federated learning system, the PS receives the superposition of the local gradients from all the M users which have the same computing power.

Since providing transmit side CSI requires significant additional overhead, we adopt the more practical scheme of blind transmitters (no CSI at the transmitter), and perfect CSI at the receiver (CSIR). We also equip the PS with multiple antennas, which are used to align the received signals (see [120, 39]), while the workers have only one antenna. Note that one may consider resource allocation and client selection for practical scenarios if the CSI is available at the worker side; however, since our primary focus is the effect of channel time variations on FL systems, these aspects and their effects jointly with time variations are left for future research.

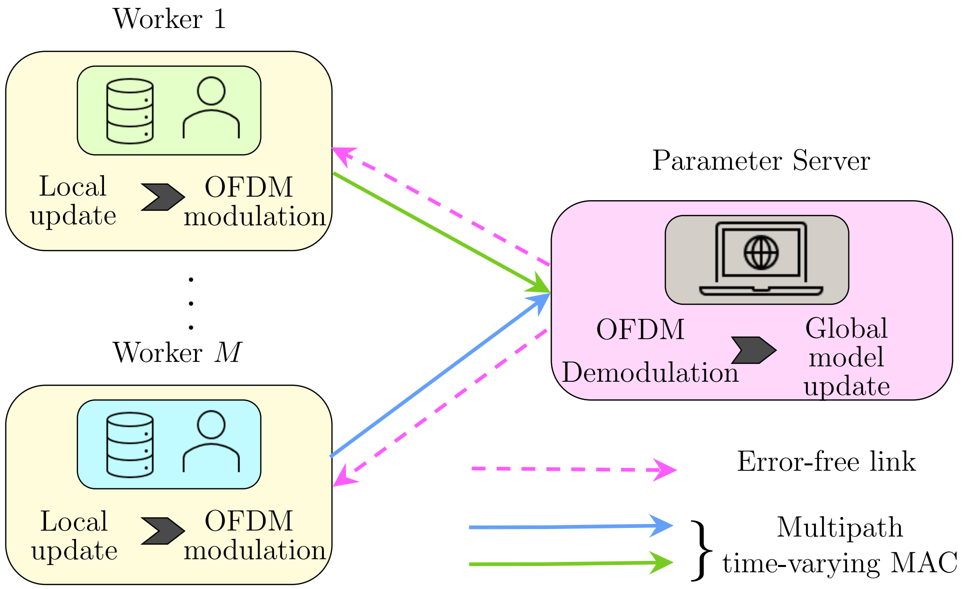


Figure 4.1: System model for federated learning at the wireless edge.

The PS combines the received signals at the K receive antennas, performs OFDM demodulation, and updates the global model parameter which is then broadcast to the workers after each iteration over an error-free shared link.

We consider the overall loss as

$$F(\boldsymbol{\theta}) = \sum_{m=1}^{M} \frac{B_m}{B} F_m(\boldsymbol{\theta}), \tag{4.1}$$

where $F_m(\boldsymbol{\theta})$ is the empirical loss function at the m-th worker for $m \in [M]$.

At the beginning of each global iteration t, the parameter server transmits the global parameter $\boldsymbol{\theta}(t)$ to all the workers over an error-free link. After receiving the global parameter, the m-th worker performs ϵ local iterations as described in the following update rule:

$$\boldsymbol{\theta}_m^{p+1}(t) = \boldsymbol{\theta}_m^{p}(t) - \eta_m^p(t) \nabla F_m\left(\boldsymbol{\theta}_m^p(t), \xi_m^p(t)\right), \tag{4.2}$$

for $p \in [\epsilon]$ where $\eta_m^p(t)$ is the learning rate for the m-th worker during the p-th local iteration of the t-th global iteration, $\nabla F_m\left(\boldsymbol{\theta}_m^p(t), \xi_m^p(t)\right)$ is the stochastic gradient estimate, $\xi_m^p(t)$ is the local mini-batch sample. The local iterations are initialized as $\boldsymbol{\theta}_m^1(t) \ = \boldsymbol{\theta}(t)$.

At the end of each iteration t, after performing ϵ local updates, worker m transmits its update to the PS, that is,

$$\Delta\boldsymbol{\theta}_m(t) = \boldsymbol{\theta}_m^{\epsilon+1}(t) - \boldsymbol{\theta}(t), \tag{4.3}$$

for $m \in [M]$. Without any noise or interference in the system, the ideal update at the PS is $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \Delta\boldsymbol{\theta}(t)$, where $\Delta\boldsymbol{\theta}(t)$ is the average of all the local updates, that is,

$$\Delta\boldsymbol{\theta}(t) = \frac{1}{M}\sum_{m=1}^{M} \Delta\boldsymbol{\theta}_m(t). \tag{4.4}$$

We consider FL over time-varying multipath fading channels with OTA aggregation via OFDM. Hence, we aim to obtain an estimate for $\Delta\boldsymbol{\theta}(t)$. For this purpose, we divide the local updates $\Delta\boldsymbol{\theta}_m \in \mathbb{R}^d$ into $E = \left\lceil \frac{d}{2N_{sc}} \right\rceil$ parts each of which is transmitted as a separate OFDM word using N_{sc} subcarriers, i.e., we form

$$\Delta\boldsymbol{\theta}_m^{e,\mathrm{re}}\left(t\right) \triangleq [\Delta\theta_{m,2(e-1)N_{sc}+1}\left(t\right), \dots, \Delta\theta_{m,(2e-1)N_{sc}}\left(t\right)]^T, \tag{4.5a}$$

$$\Delta\boldsymbol{\theta}_m^{e,\mathrm{im}}\left(t\right) \triangleq [\Delta\theta_{m,(2e-1)N_{sc}+1}\left(t\right), \dots, \Delta\theta_{m,2eN_{sc}}\left(t\right)]^T, \tag{4.5b}$$

$$\Delta\boldsymbol{\theta}_m^{e}\left(t\right) \triangleq \Delta\boldsymbol{\theta}_m^{e,\mathrm{re}}\left(t\right) + j\Delta\boldsymbol{\theta}_m^{e,\mathrm{im}}\left(t\right), \tag{4.5c}$$

for $e \in [E]$ where $\Delta\boldsymbol{\theta}_m\left(t\right)$ is zero-padded to satisfy the vector length of $2N_{sc}E$, and the i-th entry of vector $\Delta\boldsymbol{\theta}_m^e\left(t\right)$ is denoted by $\Delta\theta_{m,i}^e\left(t\right)$ for $i \in [N_{sc}]$ which is equal to

$$\Delta\theta_{m,i}^e\left(t\right) = \Delta\theta_{m,2(e-1)N_{sc}+i}\left(t\right) + j\Delta\theta_{m,(2e-1)N_{sc}+i}\left(t\right). \tag{4.6}$$

To generate the OFDM signal corresponding to $\Delta\boldsymbol{\theta}_m\left(t\right)$, we take an N_{sc}-point IDFT of the gradient vector $\boldsymbol{\Delta\vartheta}_m^e(t) \in \mathbb{C}^{N_{sc}}$ whose u-the element is obtained as

$$\Delta\vartheta_{m,u}^e(t) = \frac{1}{N_{sc}}\sum_{n=1}^{N_{sc}} \Delta\theta_{m,n}^e\left(t\right) e^{j2\pi nu/N_{sc}}, \tag{4.7}$$

for $u \in [N_{sc}]$. To combat the multipath effects, we add a length-N_{cp} cyclic prefix, i.e., we transmit the OFDM word

$$\begin{aligned}\boldsymbol{\Delta}\bar{\boldsymbol{\vartheta}}_m^e(t) = \big[&\Delta\vartheta_{m,N_{sc}-N_{cp}+1}^e(t), \dots,\ \Delta\vartheta_{m,N_{sc}}^e(t), \\ &\Delta\vartheta_{m,1}^e(t),\ \dots,\ \Delta\vartheta_{m,N_{sc}}^e(t)\big]^T,\end{aligned} \tag{4.8}$$

corresponding to the local gradient of the m-th worker. Note that in some applications, due to power limitations or fading effects, some workers may not be able to complete or transmit their local updates. However, since our primary focus is the investigation of time-varying channels on FL with blind workers, we assume that all the workers perform the local update and have enough resources to transmit their updates at every iteration. As an extension, if the CSI is available at the workers, this setup can be extended and user scheduling and resource allocation schemes can be considered to deal with limited power budget and deep fading.

We assume that the maximum delay of the multipath channel is less than the cyclic prefix duration, hence there is no intersymbol interference among consecutive OFDM words.

The time-varying impulse response of the channel for the t-th global iteration of the algorithm is given by

$$h_{mk,n}^{e,\lambda}(t) = \sum_{l=1}^{L_{tap}} h_{mkl,n}^e(t)\delta[\lambda - \tau_{mkl}^e(t)], \tag{4.9}$$

for the e-th OFDM word where $n \in [N + N_{cp}]$ is the time index, λ is the delay variable, L_{tap} is the number of channel taps, $h_{mkl,n}^e(t) \in \mathbb{C}$ is the gain of the l-th channel tap from the m-th worker to the k-th antenna of the PS, and $\tau_{mkl}^e(t)$ is the corresponding time delay. We model the time-varying channel taps as

$$h_{mkl,n}^e(t) = \sqrt{1-\alpha^2} \cdot h_{mkl,n-1}^e(t) + \alpha \cdot c, \tag{4.10}$$

for $n \in [N + N_{cp}]$ where $0 \leq \alpha \leq 1$, c, and $h_{mkl,0}^e(t)$ are independent complex random variables with zero mean and variance σ_h^2. In this model, α is the parameter of the auto-regressive channel model and represents the dependence between the consecutive samples. Clearly, larger values of α correspond to faster channel variations.

At the k-th receive chain, after removing the CP, the i-th entry of the received vector during iteration t is written as

$$y_{k,i}^e(t) = \sum_{m=1}^{M} \sum_{l=1}^{L_{tap}} h_{mkl,i}^e(t) \Delta\vartheta_{m,i-\tau_{mkl}^e(t)}^e(t) + z_{k,i}^e(t), \tag{4.11}$$

where $i \in [N_{sc}]$, $z_{k,i}^e(t) \in \mathbb{C}$ are the additive noise terms, which are i.i.d., and $z_{k,i}^e(t) \sim \mathcal{CN}(0, \sigma_z^2)$ for $k \in [K]$. Estimate of the average of the local updates $\Delta\boldsymbol{\theta}(t)$ is denoted by $\Delta\hat{\boldsymbol{\theta}}(t)$, and obtained by processing (4.11) for $\forall e \in [E]$. As a result, the PS updates the global parameter by

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \Delta\hat{\boldsymbol{\theta}}(t), \tag{4.12}$$

which is shared with all the workers for the next iteration $t+1$.

4.2 DSGD over Time-Varying Channels

4.2.1 Signal Combining for Time-Varying Channels

After removing the CP at the receiver side, we take an N_{sc}-point DFT to obtain the frequency domain signal given by

$$\begin{aligned} Y_{k,u}^e(t) &= \sum_{i=1}^{N_{sc}} y_{k,i}^e(t) \cdot e^{-j\frac{2\pi ui}{N_{sc}}} \\ &= \sum_{i=1}^{N_{sc}} \left(\sum_{m=1}^{M} \sum_{l=1}^{L_{tap}} h_{mkl,i}^e(t) \vartheta_{m,i-\tau_{mkl}^e(t)}^e(t) \right) \cdot e^{-j\frac{2\pi ui}{N_{sc}}} + Z_{k,u}^e(t), \end{aligned} \tag{4.13}$$

where $u \in [N_{sc}]$, and $Z^e_{k,u}(t)$ is the DFT of $z^e_{k,i}(t)$, i.e., $Z^e_{k,u}(t) = \sum_{i=1}^{N_{sc}} z^e_{k,i}(t) \cdot e^{-j\frac{2\pi ui}{N_{sc}}}$. Also, we have

$$\begin{aligned} \vartheta^e_{m,i-\tau^e_{mkl}}(t) &= \text{IDFT}\left(\Delta\theta^e_{m,v}(t) \cdot e^{-j\frac{2\pi v\tau^e_{mkl}(t)}{N_{sc}}}\right) \\ &= \frac{1}{N}\sum_{v=1}^{N_{sc}} \Delta\theta^e_{m,v}(t) \cdot e^{j\frac{2\pi v(i-\tau^e_{mkl}(t))}{N_{sc}}}. \end{aligned} \tag{4.14}$$

Inserting (4.14) into (4.13), we get

$$\begin{aligned} Y^e_{k,u}(t) = \frac{1}{N_{sc}}\sum_{v=1}^{N_{sc}}\sum_{m=1}^{M} \Delta\theta^e_{m,v}(t) \cdot \left(\sum_{l=1}^{L_{tap}}\left[\sum_{i=1}^{N_{sc}} h^e_{mkl,i}(t) \cdot e^{-j\frac{2\pi i(u-v)}{N_{sc}}}\right] e^{-j\frac{2\pi v\tau^e_{mkl}(t)}{N_{sc}}}\right) \\ + Z^e_{k,u}(t). \end{aligned} \tag{4.15}$$

Now, we can divide (4.15) into three parts as

$$\begin{aligned} Y^e_{k,u}(t) = &\underbrace{\sum_{m=1}^{M} H^e_{mk,uu}(t)\Delta\theta^e_{m,u}(t)}_{\text{desired term where } u=v} \\ &+ \underbrace{\sum_{\substack{v=1 \\ v\neq u}}^{N_{sc}}\sum_{m=1}^{M} H^e_{mk,uv}(t)\Delta\theta^e_{m,v}(t)}_{\text{ICI term where } u\neq v} + \underbrace{Z^e_{k,u}(t),}_{\text{channel noise}} \end{aligned}$$

where

$$H^e_{mk,uv}(t) = \frac{1}{N_{sc}}\sum_{l=1}^{L_{tap}}\sum_{i=1}^{N_{sc}} h^e_{mkl,i}(t) \cdot e^{-j\frac{2\pi\left(i(u-v)+v\tau^e_{mkl}(t)\right)}{N_{sc}}}, \tag{4.16}$$

is the frequency response of the channel from m-th worker to the k-th PS antenna. Note that, when $\alpha = 0$, the channel is time-invariant and $H^e_{mk,uv}(t) = \sum_{l=1}^{L_{tap}} h^e_{mkl,0}(t)e^{-j\frac{2\pi v\tau^e_{mkl}(t)}{N_{sc}}}\delta[u-v]$. That is, in this case there is no ICI.

Combining the frequency domain received signals from the K receive chains as in [120], we have

$$Y_u^e(t) = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{m=1}^{M} H_{mk,uu}^e(t)\right)^* Y_{k,u}^e(t), \tag{4.17}$$

where the mean of $H_{mk,uu}^e(t)$ is zero and the variance of $H_{mk,uu}^e(t)$, denoted by σ_H^2, is

$$\sigma_H^2 = \frac{1}{{N_{sc}}^2}\sum_{l_1,l_2\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\mathbb{E}\left[\left(h_{mkl_1,i_1}^e(t)\right)^* h_{mkl_2,i_2}^e(t)\right]e^{jx_1}. \tag{4.18}$$

where $x_1 = 2\pi u\left(\tau_{mkl_1}^e(t) - \tau_{mkl_2}^e(t)\right)/N_{sc}$. As shown in [10],

$$\sigma_H^2 = \frac{L_{tap}}{{N_{sc}}^2}\left(\frac{N_{sc}(1+r)}{1-r} - 2r\frac{1-r^{N_{sc}}}{(1-r)^2}\right)\sigma_h^2, \tag{4.19}$$

where $r = \sqrt{1-\alpha^2}$ for $0 < r < 1$.

Clearly, for $0 < r < 1$, $\sigma_H^2 = \mathcal{O}(1/N_{sc})\sigma_h^2$. For the time invariant case ($\alpha = 0$), $r = 1$, and $\sigma_H^2 = L_{tap}\sigma_h^2$.

Using (4.19), received signals from K receive chains (4.17) can be decomposed as

$$Y_u^e(t) = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}|H_{mk,uu}^e(t)|^2\cdot\Delta\theta_{m,u}^e(t)}_{\text{signal term}} \tag{4.20a}$$

$$+\underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{m'=1,m'\neq m}^{M}\left(H_{mk,uu}^e(t)\right)^* H_{m'k,uu}^e(t)\cdot\Delta\theta_{m',u}^e(t)}_{\text{interference term due the } u\text{-th subcarrier of other workers}} \tag{4.20b}$$

$$+\underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{v=0,v\neq u}^{N-1}\sum_{m=1}^{M}\sum_{m'=1}^{M}\left(H_{mk,uu}^e(t)\right)^* H_{m'k,uv}^e(t)\Delta\theta_{m,v}^e(t)}_{\text{ICI term}} \tag{4.20c}$$

$$+\underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\left(H_{mk,uu}^e(t)\right)^* Z_{k,u}^e(t)\,.}_{\text{noise term}} \tag{4.20d}$$

There are four different terms in (4.20): the (desired) signal component, the interference term due to the u-th subcarrier of other workers (blind workers), the ICI term and noise. Using the law of large numbers, as the number of antennas at the receiver side $K \to \infty$, the signal term approaches $Y^{e}_{\text{sig},u}(t) = \sigma_H^2 \sum_{m=1}^{M} \Delta\theta^{e}_{m,u}(t)$, for $e \in [E]$ and $u \in [N_{sc}]$.

4.2.2 Analysis of Other Workers' Interference

To analyze the interference term in the e-th transmitted signal piece for $e \in [E]$ at the t-the global iteration due to different workers' signals after combining, we define the coefficient of j-th interfering gradient $\Delta\theta^{e}_{j,u}(t)$ in (4.20b) as

$$\kappa^{e}_{j,u}(t) = \frac{1}{K} \sum_{k=1}^{K} \sum_{\substack{m=1 \\ m \neq j}}^{M} \left(H^{e}_{mk,uu}(t)\right)^{*} \cdot H^{e}_{jk,uu}(t), \tag{4.21}$$

where $u \in [N_{sc}]$, and $j \in [M]$.

As we have shown in [10], the variance of $\kappa^{e}_{j,u}(t)$, which is $\mathbb{E}\left[|\kappa^{e}_{j,u}(t)|^2\right]$, can be approximately calculated as $\mathcal{O}\left(\frac{1}{{N_{sc}}^2 K}\right)(M-1){L_{tap}}^2\sigma_h^4$. Due to interfering workers, we have M such terms in (4.20b) each with zero mean, and a variance inversely proportional to the number of PS antennas. Thus, all of these interference terms approach zero as $K \to \infty$. Therefore, using a sufficiently large number of PS antennas wipes outs the destructive effects of (4.20b) on the FL algorithm.

4.2.3 Analysis of the ICI Term

To analyze the effects of time variations, we take the ICI contribution limited to a certain portion of adjacent subcarriers, i.e., we assume that the u-th subcarrier experiences interference from $2q$ neighboring subcarriers. As shown in [121], the ICI power concentrates in the neighborhood of a specific subcarrier. Therefore, most of the interference is from the neighboring subcarriers, and further away subcarriers cause much less leakage. In our study, the rapidity of time-variations is integrated into the auto-regressive channel model (4.10) through the variable α. To numerically relate q and α, we construct an OFDM word by embedding the information into a single carrier in the frequency domain while the other subcarriers are idle (e.g., zero). After transmitting this OFDM word through the time-varying channel defined in (4.10), we determine the corresponding q by considering the leakage from neighbor subcarriers greater than a certain level (e.g., 1%) of the peak transmitted signal level. For realistic wireless channels, having small α decreases the time variations and limits the amount of interference leaked from neighboring subcarriers, resulting in small q values. Hence, for practical wireless communication scenarios, this assumption is valid.

To proceed further, we consider M different groups of ICI terms separately by defining

$$\zeta^e_{j,u}(t) = \frac{1}{K}\sum_{k=1}^{K}\sum_{\substack{v=1\\v\neq u}}^{N_{sc}}\sum_{m=1}^{M}\left(H^e_{jk,uu}(t)\right)^* H^e_{mk,uv}(t), \tag{4.22}$$

as the corresponding coefficient of each ICI term group $\Delta\theta^e_{m,v}(t)$.

As shown in [10], for fixed gradient values, the absolute value of the mean of the j-th ICI term scales with $L_{tap}\sigma_h^2\mathcal{O}\left(\frac{q}{N_{sc}}\right)$, while its variance is upper bounded by a term scaling with $\sigma_h^4\left(L_{tap}\mathcal{O}\left(\frac{q^2}{{N_{sc}}^2K}\right) + M{L_{tap}}^2\mathcal{O}\left(\frac{q^2}{{N_{sc}}^2K}\right) + {L_{tap}}^2\mathcal{O}\left(\frac{q^2}{{N_{sc}}^2}\right)\right)$. Thus, all of the ICI terms decrease as the ratio of interfering adjacent subcarriers to the number of total subcarriers decreases resulting in a limited disturbance. Note that $\frac{q}{N_{sc}}$ is a measure of the amount of time variations, and increases with α. We also note that there are terms in the variance of the $\zeta^e_{j,u}(t)$ which scale with $1/K$, hence using a large number of receive antennas helps reduce the ICI. One may infer that the destructive effect of ICI on the learning can be eliminated for a moderate amount of time variations with a sufficiently large number of antennas. In the next section, we will prove that it is indeed the case through the convergence analysis of the FL with OTA aggregation over time-varying channels.

4.2.4 Global Model Update

At global iteration t, one can obtain an estimate for $\Delta\hat{\boldsymbol{\theta}}(t)$ using (4.20) for $e \in [E]$, and $u \in [N_{sc}]$ that is,

$$\Delta\hat{\theta}_{2(e-1)N_{sc}+u}\,(t) = \frac{\mathrm{Re}\left\{Y_{k,u}^{e}(t)\right\}}{M\sigma_H^2}, \tag{4.23a}$$

$$\Delta\hat{\theta}_{(2e-1)N_{sc}+u}\,(t) = \frac{\mathrm{Im}\left\{Y_{k,u}^{e}(t)\right\}}{M\sigma_H^2}. \tag{4.23b}$$

4.3 Convergence Analysis FL over Time Varying Channels

In this part, we perform an analysis of the proposed FL over time-varying channels to show that the ICI has a limited destructive effect on the convergence of the learning algorithm. For the ease of notation, we consider $E = 1$ with $N_{sc} = \lceil d/2 \rceil$, thus superscript e is dropped in the sequel.

After combining the signals received by the K antennas of the PS during the k-th global iteration, similar to the previous section, the u-th component of the received frequency domain signal given in (4.20) can be equivalently rewritten as

$$Y_u(t) = \sum_{p=1}^{4} W_{u,p}(t), \tag{4.24}$$

for $u \in [d]$ where

$$W_{u,1}(t) = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M} |H_{mk,uu}(t)|^2 \cdot (\Delta\theta_{m,u}(t) + j\Delta\theta_{m,u+d/2}(t)), \tag{4.25a}$$

$$W_{u,2}(t) = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M} (H_{mk,uu}(t))^* \cdot H_{m'k,uu}(t) \cdot (\Delta\theta_{m',u}(t) + j\Delta\theta_{m',u+d/2}(t)), \tag{4.25b}$$

$$W_{u,3}(t) = \frac{1}{K}\sum_{k=1}^{K}\sum_{\substack{v=1\\v\neq u}}^{N_{sc}}\sum_{m=1}^{M}\sum_{m'=1}^{M} \left(H_{mk,uu}(t)\right)^{*} H_{m'k,uv}(t) \cdot \left(\Delta\theta_{m,v}(t) + j\Delta\theta_{m,v+d/2}(t)\right), \tag{4.25c}$$

$$W_{u,4}(t) = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M} \left(H_{mk,uu}(t)\right)^{*} Z_{k,u}(t). \tag{4.25d}$$

For these four different signal components, we can define the desired signal estimate as

$$\Delta\hat{\theta}_{u}(t) = \sum_{p=1}^{4} \Delta\hat{\theta}_{u,p}(t), \tag{4.26}$$

based on (4.24), where

$$\Delta\hat{\theta}_{u,p}(t) = \begin{cases} \frac{\mathrm{Re}\{W_{u,p}(t)\}}{M\sigma_H^2}, & \text{if } 1 \leq i \leq d/2, \\ \frac{\mathrm{Im}\left\{W_{u-d/2,p}(t)\right\}}{M\sigma_H^2}, & \text{otherwise,} \end{cases} \tag{4.27}$$

for $u \in [d]$.

4.3.1 Preliminaries

Consider the optimal solution of the loss function $F\left(\boldsymbol{\theta}\right)$ which is defined in (4.1) with $\boldsymbol{\theta}^{*} \triangleq \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$. The corresponding minimum value is obtained as $F^{*} = F(\boldsymbol{\theta}^{*})$ where the individual optimal value of the loss function is also denoted by F_m^{*} for worker m. To represent the bias in data and its heterogeneity across the devices, we also define

$$\Gamma = F^{*} - \sum\nolimits_{m=1}^{M} \frac{B_m}{B} F_m^{*}, \tag{4.28}$$

where $\Gamma > 0$, and it represents the bias in data across the workers. For non-i.i.d. data distribution, Γ will have a higher value, while it approaches zero for i.i.d. data with enough samples.

During global iteration t, We consider the same learning rate for all the workers, which is $\eta(t)$. Note that the learning rate for the local iterations of a given global iteration, we consider constant learning rate. As a result, at iteration t, the model update is performed at the m-th worker for the local iteration $p \in [\epsilon]$ as

$$\boldsymbol{\theta}_m^{p+1}(t) \ = \boldsymbol{\theta}_m^{p}(t) \ - \eta(t) \ \nabla F_m \left(\boldsymbol{\theta}_m^{p}(t), \xi_m^{p}(t) \ \right), \tag{4.29}$$

where $m \in [M]$.

To perform the convergence analysis, as considered in [39], we assume that the loss functions $F_1, \ldots, F_M$ for each worker are L-smooth and μ-strongly convex, i.e., we have

$$F_m(\boldsymbol{v}) - F_m(\boldsymbol{w}) \leq \langle \boldsymbol{v} - \boldsymbol{w}, \nabla F_m(\boldsymbol{w}) \rangle + \frac{L}{2} \left\| \boldsymbol{v} - \boldsymbol{w} \right\|_2^2, \tag{4.30}$$

$$F_m(\boldsymbol{v}) - F_m(\boldsymbol{w}) \geq \langle \boldsymbol{v} - \boldsymbol{w}, \nabla F_m(\boldsymbol{w}) \rangle + \frac{\mu}{2} \left\| \boldsymbol{v} - \boldsymbol{w} \right\|_2^2, \tag{4.31}$$

for $n \in [M]$ and $\forall \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^d$, respectively. We further assume that the expected squared l_2-norm of the stochastic gradients are bounded; i.e., we have

$$\mathbb{E}_{\xi} \left[\left\| \nabla F_m \left(\boldsymbol{\theta}_m^{p}(t), \xi_m^{p}(t) \ \right) \right\|_2^2 \right] \leq G^2, \tag{4.32}$$

for $m \in [M]$, local iteration $\forall p \in [\epsilon]$, and global iteration $\forall t$.

4.3.2 Convergence Rate

A bound on the convergence rate of the federated learning with OTA aggregation over time varying channels is derived in the next theorem.

Theorem 1. *Consider* $0 < \eta(t) \leq \min \left\{ 1, \frac{1}{\mu \epsilon} \right\}$, $\forall t$. *We have*

$$\mathbb{E} \left[\left\| \boldsymbol{\theta}(t) - \boldsymbol{\theta}^* \right\|_2^2 \right] \leq \left(\prod_{i=0}^{t-1} A(i) \right) \left\| \boldsymbol{\theta}(0) - \boldsymbol{\theta}^* \right\|_2^2 + \sum_{j=0}^{t-1} B(j) \prod_{i=j+1}^{t-1} A(i), \tag{4.33}$$

where

$$A(i) \triangleq 1 - \mu\eta(i) \ \left(\epsilon - \eta(i)(\epsilon - 1)\right), \tag{4.34a}$$

$$\begin{aligned} B(i) \triangleq & \frac{\eta^2(i)\epsilon^2 G^2}{K} + \frac{\sigma_Z^2 d}{2KM\sigma_H^2} + 4\left(\frac{(M+2)q^2 d}{KM} + \mathcal{O}(1/d)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2 \\ & + \ (1 + \mu(1-\eta(i)))\,\eta^2(i)G^2\frac{\epsilon(\epsilon-1)(2\epsilon-1)}{6} \\ & + (\epsilon^2 + \epsilon - 1)\eta^2(i)G^2 + 2\eta(i)(\epsilon-1)\Gamma, \end{aligned} \tag{4.34b}$$

$\hat{G}^2$ *is the upper bound of the* $\mathbb{E}\left[\Delta\theta_{m,v_1}(t)\Delta\theta_{m,v_2}(t)\right]$*,* q *is the number of subcarriers which contribute to the ICI with the amount of time variations for given* α*, and the expectation is with respect to the stochastic gradient function and the randomness of the wireless channel.*

Proof. The proof is provided in Appendix 4.6.1. □

Similar to the discussion in [39], this upper bound consists of two parts: the first one is $\|\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*\|_2^2$ which is the distance of the initial starting point to the optimal solution and scaled with the coefficient $\left(\prod_{i=0}^{t-1} A(i)\right)$ while $\sum_{j=0}^{t-1} B(j) \prod_{i=j+1}^{t-1} A(i)$ is the second one which represents the residual distance of the current model to the optimal solution. While commenting on this upper bound, one should consider both these terms. Decreasing $A(i)$ may lead to faster convergence, but it also affects the second term due to common parameters, e.g., the local iteration count ϵ. Similar to [39], increasing the number of local iterations decreases $A(i)$ leading to faster convergence while it also increases the second term resulting in a solution further away from the optimal one. Also, note that larger Γ and G values capture more biased and less symmetric data distributions, i.e., represent non-i.i.d. data [76]. The second term in the upper bound which is $\sum_{j=0}^{t-1} B(j) \prod_{i=j+1}^{t-1} A(i)$ represents the residual distance of the current parameter at global iteration t to the optimal one. Hence, for non-i.i.d. data, one should expect to have larger $B(i)$, which increases the residual distance and negatively impacts the convergence rate of the FL over time-varying channels. As a result, it may lead to a solution far from the optimal one.

Remark 1. *We consider the leakage from $2q$ neighboring subcarriers for time-varying channels, which is a valid assumption for practical time-varying wireless channels. The impact of this leakage is captured by the third term of $B(i)$ in Theorem 1, which is $4\left(\frac{(M+2)q^2d}{KM}+\mathcal{O}(1/d)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2$. This term is due to the time-varying nature of the channel and reflects the effect of leakage from other subcarriers on the one under consideration, namely, the ICI. In other words, this term enables us to assess the effects of time variations on FL with OTA aggregation analytically. As expected, when there is no leakage from the neighboring gradient elements, i.e., when $q = 0$, this term will be exactly zero resulting in no ICI; thus, we will not observe any related term in the upper bound. On the other hand, when there is leakage with nonzero q, there will be a contribution from the third term of $B(i)$. As a result, similar to non-i.i.d. data, a solution further from the optimal one may be observed. Furthermore, as explained in Section 4.2.3, the value of q is proportional to the amount of time variations in the channel gains. Hence, when we have a rapidly-varying channel with higher q, the contribution of the ICI term to the upper bound in Theorem 1 will be higher, resulting in an inferior learning performance. As the time variations are reduced, i.e., when q is lower (i.e., when α is smaller), the contribution of this term to the upper bound will diminish. In addition to these, this impact can be alleviated using a higher number of PS antennas since it is inversely proportional to K.*

Remark 2. *As we have shown in Section 4.2.1, the parameter server can recover the average of the local updates from all the workers; however, there are three additional interference terms, i.e., due to blind workers (i.e., the interference due to the subject subcarriers of other workers), ICI, and noise, respectively; and all those interference terms contribute to the upper bound in Theorem 1. In $B(i)$, the first term $\frac{\eta^2(i)\epsilon^2G^2}{K}$ is due to blind workers and the desired term, the second term $\frac{\sigma_Z^2d}{2KM\sigma_H^2}$ is result of the channel noise while the third term is due to the time-variations in the channel (Remark 1). Hence, one can simply remove the contributions of these interference terms and assess the performance of federated learning without any imperfections; thus, we can easily compare our findings in Theorem 1 with the standard federated learning. Furthermore, note that these interference terms are inversely proportional to the number of PS antennas, K; hence, we can reduce their effects by increasing the number of receive antennas.*

Table 4.2: CNN architecture for MNIST dataset.

MNIST dataset
5 × 5 convolutional layer, 10 channels, ReLu activation, stride: (1,1)
5 × 5 convolutional layer, 20 channels, ReLu activation, stride: (1,1)
dropout with probability 0.5
fully connected layer with 320 units, ReLu activation
fully connected layer with 50 units, ReLu activation
softmax output layer with 10 units

We can further improve the performance by increasing the number of workers, which reduces the upper bound.

Corollary 1. *After performing T global iterations, using the L-smoothness of the loss functions, one can upper bound the convergence rate of FL over time-varying channels as*

$$\begin{aligned}\mathbb{E}\left[F(\boldsymbol{\theta}(T))\right] - F^* &\leq \frac{L}{2}\mathbb{E}\left[\|\boldsymbol{\theta}(T) - \boldsymbol{\theta}^*\|_2^2\right] \\ &\overset{a}{\leq} \frac{L}{2}\left(\prod_{i=0}^{T-1} A(i)\right)\|\boldsymbol{\theta}(0) - \boldsymbol{\theta}^*\|_2^2 + \frac{L}{2}\sum_{j=0}^{T-1} B(j)\prod_{i=j+1}^{T-1} A(i), \quad (4.35)\end{aligned}$$

where (a) follows from (4.33)*.*

Note that $B(i)$ in (4.33), (4.34b) and (4.35) has some terms which scale with the learning rate $\eta(i)$ while the term $\frac{\sigma_Z^2 d}{2KM\sigma_H^2} + 4\left(\frac{(M+2)q^2 d}{KM} + \mathcal{O}(1/d)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2$, which is due to time-varying channel response and channel noise, does not scale with the learning rate. Having a decreasing learning rate with $\lim_{t\to\infty}\eta(t) = 0$ will help cancel the effect of the terms scaled with $\eta(t)$ while the others will still have a nonzero contribution to the upper bound in (4.33) and (4.35) resulting in $\mathbb{E}\left[F(\boldsymbol{\theta}(T))\right] - F^* \neq 0$ and $\mathbb{E}\left[\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}^*\|_2^2\right] \neq 0$, which is similar to the conclusion reached in [39]. However, its destructive effect can be reduced by increasing the number of antennas at the PS.

4.4 Numerical Examples

We perform numerical experiments using both MNIST [117] and CIFAR-10 [122] datasets to evaluate the performance of FL with OTA aggregation over wireless channels with time variations. For MNIST, we train a convolutional neural network (CNN) given in Table 4.2 resulting in a parameter size of $d = 21840$. We consider both i.i.d. and non-i.i.d. data distribution for MNIST. In the i.i.d. case, data samples are randomly distributed to all the workers, while in the non-i.i.d. case, we divide the training samples into groups, each with 100 samples only from the same class. Then, five groups are randomly selected and assigned to each worker. The mini-batch size is $|\xi_m^p(t)| = 500$. The number of local iterations is $\epsilon = 3$ for the MNIST. We consider SGD optimizer with learning rate of $0.5 - 5t \times 10^{-4}$ during the global iteration t, and we continue to training for $T = 160$ iterations.

For the CIFAR-10 dataset, we consider the same CNN architecture given in [39] resulting in a parameter size of $d = 307498$ with i.i.d. data distribution while the mini-batch size is taken as $|\xi_m^p(t)| = 2000$. The number of local iterations is $\epsilon = 5$. We employ the Adam optimizer [118] with a learning rate of $10^{-3} - 4t \times 10^{-6}$ for the t-th global iteration.

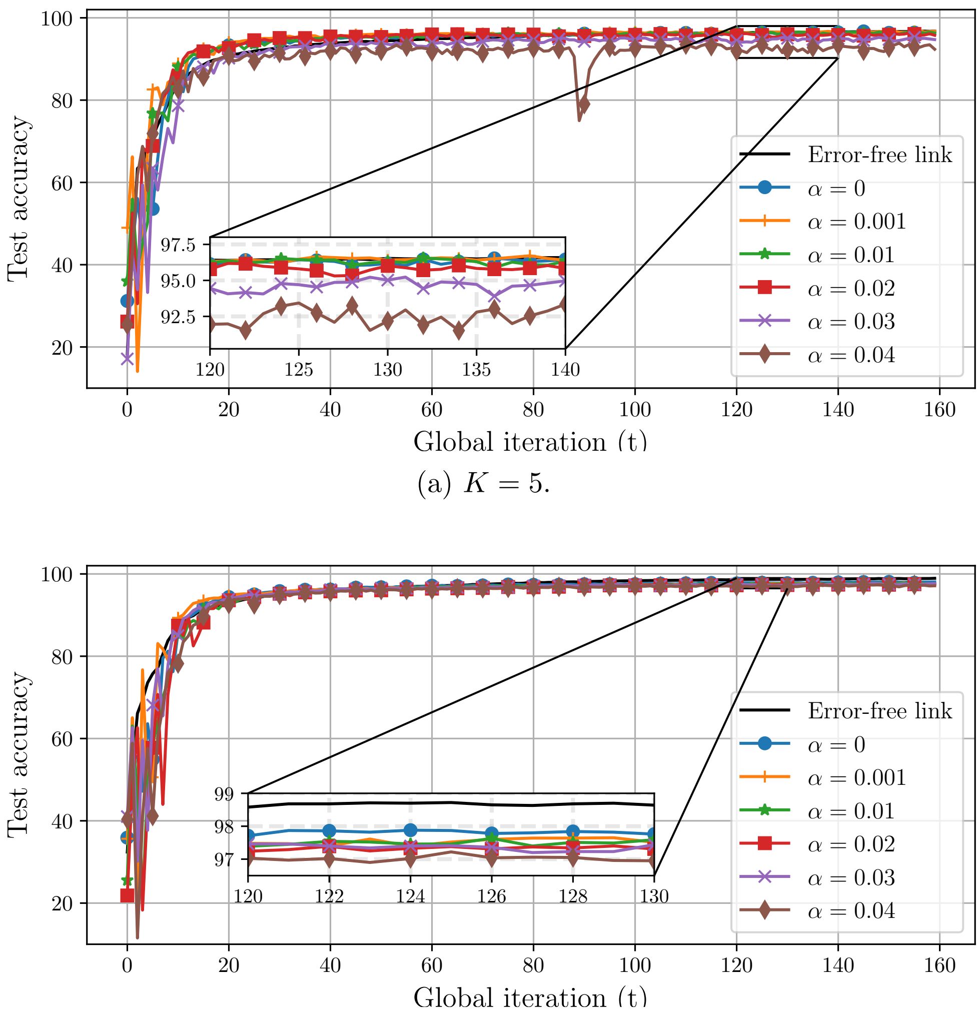
In our setup, there are M workers connected to a PS through a time-varying $L_{tap} = 3$ tap fading MAC where u and $h_{mkl,0}^e(t)$ are independent circularly symmetric zero mean complex Gaussian random variables with variance $\sigma_h^2 = 1/L_{tap}$, i.e., $u \sim \mathcal{CN}(0,\, 1/L_{tap})$, and $h_{mkl,0}^e(t) \sim \mathcal{CN}(0,\, 1/L_{tap})$, $\forall m, k, l, e, t$. Thus, we have a normalized uniform multipath delay profile where each tap experiences time-varying Rayleigh fading with the correlations among channel samples as given in (4.10). The line-of-sight transmission for each worker is taken without any delay so that $\tau_{mk1}^e(t) = 0$, $\forall m, k, e, t$; while $\tau_{mk2}^e(t)$ and $\tau_{mk2}^e(t)$ are randomly and uniformly selected between 0 and $1000T_s$ for $\forall m, k, e, t$, where $T_s = T_w/N$ is the sampling period with OFDM word duration T_w. The CP length is set to $N_{\text{cp}} = 1024$, which is enough to remove the ISI caused by the multipath.

The number of subcarriers is taken as $N_{sc} = 4096$, hence there are 4096 subchannels for each OFDM word resulting in $\lceil \frac{d}{2 \times 4096} \rceil$ OFDM words for the model update transmissions.

In our experiments, we take a range of values for α, i.e., $\alpha \in \{0.001, 0.01, 0.02, 0.03, 0.04\}$, to study different realistic communication systems for both over-the-air wireless networks with different levels of time variations. We note that in 4G or 5G wireless systems, the value of α is small, and the ICI can be negligible. However, in many of other communication systems, it may become significant. Consider a scenario with a relatively small subcarrier spacing which may be preferable for supporting massive connectivity required by massive machine type communications and reducing the impact of the delay spread [123], which results in an increased ICI effect. For instance, a relative speed of 100 km/h corresponds to $\alpha = 0.001$ in a subcarrier separation of $\Delta f = 100$ Hz for transmission around 1 GHz. Note that this is obtained by using the Jakes' model [124] to determine the relative speed for which the correlation coefficient between the consecutive samples drops to $\sqrt{1 - \alpha^2}$.

In Fig. 4.2a, the test accuracies for the i.i.d. data distribution case with the MNIST dataset are given for $K = 5$, noise variance $\sigma_z^2 = 1 \times 10^{-9}$, and $\alpha \in \{0.001, 0.01, 0.02, 0.03, 0.04\}$. The results are compared with two cases: 1) there is no Doppler spread/ICI ($\alpha = 0$), and 2) FL over the error-free shared link, which can be interpreted as a performance upper bound for the given CNN structure and learning parameters. Even though the time variations are extremely high, the ICI causes only a slight decrease in the performance of the learning algorithm, that is, averaged over the iterations, ICI causes $0.52\%, 3.09\%, 4.16\%$ decrease in accuracy with respect to the time-invariant case, respectively, for $\alpha \in \{0.02, 0.03, 0.04\}$; while there is almost no loss for $\alpha \in \{0.001, 0.01\}$. In Fig. 4.3a, we present the upper bound for the convergence rate provided in Corollary 1 for the same system given in Fig. 4.2a[1].

[1] To calculate this upper bound analytically, one needs the value of q corresponding to the given α. Noting that q represents the interference due to neighboring subcarriers, we consider transmission of an OFDM word with only one active subchannel, and use the corresponding channel output to determine the value of q to account for leakage greater than 1% of the peak transmitted signal level.



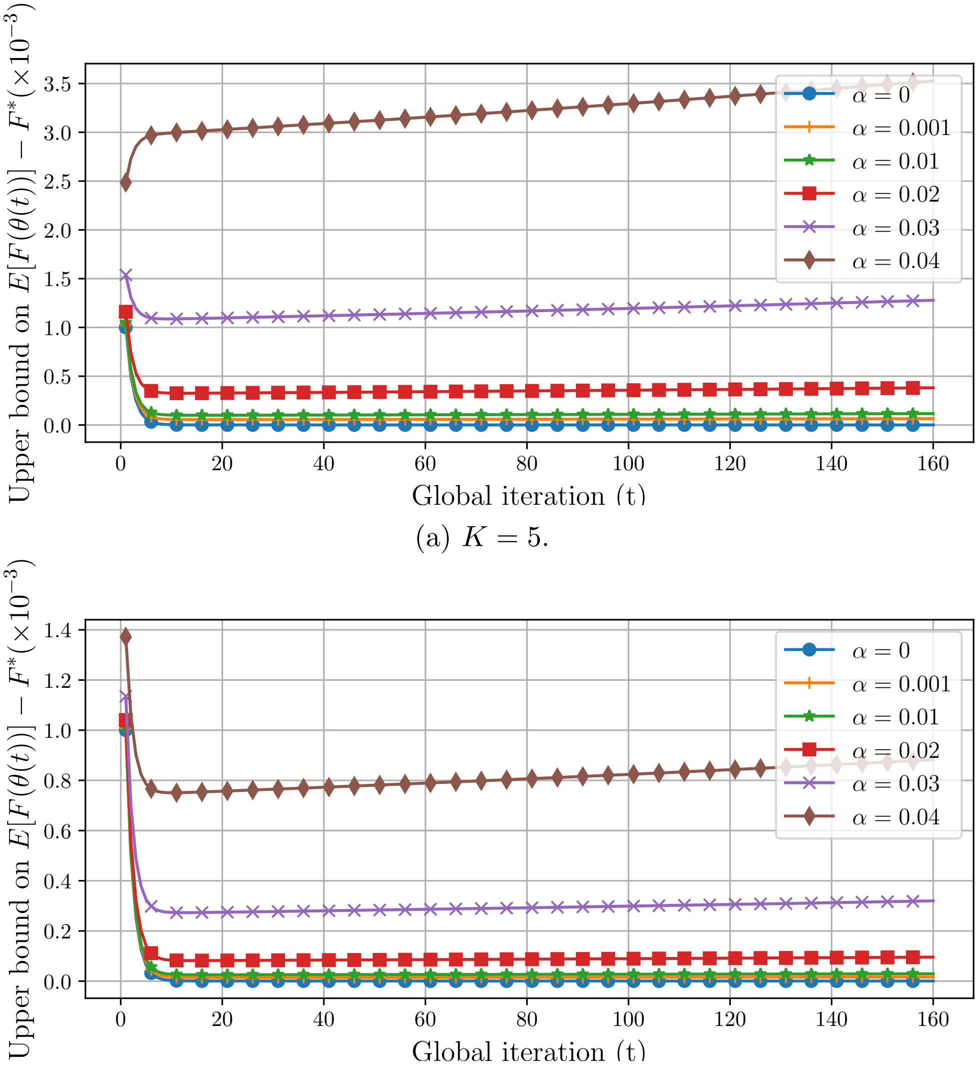
(a) $K = 5$.

(b) $K = 20$.

Figure 4.2: Test accuracy of the system with MNIST i.i.d. data distribution, $M = 20$, $\alpha \in \{0, 0.001, 0.01, 0.02, 0.03, 0.04\}$, and channel noise variance $\sigma_z^2 = 1 \times 10^{-9}$.

Even though the given results are upper bounds, the results are still consistent with the analytical expectations especially for lower time variations with smaller α values. It can be seen that the convergence rate is very close to that of the ideal case of $\alpha = 0$ and to the best case with error-free shared link.

In Fig. 4.2b, we give the test accuracies for the same setup, except that, we increase the number of PS antennas to $K = 20$.

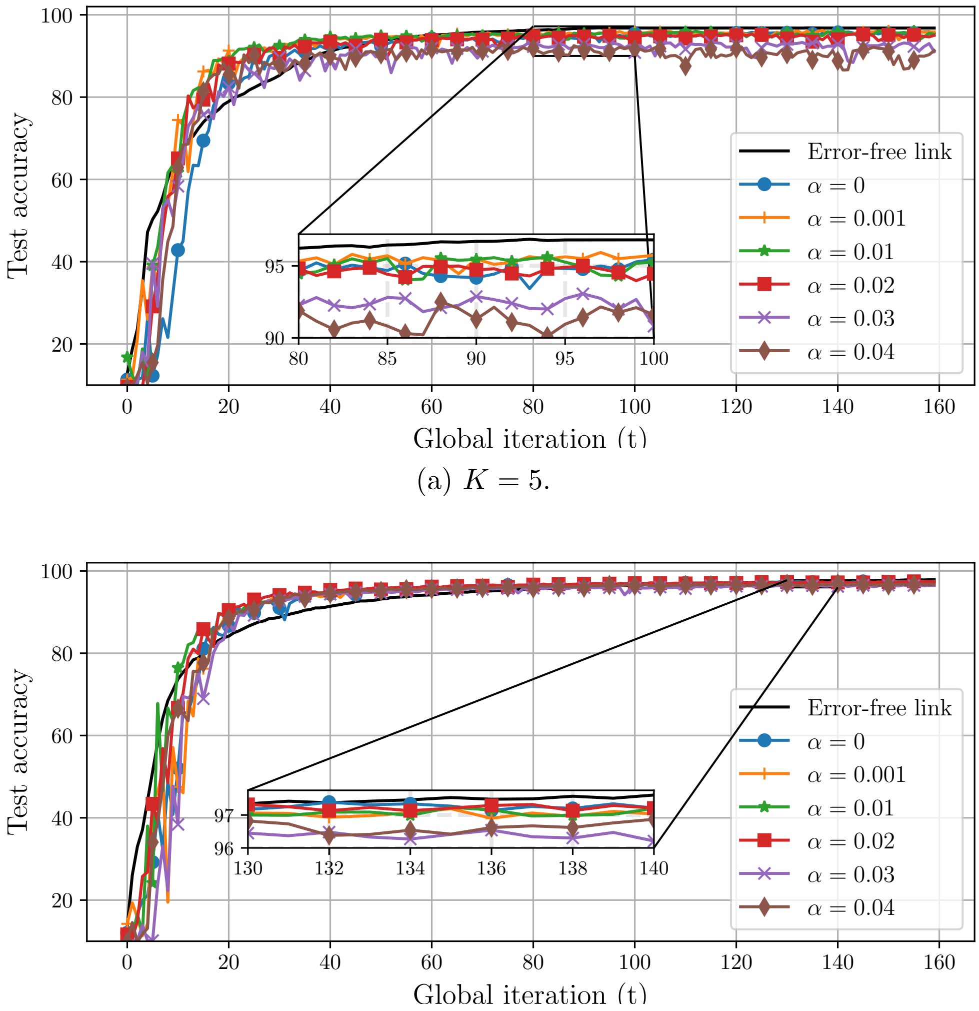


(a) $K = 5$.

(b) $K = 20$.

Figure 4.3: Upper bound on $\mathbb{E}\left[F(\boldsymbol{\theta}(T))\right] - F^*$ with i.i.d. MNIST. The parameter size is $d = 21840$, and it is assumed that a single OFDM word is generated. We consider $\sigma_z^2 = 10^{-9}$, $L_{tap} = 3$, $\epsilon = 3$, $\mu = 1$, $L = 1$, $G^2 = \Gamma = 1$, $\hat{G}^2 = 10^{-3}$, $\|\theta(0) - \theta^*\|_2^2 = 10^3$ for different level of time variations. Note that $\mathcal{O}(1/d)$ is taken as $1/d$.

By increasing the number of PS antennas, the performance of the federated learning system is improved for all given time-variation levels, while almost no performance loss is observed for $\alpha \in \{0.001, 0.01\}$. In Fig. 4.3b, the upper bound for the convergence rate is provided for the same system given in Fig. 4.2b which are again consistent with the simulations.



(a) $K = 5$.

(b) $K = 20$.

Figure 4.4: Test accuracy of the system with MNIST non-i.i.d. data distribution, $M = 20$, $\alpha \in \{0, 0.001, 0.01, 0.02, 0.03, 0.04\}$, and channel noise variance $\sigma_z^2 = 1 \times 10^{-9}$.

In Figs. 4.4a and 4.4b, we consider a similar time-varying federated learning setup except that the data distribution is non-i.i.d.. Similar to the observations in Fig. 4.2, when we have higher time variations, i.e., higher α values, the test accuracy of the FL system deteriorates; however, increasing the number of PS antennas improves the performance of all the cases. Additionally, the imperfections introduced by the blind workers are also mitigated by using multiple PS antennas.

Note that both in Figs. 4.2 and 4.4, the curves corresponding to $\alpha = 0.001$ and $\alpha = 0.01$ are very close to the that of $\alpha = 0$ and the error-free shared link with no significant degradation which shows that moderate amount of variations for FL systems can be tolerated without sacrificing to much of the performance. Furthermore, one can validate this observation for the case of i.i.d. data in Fig. 4.3 where the upper bound on $\mathbb{E}\left[F(\boldsymbol{\theta}(T))\right] - F^*$ for $\alpha = 0$, $\alpha = 0.001$ and $\alpha = 0.01$ are very low with $K = 20$, and will be even smaller by increasing the number of PS antennas.

In Fig. 4.5, we consider FL over time-varying channels for the CIFAR-10 classification task with CNN. The number of PS antennas is $K = 2M$ with $M = 20$ workers, and we use the same set of α values as we have used previously in MNIST classification simulations. Even though the estimate of the model update is obtained from a signal corrupted by both ICI and blind transmitters, with a sufficient number of PS antennas, we can obtain a reasonably high learning accuracy while maintaining a setup that represents realistic scenarios with practical implementation constraints.

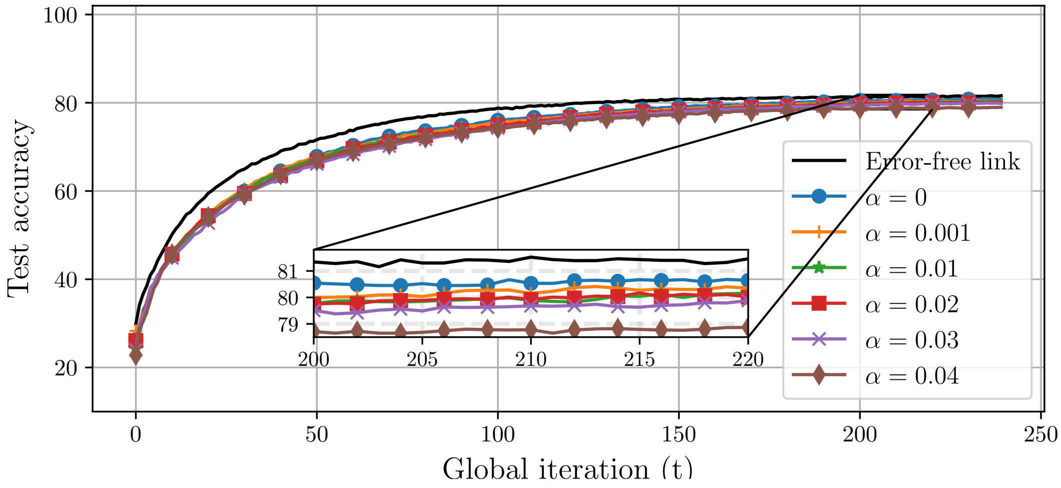


Figure 4.5: Test accuracy of the system with CIFAR-10 i.i.d. data distribution, $K = 2M$, $\alpha \in \{0, 0.001, 0.01, 0.02, 0.03, 0.04\}$, and channel noise variance $\sigma_z^2 = 1 \times 10^{-9}$.

The above results validate our expectation that we can design an efficient and accurate federated learning system even if there are time variations in the channel due to the mobility of the workers and/or PS, or changes in the medium. The results also show that even though the setup with blind transmitters results in additional interference term due to the signal alignment at the PS, we show via both analytical and simulation results that we can alleviate its destructive effect by employing a sufficient number of PS antennas.

4.5 Chapter Summary

We study an OFDM-based blind federated learning system with OTA aggregation over time-varying channels, which destroys the orthogonality among the subcarriers and results in ICI. Hence, the receiver observes a corrupted version of the local gradients, i.e., the gradient vectors' elements experience interference from the other (neighboring) gradient values.

We analyze the effects of the channel time variations on the convergence of the time-varying over-the-air FL systems and derive an upper bound for the convergence rate. We demonstrate that the ICI term has a limited destructive effect on the learning performance, especially if there is a moderate or small amount of time variation as typically experienced in wireless communications. Also, using a sufficiently large number of receive antennas helps reduce the ICI, resulting in an excellent performance of the federated learning algorithm over practical time-varying channels. This behavior is confirmed analytically through a convergence rate analysis. We further validate our results via extensive numerical experiments with both i.i.d. and non-i.i.d. data distributions, which show that the overall learning performance is only slightly deteriorated compared to the scenarios over time-invariant channels.

4.6 Appendices

4.6.1 Appendix A: Proof of Theorem 1

As in [39], an auxiliary variable $\boldsymbol{v}(t)$ is defined as

$$\boldsymbol{v}(t+1) \triangleq \boldsymbol{\theta}(t) \ + \Delta\boldsymbol{\theta}(t), \tag{4.36}$$

where $\Delta\boldsymbol{\theta}(t)$ is defined in (4.4). At iteration $t+1$, we note that $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) \ + \Delta\hat{\boldsymbol{\theta}}(t)$. As a result, we have

$$\begin{aligned}\|\boldsymbol{\theta}(t+1) - \boldsymbol{\theta}^*\|_2^2 &= \|\boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1) + \boldsymbol{v}(t+1) - \boldsymbol{\theta}^*\|_2^2 \\ &= \|\boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1)\|_2^2 + \|\boldsymbol{v}(t+1) - \boldsymbol{\theta}^*\|_2^2 \\ &\quad + 2\langle \boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1), \boldsymbol{v}(t+1) - \boldsymbol{\theta}^* \rangle.\end{aligned}$$

Three terms on the right hand side can be bounded by using the following lemmas.

Lemma 2. *For the first term, we have*

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1)\|_2^2\right] \leq{}& \frac{\eta^2(t)\tau^2 G^2}{K} + \frac{\sigma_Z^2 d}{2KM\sigma_H^2} \\ &+ 4\left(\frac{(M+2)q^2 d}{KM} + \mathcal{O}(1/d)\frac{L_{tap} q^2 \sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2.\end{aligned} \tag{4.37}$$

Proof. See Appendix 4.6.2. □

Lemma 3. *For the second and third terms, we have*

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{v}(t+1) - \boldsymbol{\theta}^*\|_2^2\right] \leq{}& (1 - \mu\eta(t) \ \ (\tau - \eta(t)(\tau-1)))\,\mathbb{E}\left[\|\boldsymbol{\theta}(t) \ - \boldsymbol{\theta}^*\|_2^2\right] \\ &+ (1 + \mu(1 - \eta(t)))\,\eta^2(t) G^2 \frac{\tau(\tau-1)(2\tau-1)}{6} \\ &+ \eta^2(t)(\tau^2 + \tau - 1)G^2 + 2\eta(t)(\tau-1)\Gamma,\end{aligned} \tag{4.38}$$

and

$$\mathbb{E}\big[\langle \boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1), \boldsymbol{v}(t+1) - \boldsymbol{\theta}^* \rangle\big] = 0. \tag{4.39}$$

Proof. Note that the second term only contains the global parameter $\boldsymbol{\theta}(t)$, average of the local updates $\Delta\boldsymbol{\theta}(t)$ and the optimal model $\boldsymbol{\theta}^*$. Thus, there is no term depending on the channel model. Hence, the proof is same as Lemma 2 in [39]. For the third term, the proof is same as Lemma 3 in [39]. □

4.6.2 Appendix B: Proof of Lemma 2

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\theta}(t+1)-\boldsymbol{v}(t+1)\|_2^2\right] &= \mathbb{E}\left[\left\|\Delta\hat{\boldsymbol{\theta}}(t)-\Delta\boldsymbol{\theta}(t)\ \right\|_2^2\right] \\ &= \sum_{u=1}^{d}\mathbb{E}\left[\left(\Delta\hat{\theta}_u(t)\ -\Delta\theta_u(t)\ \right)^2\right], \qquad (4.40)\end{aligned}$$

where $\Delta\theta_u$ is the u-th entry of vector $\Delta\boldsymbol{\theta}(t)$, for $u\in[d]$. In the following, we bound $\mathbb{E}\left[\left(\Delta\hat{\theta}_u(t)\ -\Delta\theta_u(t)\ \right)^2\right]$, $\forall u$. Here we remind that $\Delta\hat{\theta}_u\,(t)=\sum_{p=1}^{4}\Delta\hat{\theta}_{u,p}\,(t)$, where $\Delta\hat{\theta}_{u,p}\,(t)$ is defined in (4.27). Note that, we can write

$$\mathbb{E}\left[\left(\Delta\hat{\theta}_u(t)\ -\Delta\theta_u(t)\ \right)^2\right]=\mathbb{E}\left[\left(\Delta\hat{\theta}_{u,1}(t)\ -\Delta\theta_u(t)\ \right)^2\right]+\sum_{p=2}^{4}\mathbb{E}\left[\left(\Delta\hat{\theta}_{u,p}(t)\right)^2\right], \qquad (4.41)$$

using the independence of channel realization $H_{mk,uv}$ and channel noise $z_{k,u}$, $\forall m,\ k,u$. For the rest of the proof, as stated in Section 4.3, it is assumed that a single OFDM word is generated from the parameters, i.e., $N_{sc}=\lceil\frac{d}{2}\rceil$. For simplicity, we further assume $N_{sc}=d/2\in\mathbb{Z}^+$.

Lemma 4. *For the first signal term in* (4.25)*, we have*

$$\sum_{u=1}^{d}\mathbb{E}\left[\left(\Delta\hat{\theta}_{u,1}(t)-\Delta\theta_u(t)\right)^2\right]=\frac{1}{KM^2}\sum_{m=1}^{M}\mathbb{E}\left[\ \|\Delta\boldsymbol{\theta}_m(t)\|_2^2\ \right]. \qquad (4.42)$$

Proof. Using the definition of $\Delta\hat{\theta}_{u,1}(t)$ given in (4.27), we have

$$\mathbb{E}\left[\left(\Delta\hat{\theta}_{u,1}(t)-\Delta\theta_u(t)\right)^2\right]=\mathbb{E}\left[\left(\frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{K\sigma_H^2}\sum_{k=1}^{K}|H_{mk,uu}(t)|^2-1\right)\Delta\theta_{m,u}(t)\right)^2\right]$$

$$= \mathbb{E}\Bigg[\frac{1}{M^2}\sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\Bigg(1 - \frac{1}{K\sigma_H^2}\sum_{k=1}^{K}|H_{m_1k,uu}(t)\ |^2 - \frac{1}{K\sigma_H^2}\sum_{k=1}^{K}|H_{m_2k,uu}(t)\ |^2$$
$$+ \frac{1}{K^2\sigma_H^4}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}|H_{m_1k_1,uu}(t)\ |^2\,|H_{m_2k_2,uu}(t)\ |^2\Bigg)\cdot\Delta\theta_{m_1,u}(t)\Delta\theta_{m_2,u}(t)\Bigg]. \tag{4.43}$$

To compute (4.43), we need to analyze two cases:

- $m_1 \neq m_2$: $\Delta\theta_{m_1,u}(t)$ and $\Delta\theta_{m_2,u}(t)$ have zero mean and they are independent. Thus, there will be no contribution from this case.

- $m_1 = m_2$: We have

$$\mathbb{E}\Bigg[\left(\Delta\hat{\theta}_{u,1}(t) - \Delta\theta_u(t)\ \right)^2\Bigg]\Bigg|_{m_1=m_2} \overset{(a)}{=} \frac{1}{M^2}\sum_{m=1}^{M}\Bigg(1 - \frac{2}{K\sigma_H^2}\sum_{k=1}^{K}\mathbb{E}\Big[\ |H_{mk,uu}(t)\ |^2\Big]$$
$$+ \frac{1}{K^2\sigma_H^4}\sum_{k_1=1}^{K}\sum_{k=2}^{K}\mathbb{E}\Big[\ |H_{mk_1,uu}(t)\ |^2\cdot|H_{mk_2,uu}(t)\ |^2\Big]\Bigg)\mathbb{E}\Big[\Delta\theta_{m,u}^2(t)\Big]$$
$$\overset{(b)}{=} \mathbb{E}\left[\frac{1}{KM^2}\sum_{m=1}^{M}\Delta\theta_{m,u}^2(t)\ \right], \tag{4.44}$$

where (a) is due to independence of $\Delta\theta_{m,u}$ and $H_{mk,uv}$, $\forall m,\ k,\ u,\ v$, and (b) is obtained using $\mathbb{E}\Big[\ |H_{mk,uu}(t)|^2\ \Big] = \sigma_H^2$ and $\mathbb{E}\Big[\ |H_{mk,uu}(t)|^4\ \Big] = 2\sigma_H^4$.

□

Lemma 5. *For the second term in* (4.25)*, which is the interference due to the u-th subcarrier of other workers, we have*

$$\sum_{u=1}^{d}\mathbb{E}\left[\Delta\hat{\theta}_{u,2}^2(t)\ \right] = \frac{(M-1)}{KM^2}\sum_{m=1}^{M}\mathbb{E}\big[\ \|\Delta\boldsymbol{\theta}_m(t)\ \|_2^2\ \big]. \tag{4.45}$$

Proof. Analysis of this term is similar to the one given in [39]. For $1 \leq u \leq d/2$, we have

$$\begin{aligned}
\mathbb{E}\left[\Delta\hat{\theta}^2_{u,2}(t)\ \right] &= \mathbb{E}\Bigg[\Big(\frac{1}{KM\sigma_H^2}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\sum_{k=1}^{K}\mathrm{Re}\Big\{(H_{mk,uu}(t)\ )^* H_{m'k,uu}(t) \\
&\qquad\qquad \cdot\big(\Delta\theta_{m',u}(t) + j\Delta\theta_{m',d/2+u}(t)\ \big)\Big\}\Big)^2\Bigg] \\
&\overset{(a)}{=} \mathbb{E}\Big[\frac{1}{K^2M^2\sigma_H^2}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\sum_{k=1}^{K}\Big(\mathrm{Re}\Big\{(H_{mk,uu}(t))^* H_{m'k,uu}(t) \\
&\qquad\qquad \cdot\big(\Delta\theta_{m',u}(t) + j\Delta\theta_{m',d/2+u}(t)\ \big)\Big\}\Big)^2 \\
&\qquad\qquad + \mathrm{Re}\Big\{(H_{mk,uu}(t))^* H_{m'k,uu}(t) \\
&\qquad\qquad \cdot\big(\Delta\theta_{m',u}(t) + j\Delta\theta_{m',d/2+u}(t)\ \big)\Big\} \\
&\qquad\qquad \cdot\mathrm{Re}\Big\{(H_{m'k,uu}(t)\ )^* H_{mk,uu}(t) \\
&\qquad\qquad \cdot\big(\Delta\theta_{m,u}(t) + j\Delta\theta_{m,d/2+u}(t)\ \big)\Big\} \\
&\overset{(b)}{=} \mathbb{E}\Big[\frac{1}{2KM^2}\sum_{m=1}^{M}\Big((M-1)\big(\Delta\theta^2_{m,u}(t) + \Delta\theta^2_{m,d/2+u}(t)\ \big) \\
&\qquad\qquad + \sum_{m'=1,m'\neq m}^{M}\big(\Delta\theta_{m,u}(t)\ \Delta\theta_{m',i}(t) \\
&\qquad\qquad - \Delta\theta_{m,d/2+u}(t)\ \Delta\theta_{m',d/2+u}(t)\big)\Big)\Big],
\end{aligned} \tag{4.46}$$

(a) is due to independence of $H_{mk,uv}$, $\forall m,\ k$, and (b) is obtained using $\mathbb{E}\Big[\ |H_{mk,uu}(t)|^2\ \Big] = \sigma_H^2$. As in [39], one can perform the similar analysis for $d/2+1 \leq i \leq d$, and obtain

$$\begin{aligned}
\mathbb{E}\left[\Delta\hat{\theta}^2_{u,2}(t)\ \right] &= \mathbb{E}\Bigg[\frac{1}{2KM^2}\sum_{m=1}^{M}\Big((M-1)\big(\Delta\theta^2_{m,u-d/2}(t)\ + \Delta\theta^2_{m,u}(t)\ \big) \\
&\qquad + \sum_{\substack{m'=1\\ m'\neq m}}^{M}\big(\Delta\theta_{m,u}(t)\ \Delta\theta_{m',u}(t)\ - \Delta\theta_{m,u-d/2}(t)\ \Delta\theta_{m',u-d/2}(t)\big)\Big)\Bigg].
\end{aligned} \tag{4.47}$$

Combining (4.46) and (4.47), we have

$$\sum_{u=1}^{d} \mathbb{E}\left[\Delta\hat{\theta}^2_{u,2}(t)\ \right] = \frac{(M-1)}{KM^2} \sum_{m=1}^{M} \mathbb{E}\big[\ \|\Delta\boldsymbol{\theta}_m(t)\ \|_2^2\big]. \tag{4.48}$$

□

Lemma 6. *For the third term in* (4.25)*, which is the ICI term, we have an upper bound*

$$\sum_{u=1}^{d} \mathbb{E}\left[\Delta\hat{\theta}^2_{u,3}(t)\ \right] \le 4\left(\frac{(M+2)q^2 d}{KM} + \mathcal{O}(1/d)\frac{L_{tap} q^2 \sigma_h^4}{KM\sigma_H^4}\right)\hat{G}. \tag{4.49}$$

Proof. For $1 \le u \le d/2$, we have

$$\begin{aligned}\mathbb{E}\left[\Delta\hat{\theta}^2_{u,3}(t)\ \right] &= \mathbb{E}\Bigg[\Big(\frac{1}{KM\sigma_H^2}\sum_{k=1}^{K}\sum_{\substack{v=u-q\\ v\neq u}}^{u+q}\sum_{m=1}^{M}\sum_{m'=1}^{M}\mathrm{Re}\Big\{(H_{mk,uu}(t)\ )^* \\ &\qquad \cdot H_{m'k,uv}(t)\big(\Delta\theta_{m,v}(t)\ \ + j\Delta\theta_{m,d/2+v}(t)\ \big)\Big\}\Big)^2\Bigg] \\ &= \frac{1}{4K^2M^2\sigma_H^4}\sum_{\substack{k_1=1\\ k_2=1}}^{K}\sum_{\substack{v_1=u-q\\ v_2=u-q\\ v_1\neq u\\ v_2\neq u}}^{u+q}\sum_{\substack{m_1=1\\ m_2=1\\ m_3=1\\ m_4=1}}^{M}\mathbb{E}\left[(A_1+A_2+A_3+A_4)\right],\end{aligned} \tag{4.50}$$

where

$$\begin{aligned}A1 \triangleq\ & (H_{m_1k_1,uu}(t))^*\,(H_{m_3k_2,uu}(t))^*\,H_{m_2k_1,uv_1}(t)H_{m_4k_2,uv_2}(t) \\ & \cdot\Big(\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2}(t) + j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2+d/2}(t) \\ & + j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2}(t) \\ & - \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2+d/2}(t)\Big),\end{aligned} \tag{4.51a}$$

$$\begin{aligned}A2 \triangleq\ & (H_{m_1k_1,uu}(t))^*\,H_{m_3k_2,uu}(t)H_{m_2k_1,uv_1}(t)\,(H_{m_4k_2,uv_2}(t))^* \\ & \cdot\Big(\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2}(t) - j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2+d/2}(t) \\ & + j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2}(t)\end{aligned}$$

$$\begin{aligned}&\qquad\qquad + \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2+d/2}(t)\Big), &(4.51\text{b})\\
A3 \triangleq\ & H_{m_1k_1,uu}(t)\left(H_{m_3k_2,uu}(t)\right)^*\left(H_{m_2k_1,uv_1}(t)\right)^* H_{m_4k_2,uv_2}(t)\\
&\cdot\Big(\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2}(t) + j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2+d/2}(t)\\
&\quad - j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2}(t)\\
&\qquad\qquad + \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2+d/2}(t)\Big), &(4.51\text{c})\\
A4 \triangleq\ & H_{m_1k_1,uu}(t)H_{m_3k_2,uu}(t)\left(H_{m_2k_1,uv_1}(t)\right)^*\left(H_{m_4k_2,uv_2}(t)\right)^*\\
&\cdot\Big(\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2}(t) - j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2+d/2}(t)\\
&\quad - j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2}(t)\\
&\qquad\qquad - \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2+d/2}(t)\Big). &(4.51\text{d})\end{aligned}$$

To compute (4.50), we need to analyze these for terms:

- $\mathbb{E}[A_1]$: Since the channel coefficients are independent for different m and k values; the first multiplier

  $$\mathbb{E}\Big[\left(H_{m_1k_1,uu}(t)\right)^*\left(H_{m_3k_2,uu}(t)\right)^* H_{m_2k_1,uv_1}(t)H_{m_4k_2,uv_2}(t)\Big],$$

  has nonzero value only for the cases $\{m_1 = m_2,\ m_3 = m_4,\ m_1 \neq m_3,\ k_1 = k_2\}$, $\{m_1 = m_4,\ m_2 = m_3,\ m_1 \neq m_3,\ k_1 = k_2\}$ and $\{m_1 = m_2 = m_3 = m_4,\ k_1 = k_2\}$. For these conditions, the second multiplier in A_1 is

  $$\begin{aligned}\mathbb{E}\big[&\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2}(t) + j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_3,v_2+d/2}(t)\\
  &+ j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2}(t) - \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_3,v_2+d/2}(t)\big].\end{aligned} \tag{4.52}$$

  For $\{m_1 = m_2,\ m_3 = m_4,\ m_1 \neq m_3,\ k_1 = k_2\}$ and $\{m_1 = m_4,\ m_2 = m_3,\ m_1 \neq m_3,\ k_1 = k_2\}$, the second term will be zero since $\Delta\theta_{m,v}$ terms are zero mean and independent for different m values. We also model the gradients and corresponding local model updates as WSS processes, that is $\mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2}(t) - \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2+d/2}(t)\big] = 0$. Furthermore, we assume that the gradients and corresponding local updates decorrelates sufficiently fast, e.g., there is no correlation between the gradient samples which are $d/2-q$ index away from each other. As a result, for $\{m_1=m_2=m_3=m_4,\ k_1=k_2\}$, $\mathbb{E}\left[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2+d/2}(t)\right]=0$, and $\mathbb{E}\left[\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2}(t)\right]=0$[2]. Note that this can be achieved since we consider interference due to $2q$ neighboring subcarriers, i.e., $\{v_1\in[u-q,u+q],\ v_1\neq u\}$ and $\{v_2\in[u-q,u+q],\ v_2\neq u\}$ which is a valid assumption for practical wireless communication scenarios. Thus, there will be no contribution from this case.

- $\mathbb{E}[A_2]$: Similar to the previous term, the first multiplier

$$\mathbb{E}[A_{2,1}]=\mathbb{E}\big[\left(H_{m_1k_1,uu}(t)\right)^*H_{m_3k_2,uu}(t)H_{m_2k_1,uv_1}(t)\left(H_{m_4k_2,uv_2}(t)\right)^*\big],\quad(4.53)$$

has nonzero value only for the cases $\mathcal{C}_1=\{m_1=m_3,\ m_2=m_4,\ m_1\neq m_2,\ k_1=k_2\}$ and $\mathcal{C}_2=\{m_1=m_2=m_3=m_4,\ k_1=k_2\}$. Note that for $\{m_1=m_2,\ m_3=m_4,\ m_1\neq m_3,\ k_1=k_2\}$ and $\{m_1=m_4,\ m_2=m_3,\ m_1\neq m_3,\ k_1=k_2\}$, the second multiplier in A_2 is zero due to independence of $\Delta\theta_{m,v}$'s for different m's. Hence, there is no need to investigate these cases.

  - For $\mathcal{C}_1$: Since $m_1\neq m_2$, due to independence, we have

$$\mathbb{E}[A_{2,1}]\Big|_{\mathcal{C}_1}=\mathbb{E}\left[|H_{m_1k_1,uu}(t)|^2\right]\mathbb{E}\left[H_{m_2k_1,uv_1}(t)\left(H_{m_2k_1,uv_2}(t)\right)^*\right].$$

As shown in (4.19), $\mathbb{E}\left[|H_{m_1k_1,uu}(t)|^2\right]=\sigma_H^2$. For the second term, we have

$$\begin{aligned}&\mathbb{E}\left[H_{m_2k_1,uv_1}(t)\left(H_{m_2k_1,uv_2}(t)\right)^*\right]\\&\quad=\frac{1}{{N_{sc}}^2}\sum_{l_1,l_2\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\mathbb{E}\Big[h_{m_2kl_1,i_1}(t)\cdot\left(h_{m_2kl_2,i_2}(t)\right)^*\Big]e^{-jx_1},\end{aligned}\quad(4.54)$$

[2] Note that this assumption is not essential in the upper bound calculations. One may also bound $\mathbb{E}\left[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2+d/2}(t)\right]$ term with $\sqrt{\mathrm{Var}(\Delta\theta_{m_1,v_1}(t))\mathrm{Var}(\Delta\theta_{m_1,v_2+d/2}(t))}$.

where

$$x_1 = \frac{2\pi(i_1(u-v_1)+v_1\tau_{m_2kl_1}(t)}{N_{sc}} - \frac{i_2(u-v_2)+v_2\tau_{m_2kl_2}(t))}{N_{sc}}. \quad (4.55)$$

Using independency of $h_{m_2kl_1,i_1}(t)$ and $h_{m_2kl_2,i_2}(t)$ when $l_1 \neq l_2$, we have

$$\begin{aligned}&\mathbb{E}\left[H_{m_2k_1,uv_1}(t)\left(H_{m_2k_1,uv_2}(t)\right)^*\right]\\&= \frac{1}{{N_{sc}}^2}\sum_{l_1\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\mathbb{E}\Big[h_{m_2kl_1,i_1}(t)\cdot(h_{m_2kl_1,i_2}(t))^*\Big]e^{-jx_1}, \quad (4.56)\end{aligned}$$

whose absolute value can be upper bounded by

$$\begin{aligned}&\Big|\mathbb{E}\big[H_{m_2k_1,uv_1}(t)\left(H_{m_2k_1,uv_2}(t)\right)^*\big]\Big|\\&\leq \frac{1}{{N_{sc}}^2}\sum_{l_1\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\big|\mathbb{E}\big[h_{m_2kl_1,i_1}(t)\cdot(h_{m_2kl_1,i_2}(t))^*\big]\big|\cdot\big|e^{-jx_1}\big| \quad (4.57a)\\&\overset{(a)}{=} \frac{1}{{N_{sc}}^2}\sum_{l_1\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\big|\mathbb{E}\big[h_{m_2kl_1,i_1}(t)\cdot(h_{m_2kl_1,i_2}(t))^*\big]\big| \quad (4.57b)\\&\overset{(b)}{=} \frac{1}{{N_{sc}}^2}\sum_{l_1\in[L_{tap}]}\sum_{i_1,i_2\in[N_{sc}]}\mathbb{E}\big[h_{m_2kl_1,i_1}(t)\cdot(h_{m_2kl_1,i_2}(t))^*\big] \quad (4.57c)\\&\overset{(c)}{=} \sigma_H^2, \quad (4.57d)\end{aligned}$$

where (a) follows from $|e^{-jx_1}| = 1$, (b) is achieved ny

$$\mathbb{E}\left[h_{m_2kl_1,i_1}(t)\left(h_{m_2kl_1,i_2}(t)\right)^*\right] = (1-\alpha^2)^{|i_1-i_2|/2}\sigma_h^2 \in \mathbb{R}_{\geq 0},$$

and (c) is due to the definition of σ_H^2 given in (4.19). Combining these two term, we have

$$|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_1} \leq \sigma_H^4. \quad (4.58)$$

- For $\mathcal{C}_2$, we have $m_1 = m_2 = m_3 = m_4 = m$ and $k_1 = k_2 = k$ which gives

$$\mathbb{E}[A_{2,1}]\Big|_{\mathcal{C}_2} = \mathbb{E}\big[\,(H_{mk,uu}(t))^*\,H_{mk,uu}(t)\cdot H_{mk,uv_1}(t)\,(H_{mk,uv_2}(t))^*\,\big] \tag{4.59}$$

$$= \frac{1}{{N_{sc}}^4}\sum_{\substack{l_c\in[L_{tap}]\\ c\in[4]}}\sum_{\substack{i_g\in[N_{sc}]\\ g\in[4]}}\mathbb{E}\big[\,(h_{mkl_1,i_1}(t))^*\,h_{mkl_2,i_2}(t)$$

$$\cdot\,h_{mkl_3,i_3}(t)\,(h_{mkl_4,i_4}(t))^*\,\big]e^{-jx_2}, \tag{4.60}$$

where

$$\begin{aligned} x_2 =& \frac{-2\pi u\tau_{mkl_1}(t)+2\pi u\tau_{mkl_2}(t)+2\pi i_3(u-v_1)}{N_{sc}} \\ &+\frac{v_1\tau_{mkl_3}(t)-2\pi i_4(u-v_2)-v_2\tau_{mkl_4}(t)}{N_{sc}}. \end{aligned}$$

Similar to $\mathcal{C}_1$, we can upper bound the absolute value of (4.59) as

$$\begin{aligned} |\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_2} &= \big|\mathbb{E}\big[\,(H_{mk,uu}(t))^*\,H_{mk,uu}(t)\cdot H_{mk,uv_1}(t)\,(H_{mk,uv_2}(t))^*\,\big]\big| \\ &\le \frac{1}{{N_{sc}}^4}\sum_{\substack{l_c\in[L_{tap}]\\ c\in[4]}}\sum_{\substack{i_g\in[N_{sc}]\\ g\in[4]}}\big|\mathbb{E}\big[\,(h_{mkl_1,i_1}(t))^* \\ &\qquad\cdot\,h_{mkl_2,i_2}(t)h_{mkl_3,i_3}(t)\,(h_{mkl_4,i_4}(t))^*\,\big]\big|. \end{aligned} \tag{4.61}$$

For (4.61), we have four nonzero cases:

$$\begin{aligned} &\mathcal{C}_{2,1} = \{l_1 = l_2 = l_3 = l_4\}, \\ &\mathcal{C}_{2,2} = \{l_1 = l_2,\ l_3 = l_4,\ l_1 \neq l_3\}, \\ &\mathcal{C}_{2,3} = \{l_1 = l_3,\ l_2 = l_4,\ l_1 \neq l_2\}, \\ &\mathcal{C}_{2,4} = \{l_1 = l_4,\ l_2 = l_3,\ l_1 \neq l_3\}. \end{aligned}$$

For $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,1}}$, firstly consider $\mathbb{E}\left[(h_{mkl,i_1}(t))^*\,h_{mkl,i_2}(t)h_{mkl,i_3}(t)\,(h_{mkl,i_4}(t))^*\right]$. Since $h_{mkl,i_1}(t)$, $h_{mkl,i_2}(t)$, $h_{mkl,i_3}(t)$, and $h_{mkl,i_4}(t)$ are jointly Gaussian complex random variables; using [125], we have

$$\begin{aligned}&\mathbb{E}\left[\left(h_{mkl,i_1}(t)\right)^* h_{mkl,i_2}(t) h_{mkl,i_3}(t) \left(h_{mkl,i_4}(t)\right)^*\right] \\ &\qquad = \left(r^{(|i_1-i_2|+|i_3-i_4|)/2} + r^{(|i_1-i_3|+|i_2-i_4|)/2}\right)\sigma_h^4,\end{aligned}$$

where $r = \sqrt{1-\alpha^2}$. Clearly, (4.62) has a nonnegative value; thus, its absolute value is equal to itself. As we have shown in [10],

$$\sum_{\substack{i_g \in [N_{sc}] \\ g \in [4]}} \mathbb{E}\Big[\left(h_{mkl_1,i_1}(t)\right)^* h_{mkl_2,i_2}(t) h_{mkl_3,i_3}(t) \left(h_{mkl_4,i_4}(t)\right)^*\Big],$$

is $\sigma_h^4 \mathcal{O}({N_{sc}}^2)$, resulting in $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,1}} \leq L_{tap}\sigma_h^4\mathcal{O}(1/{N_{sc}}^2)$ where $\sigma_H^2 = \sigma_h^4\mathcal{O}(1/{N_{sc}}^2)$.

For $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,2}}$, $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,3}}$ and $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,4}}$ are obtained with the similar idea to (4.57), resulting in $|\mathbb{E}[A_{2,1}]|\Big|_{\mathcal{C}_{2,i}} \leq \sigma_H^4$ for $i \in \{2,3,4\}$.

For the second multiplier in A_2, we have

$$\begin{aligned}\mathbb{E}[A_{2,2}] = \mathbb{E}\big[&\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2}(t) \\ &- j\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2+d/2}(t) \\ &+ j\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2}(t) \\ &+ \Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2+d/2}(t)\big],\end{aligned} \tag{4.62}$$

for the nonzero cases. Terms $\mathbb{E}\left[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2+d/2}(t)\right]$ and $\mathbb{E}\left[\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2}(t)\right]$ can be safely approximated as zero by using the assumption of fast decorrelation of gradients as stated in A_1. For the remaining terms, we have

$$\begin{aligned}\mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2}(t)\big] =& \mathrm{Cov}\left(\Delta\theta_{m_1,v_1}(t), \Delta\theta_{m_1,v_2}(t)\right) \\ &+ \mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\big]\mathbb{E}\big[\Delta\theta_{m_1,v_2}(t)\big],\end{aligned} \tag{4.63}$$

which is equal to $\mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2}(t)\big] = \mathrm{Cov}\left(\Delta\theta_{m_1,v_1}(t), \Delta\theta_{m_1,v_2}(t)\right)$ since $\mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\big] = 0$ and $\mathbb{E}\big[\Delta\theta_{m_1,v_2}(t)\big] = 0$.

Note that $\mathrm{Cov}\left(\Delta\theta_{m_1,v_1}(t), \Delta\theta_{m_1,v_2}(t)\right) \leq \max\left(\mathrm{Var}\left(\Delta\theta_{m_1,v_1}(t)\right), \mathrm{Var}\left(\Delta\theta_{m_1,v_2}(t)\right)\right)$.

Thus, $\mathbb{E}\big[\Delta\theta_{m_1,v_1}(t)\Delta\theta_{m_1,v_2}(t)\big]$ can be upper bounded by a constant $\hat{G}^2$ which can be implied by Assumption 3. Similarly, one can also bound $\mathbf{E}\big[\Delta\theta_{m_1,v_1+d/2}(t)\Delta\theta_{m_1,v_2+d/2}(t)\big]$ by $\hat{G}^2$.

- $\mathbb{E}[A_3]$ is similar to $\mathbb{E}[A_2]$, and $\mathbb{E}[A_4]$ is similar to $\mathbb{E}[A_1]$.

Inserting nonzero values, i.e., $\mathbb{E}[A_{2,1}]$, $\mathbb{E}[A_{2,2}]$, $\mathbb{E}[A_{4,1}]$, and $\mathbb{E}[A_{4,2}]$ into (4.50), we have

$$\mathbb{E}\left[\Delta\hat{\theta}_{u,3}^2(t)\ \right] \leq 4\Big(\frac{q^2(M-1)}{KM} + \frac{3q^2}{KM} + \mathcal{O}(1/{N_{sc}}^2)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\Big)\hat{G}^2 \tag{4.64a}$$

$$= 4\left(\frac{(M+2)q^2}{KM} + \mathcal{O}(1/{N_{sc}}^2)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2. \tag{4.64b}$$

The same procedure follows for $d/2+1 \leq u \leq d$, and the proof is concluded by inserting (4.64) to $\sum\limits_{u=1}^{d}\mathbb{E}\left[\Delta\hat{\theta}_{u,3}^2(t)\ \right]$. □

Lemma 7. *For the last term in* (4.25)*, which is due to channel noise, we have*

$$\sum_{u=1}^{d}\mathbb{E}\left[\Delta\hat{\theta}_{u,4}^2(t)\ \right] = \frac{\sigma_Z^2 d}{2KM\sigma_H^2}. \tag{4.65}$$

Proof. The proof is similar to that of Lemma 6 in [39]. □

By plugging the results of Lemmas 4-7 into (4.40), and using the convexity of $\|\cdot\|_2^2$ and the assumption of bounded expected l_2-norm of the stochastic gradients, similar to Lemma 1 of [39], we obtain

$$\mathbb{E}\left[\|\boldsymbol{\theta}(t+1) - \boldsymbol{v}(t+1)\|_2^2\right] \leq \frac{\eta^2(t)\tau^2G^2}{K} + \frac{\sigma_Z^2 d}{2KM\sigma_H^2}$$
$$+ 4\left(\frac{(M+2)q^2d}{KM} + \mathcal{O}(1/d)\frac{L_{tap}q^2\sigma_h^4}{KM\sigma_H^4}\right)\hat{G}^2. \tag{4.66}$$

Chapter 5

Transformation-Invariant Over-the-Air Combining for Multi-Sensor Wireless Inference

Deep neural networks are powerful but computationally complex machine learning models, which pose challenges for running DNN-based applications on simple edge units, such as IoT devices, due to their limited computational capabilities and energy constraints. To address this, pruning techniques can be used to remove insignificant weights and enhance performance and speed up learning and inference processes [85, 86, 87]. Another approach is to split the network between the edge device and a more powerful server, improving efficiency.

In this chapter, we examine sensor networks where multiple sensors gather data from overlapping regions and perform inference on some shared phenomenon. To reduce the computational burden of deep learning techniques, we split the network into two parts: a front-end on the sensor side and a back-end on a device with more computational power. Clearly, proper design of such a system requires considering sensor capabilities, wireless link requirements, and system latency to ensure accuracy and reliability while minimizing data transmission. In [101], a similar computing hierarchy consisting of cloud, edge and end devices where the

DNN layers are mapped into hierarchical parts and the whole network is optimized to maximize the usefulness of the features, however, the effect of wireless channels and efficiency of transmission is ignored. In [102], deep over-the-air computation is introduced for transmission efficient distributed inference over wireless channels using multiple sensors. The study employs averaging operation as the feature fusion method; however, previous works reveal that it may be possible to use different sensor fusion operations to increase representativeness of the feature vectors [101, 126, 127, 128]. With this motivation, we propose computing over-the-air maximum approximations by using an L_p-norm inspired function and LogSumExp for feature fusion for both unimodal and multi-modal networks to improve the usefulness of the overall gathered data. With this approach, we leverage the multiple sensors to obtain transformation-invariant features in a bandwidth-efficient manner.

The chapter is organized as follows. Section 5.1 introduces the system model. Over-the-air transformation-invariant combining for multi-sensor wireless inference is introduced in Section 5.2. Performance of several tasks over AWGN channels are studied via simulations in Section 5.3, and the chapter is concluded in Section 5.4.

5.1 System Model

Consider a multi-sensor wireless inference system where M sensors with limited computational capabilities collect data for the same object from different angles or sources, aiming to infer common information from all the collected data. This could involve any type of data, such as text, audio, or visual data, and the sensors may use different modalities or sensing techniques. Specifically, we propose a transformation invariant over-the-air multi-sensor network model as illustrated in Fig. 5.1 where the i-th sensor computes an intermediate feature vector $\mathbf{F}_i$ based on its own sensed data and front-end network branch. The intermediate feature vector is then preprocessed by the function $\phi(\cdot)$ and transmitted over a wireless channel to a central device. The central device operates on the fused

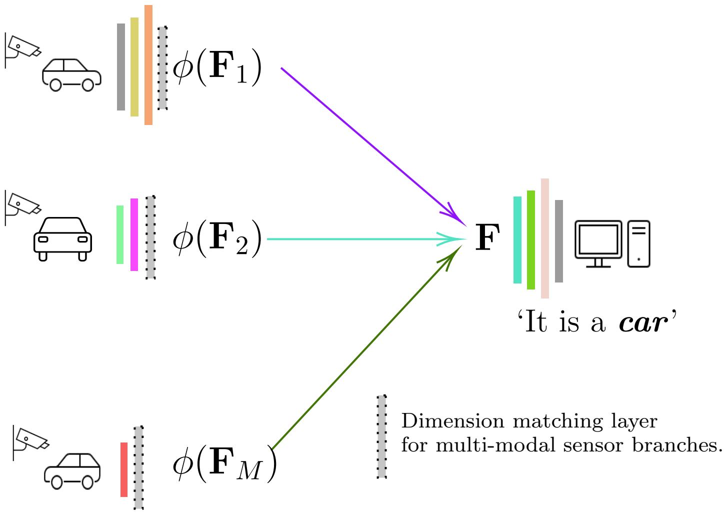


Figure 5.1: System model for the proposed multi-sensor wireless inference approach with over-the-air combining.

feature vector $\mathbf{F}$ and completes the inference task by employing the back-end of the network. This approach allows an implicit collaboration between the sensors and central device for collaborative inference in multi-sensor networks.

With the proposed approach, the activations of each sensor are transmitted over an AWGN MAC and combined over-the-air, leading to an efficient solution that decreases transmission overhead and bandwidth requirements while using a single helper device. The goal of the over-the-air combining operation is to obtain a single, reliable, transformation-invariant and representative feature vector from multiple sensors, instead of a separate vector for each sensor. This approach only requires the transmission of feature vectors on a shared link and the central device aims to recover the combination of these features. Therefore, our system requires less bandwidth compared to traditional methods, which require separate transmission and recovery of each signal. We further consider multi-modal layer structure for the front branches of the neural networks employed at the sensor side; hence, sensors with higher computational capabilities can use more complicated layers while the less capable ones will utilize only shadow networks. Thus, our solution is a transmission-efficient wireless inference method with multi-modal sensor networks while increasing the performance of both powerful and weak sensors by exploiting sensor fusion.

5.2 Transformation-Invariant Over-the-Air Combining for Multi-Sensor Wireless Inference

In wireless networks, over-the-air aggregation is a technique used to improve spectral efficiency by allowing multiple transmitters to share the same communication resources. In this technique, the transmitters send their data to the receiver simultaneously, and the receiver can directly obtain the summation or average of the transmitted symbols without explicitly decoding the individual symbols using the superposition property of the MAC. This property has been exploited in various wireless communication scenarios, including over-the-air computation scenarios (see, e.g., [39, 129, 11, 130]). Acquiring transformation-invariant features is indispensable for ensuring reliable outcomes across diverse data types, including image, video, sound or radar data, when faced with alterations such as rotations, crops, time shifts, amplitude changes, phase shifts or other variations that are inherent to each domain. A simple yet effective solution is data augmentation, as presented in [131]. A more sophisticated approach is a transformation-invariant (TI) pooling operator [126], implemented in [127]. For instance, for vision problems, this method feeds different rotated versions of the same sample into the first part of the network, and combines them using a TI-pooling layer to perform the remaining operations required for the network. This TI-pooling layer implements an element-wise maximum operator, resulting in a transformation-invariant feature vector. In [128], the authors design a multi-view convolutional neural network (MVCNN) for 3D shape recognition using 2D section images of the 3D model, using the maximum operator to pool and combine layers in a similar manner.

Standard maximum operation requires the receiver to recover each transmitted signal separately before taking the element-wise maximum. However, this process demands a significant amount of bandwidth and can also be computationally expensive, making it infeasible for real-time applications. In this chapter, we are interested in transformation-invariant over-the-air sensor fusion for multi-sensor networks; however, to improve the representativeness of the fused feature vector, we aim to move beyond simple averaging techniques. To enable an approximate maximum operation for the multi-sensor wireless inference setup in a bandwidth efficient manner, we propose using an L_p-norm inspired function and LogSumExp. These approximations allow us to approximate the maximum operation using a combination of elementary functions that are computationally efficient to implement. By leveraging the superposition property of the MAC, we can exploit the benefits of maximum combining without sacrificing spectral efficiency while improving the overall inference performance and robustness of the system.

It is worth noting that while the maximum operation is utilized in [127] to train a robust network for rotation or scale changes, and [128] focuses on computationally efficient 3D shape recognition using 2D views, our study takes a unique approach by considering a distributed multi-sensor network rather than relying on a single machine for computations and data sensing. Specifically, we explore the use of locally obtained data from each sensor, which is then combined in a bandwidth-efficient way with the approximate maximum operation for further processing and joint inference, distinguishing our work from the aforementioned studies.

Remark 3. *We consider two different structures for the multi-sensor wireless inference system under consideration:*

(I) Unimodal wireless inference using sensors with the same computational power: *One can employ the same front-end branches with the same number of layers, neurons, and filters with weight sharing among the sensors. For this setup, it is enough to train one front-end and one back-end branch jointly due to weight sharing. We refer to this structure as the unimodal wireless inference setup.*

(II) Multi-modal wireless inference using sensors with different computational capabilities: *In this setup, we allocate different front-end branches at various sensors based on their computational power and capabilities; hence the sensors with higher capabilities may deploy deeper network branches to improve the overall performance while the less capable ones may only employ shallow network branches to contribute to the overall inference. We note that because of* *multi-modality, the resulting intermediate feature vectors may have different dimensions. Hence, one should utilize a dimension-matching layer at the end of each front-end branch, as shown in Fig. 5.1.*

Remark 4. *Note that, unlike the distributed and federated learning, this study focuses on the inference phase. Our approach involves offline training, which can be performed in a powerful device beforehand, followed by the sharing of network branch weights between the sensors and the central device for real-time inference. It is worth noting that although the pre-trained network weights are shared with the sensors before the inference phase, there is no explicit communication among the sensors during the inference. This makes the proposed approach highly scalable and suitable for large-scale networks with many sensors. Overall, our approach provides an effective solution for achieving real-time, bandwidth-efficient inference using multiple sensors in a network.*

5.2.1 LogSumExp Approximation for Over-the-Air Maximum

Since we are interested in obtaining the maximum of the transmitted features in a multi-sensor network with OTA sensor fusion, the m-th worker deploys the front-end of the network resulting in feature vector $\mathbf{F}_m$. Instead of directly transmitting $\mathbf{F}_m$, each sensor will deploy the preprocessing function $\phi(\cdot)$, then the m-th sensor will obtain and transmit

$$\mathbf{x}_m = \phi(\mathbf{F}_m) = e^{\xi \mathbf{F}_m}, \tag{5.1}$$

for the LSE approximation where $\xi > 0$ is used as a scaling parameter. Note that $\mathbf{F}_m$ can be 1D vectors, 2D matrices or 3D tensors depending on the network structure and layers. For simplicity, we assume $\mathbf{F}_m \in \mathbb{R}^d$ is a one-dimensional vector that can be obtained by flattening 2D matrices or 3D tensors where d depends on the input sample, and network parameters. We further define

$$e^{\xi \mathbf{F}_m} \triangleq \left[e^{\xi \mathbf{F}_m(1)}, e^{\xi \mathbf{F}_m(2)}, \cdots, e^{\xi \mathbf{F}_m(d)}\right], \tag{5.2}$$

where $\mathbf{F}_m(i)$ is the i-th element of the vector $\mathbf{F}_m$.

The received signal at the central device is

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{x}_m + \mathbf{n} = \sum_{m=1}^{M} e^{\xi \mathbf{F}_m} + \mathbf{n}, \tag{5.3}$$

where $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{n} \in \mathbb{R}^d$ is the AWGN noise vector with variance σ_n^2. Note that one can also consider transmission over other channel models, e.g., fading channels. The i-th element of the received signal (5.3) can be written as

$$y(i) = \sum_{m=1}^{M} e^{\xi F_m(i)} + n(i), \tag{5.4}$$

for $i \in \{1, \cdots, d\}$. To obtain an approximation for the element-wise maximum with LSE, the receiver takes the natural logarithm of the received signal (5.8), and obtains

$$\begin{aligned} \max\{F_1(i), \cdots, F_M(i)\} &\approx \frac{\log\left(y(i)\right)}{\xi} \\ &= \frac{\log\left(\sum_{m=1}^{M} e^{\xi F_m(i)} + n(i)\right)}{\xi}. \end{aligned} \tag{5.5}$$

After obtaining an approximation for the element-wise maximum for each feature vector element, the resulting vector is fed to the back end of the network at the central device, and a single fused inference result is obtained.

Note that one needs to be careful in the selection of the parameter ξ. Higher ξ will provide a better approximation, while it may also lead to unstable calculations due to extremely high values. Additionally, increasing ξ leads to a higher transmit power, which may not be desirable. Therefore, it is crucial to carefully weigh the trade-off between the inference performance and the transmit power, while also considering the stability of the calculations when selecting the scaling parameter.

5.2.2 L_p-Norm Inspired Approximation for Over-the-Air Maximum

In this case, similar to the LSE approximation, instead of the resulting feature vector, which is the output of the front-end of the network, sensor m transmits

$$\mathbf{x}_m = \phi(\mathbf{F}_m) = \mathbf{F}_m^p, \tag{5.6}$$

for the L_p-norm inspired approximation where $p > 0$[1]. It is worth mentioning that $\mathbf{F}_m^p$ can be expressed as a vector consisting of d elements, denoted by $[\mathbf{F}_m(1)^p, \mathbf{F}_m(2)^p, \cdots, \mathbf{F}_m(d)^p]$, where $\mathbf{F}_m(i)$ is a scalar representing the i-th element of vector $\mathbf{F}_m$. Note that L_∞ provides the largest magnitude among each vector element; hence having larger p enables a better approximation of the maximum. We further emphasize that $\mathbf{F}_m$'s are the outputs of commonly used activation functions (e.g., rectified linear unit (ReLU) and sigmoid). Hence, the approximation for the largest magnitude will be an approximation for the maximum.

The received signal at the device is

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{x}_m + \mathbf{n} = \sum_{m=1}^{M} \mathbf{F}_m^p + \mathbf{n}, \tag{5.7}$$

where $\mathbf{y} \in \mathbb{R}^d$, and $\mathbf{n} \in \mathbb{R}^d$ is the AWGN noise vector with variance σ_n^2.

The i-th element of the received signal (6.2) is

$$y(i) = \sum_{m=1}^{M} F_m(i)^p + n(i), \tag{5.8}$$

for $i \in \{1, \cdots, d\}$. Before processing the received signal at the back-end part of

[1] The function given in (5.6) is inspired by the L_p norm but is adapted for usage when $p > 0$, diverging from the conventional L_p norm definition for $0 < p < 1$.

the network, one needs to take the $\frac{1}{p}$-th power of the received signal which is

$$\begin{aligned}\max\{F_1(i), \cdots, F_M(i)\} &\approx y(i)^{\frac{1}{p}} \\ &= \left(\sum_{m=1}^{M} F_m(i)^p + n(i)\right)^{\frac{1}{p}}.\end{aligned} \tag{5.9}$$

This operation provides an approximation for the element-wise maximum and the resulting vector is fed to the back end of the network which will complete the inference operation based on the fused feature vector.

5.3 Numerical Examples

We perform numerical experiments using both MNIST [117] and Princeton ModelNet 40-class subset dataset [132] to evaluate the performance of transformation-invariant over-the-air wireless inference with multi-sensor networks. ModelNet originally provides 3D CAD models for objects, and [128] constructs multiview 2D samples from 3D CAD models using a 12-view camera setup. In our experiments, we use the 2D multiview dataset provided by [128]. We consider offline training for learning, and assume that the weights of the network branches are shared with the sensors and central device, which will perform real-time inference. For performance evaluation, we consider three baselines: 1) the exact maximum for intermediate features, which can be considered as a performance upper bound for transformation-invariant networks with both ideal (no channel noise) scenario and transmission over AWGN channels, 2) averaging of intermediate features as a sensor fusion operation which can be directly implemented in an over-the-air manner, 3) single sensor performance for both unimodal and multi-modal structures. To compare the transmission efficiency of the proposed approach with the raw image transmission, we assume that the resolution of the input image pixels is the same as the resolution of the intermediate feature vector elements.

For the MNIST dataset, we consider a five-sensor unimodal wireless inference network where each sensor senses the same digit, but each sample is randomly

Table 5.1: Network architecture for the unimodal wireless inference with the MNIST dataset.

Front-end (Sensor side)	Image: 28×28
	5×5 convolutional layer, 10 channels, ReLu activation, stride: (1,1)
Back-end (Device side)	5×5 convolutional layer, 20 channels, ReLu activation, stride: (1,1)
	dropout with probability 0.5
	fully connected layer with 320 units, ReLu activation
	fully connected layer with 50 units, ReLu activation
	softmax output layer with 10 units

rotated about the origin. The angle of rotation is uniformly distributed over $[0,\ \pi]$. We consider i.i.d. data distribution with batch size 2000, and Adam optimizer [118] is employed with an initial learning rate of $\eta = 10^{-3}$ over $T = 200$ iterations. The network architecture for both the sensor side front-end and device side back-end is provided in Table 5.1. First, each sensor employs the front-end branch of the network, and transmits $e^{\xi \mathbf{F}_m}$ where $\mathbf{F}_m$ is the intermediate feature vector for the m-th sensor. We consider decreasing scaling parameter ξ for LSE approximation as

$$\xi = \begin{cases} 5, & \text{if epoch} \leq 2 \\ 3, & \text{if epoch} \leq 4 \\ 2, & \text{if epoch} \leq 30 \\ 1, & \text{otherwise.} \end{cases}$$

It should be noted that we utilize a functional scaling parameter, which achieves satisfactory results without optimization, even though further optimization is possible. We model the wireless channel between the sensors and device as an AWGN MAC as in (5.3) where the channel SNR is 10 dB. Employing the LSE approximation method (detailed in Section 5.2.1), we estimate the element-wise maximums of the transmitted intermediate features. We compare the results of the proposed approach with three baselines in Fig. 5.2.

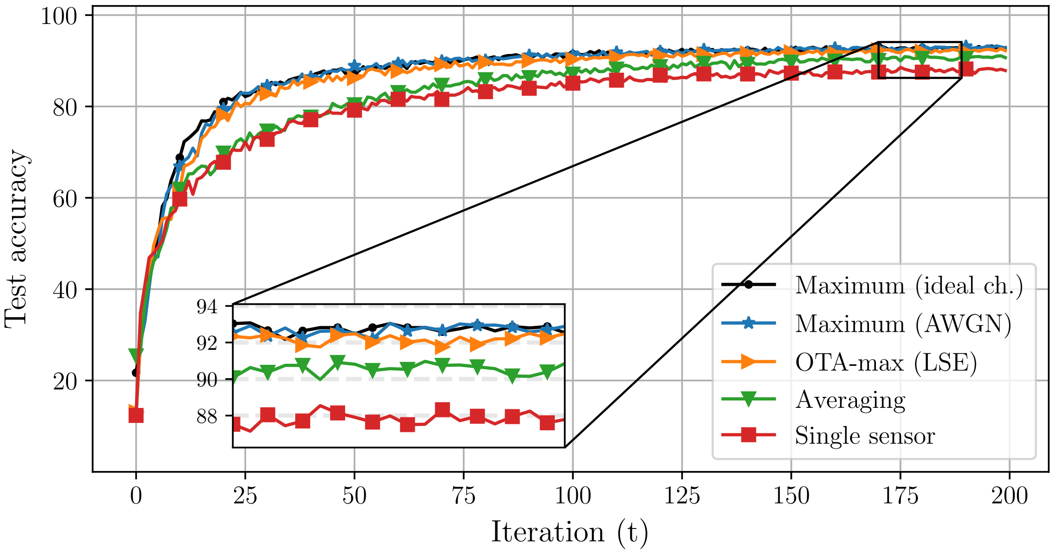


Figure 5.2: Inference accuracy for MNIST dataset with $M = 5$ sensors each observing the same object with random rotations for unimodal wireless inference.

The exact maximum achieves the highest accuracy, while a single sensor without leveraging multi-view data underperforms other fusion methods. The proposed over-the-air LSE approximation approaches the performance of the exact maximum, outperforming averaging in sensor fusion. This result shows that sensor fusion in transformation invariant multi-sensor wireless inference networks can be implemented in an efficient manner by employing over-the-air computations.

We further note that, in the above setup, the input image size is $28 \times 28 = 784$ pixels, and the transmitted vector size of $\mathbf{F}_m$ is $d = 10 \times 112 \times 12 = 1440$ due to data amplification. Note that this size is affected by the splitting point, and can be reduced by later division or employing an auto-encoder. However, since there are five sensors, the overall number of parameters to be transmitted with over-the-air sensor fusion is $10 \times 112 \times 12 = 1440$ compared to that of separate raw image transmission, which is 784 for each sensor with overall $5 \times 784 = 3920$ transmitted pixel values. Hence, the proposed approach provides significant reduction on the required bandwidth.

We also evaluate the performance of the proposed approach on the ModelNet

Table 5.2: Network architecture for the unimodal wireless inference with the ModelNet dataset.

Front-end (Sensor side)	Image: $224 \times 224 \times 3$
	3×3 convolutional layer, 32 channels, ReLU, stride:4, padding: 2
	2D maxpooling with kernelsize=3, stride=2
	dropout with probability 0.5
Back-end (Device-size)	fully connected layer (23328, 8196) ReLU
	dropout with probability 0.5
	fully connected layer (8196, 2048) ReLU
	Output layer: fully connected layer (2048, 40)

dataset using both unimodal and multi-modal wireless inference setups. We use a mini-batch size of 10 and Adam optimizer [118] with an initial learning rate of $\eta = 10^{-4}$ over $T = 80$ iterations. In both cases, we use the AWGN MAC channel with 10 dB SNR and the over-the-air approach described in Section 5.2.2 with $p = 2$ for approximating the maximum operation. For the unimodal case, we use a 12-camera setup as described in [128], and the network structure for the sensor and device-edge is provided in Table 5.2.

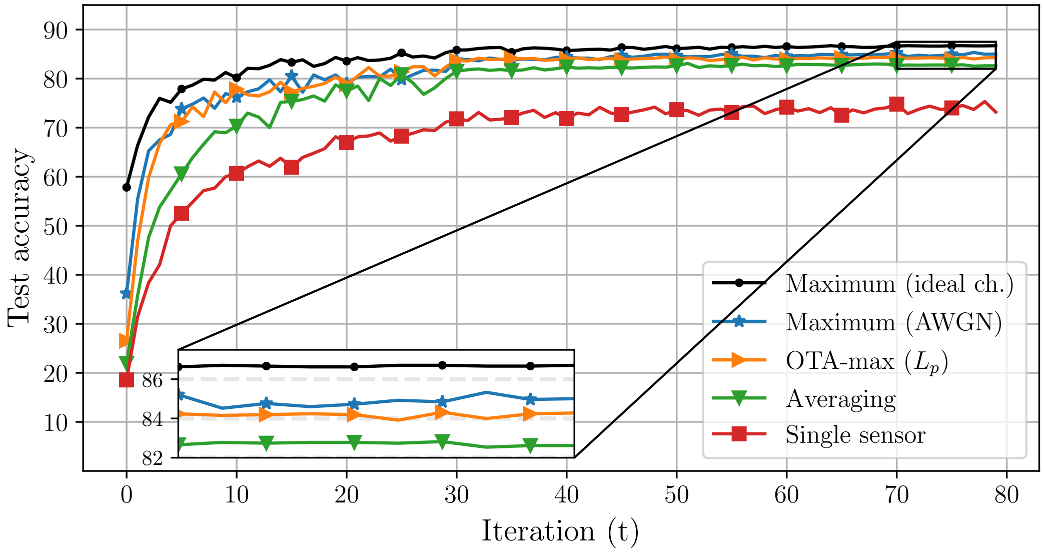


Figure 5.3: Inference accuracy for MoelNet dataset with $M = 12$ sensors each observing the same object from a different angle for unimodal wireless inference.

We compare the inference accuracies of the proposed approach with those of the baselines, and report the results in Fig. 5.3. Similar to the MNIST simulations, the exact maximum achieves the highest inference accuracy, while the approach proposed in Section 5.2.2 for the maximum achieves nearly the same level of accuracy and outperforms the other two baselines (sensor fusion with averaging and single sensor solutions).

For the multi-modal wireless inference setup, we investigate the network structure given in Table 5.3 with four sensors in which each sensor uses data randomly sampled from 12 views of the ModelNet dataset.

Table 5.3: Network architecture for the multi-modal system with the ModelNet dataset.

Image: $224 \times 224 \times 3$		
Front-end (Sensor side)	**Sensor 1**	**Sensor 2**
	7×7 conv layer, 32 channels, ReLU, stride: 4, padding: 2	5×5 conv layer, 32 channels, ReLU, stride: 4, padding: 2
	Max pooling with kernel size: 3, stride: 2	Max pooling with kernel size: 3, stride: 2
	5×5 conv layer, 32 channels, ReLU, stride: 1, padding: 2	3×3 conv layer, 16 channels, ReLU, stride: 1, padding: 2
	Max pooling with kernel size: 3, stride: 2	Max pooling with kernel size: 3, stride: 2
	Dropout layer ($p = 0.5$)	Dropout layer ($p = 0.5$)
	Fully connected layer (5408, 2048)	Fully connected layer (3136, 2048)
	Sensor 3	**Sensor 4**
	5×5 conv layer, 16 channels, ReLU, stride: 4, padding: 2	3×3 conv layer, 16 channels, ReLU, stride: 4, padding: 2
	Max pooling with kernel size: 3, stride: 2	Max pooling with kernel size: 3, stride: 2
	Dropout layer ($p = 0.5$)	Dropout layer ($p = 0.5$)
	Fully connected layer (11664, 2048)	Fully connected layer (12544, 2048)
Back-end (Device side)	Fully connected layer (2048, 1024), ReLU activation	
	Fully connected layer (1024, 40), ReLU activation	

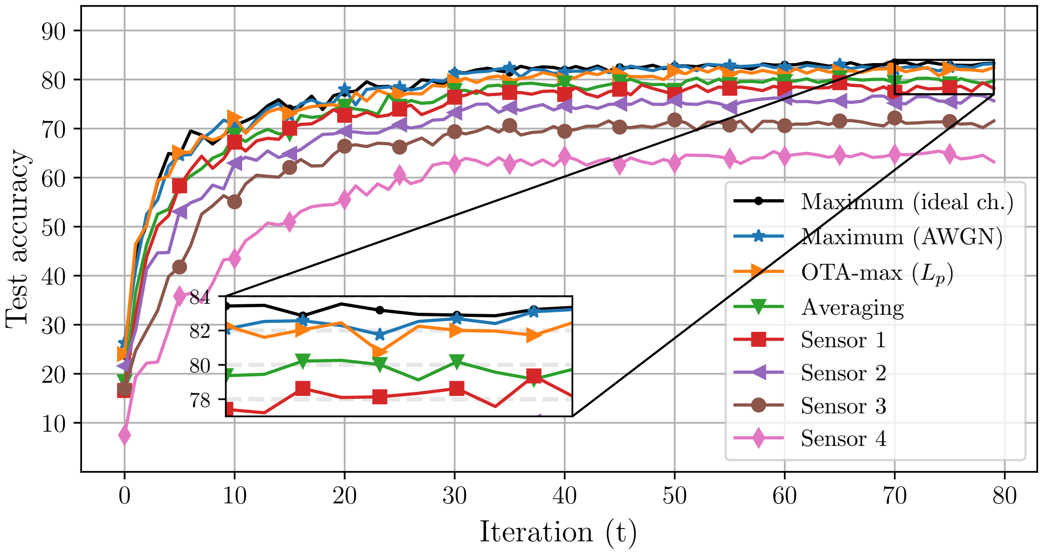


Figure 5.4: Inference accuracy for the ModelNet dataset with a multi-modal system with $M = 4$ sensor each having different computational capabilities.

In this simulation setup, the complexity of branches is adjusted according to the sensor capabilities, i.e., we assume that the first sensor is the most powerful one in terms of computational capabilities while sensor four has the lowest capability. As shown in Fig. 5.4, the proposed over-the-air approximation achieves a performance close to that of the exact maximum, which can be considered as a performance upper bound for the given system model, while significantly outperforming the sensor fusion with averaging. Unlike the previous simulations, the accuracy of single-sensor setups varies due to the multi-modality in the front-end network architectures and depends on the computational capabilities of the corresponding sensor, i.e., the complexity of the network branch. However, it is clear that sensor fusion helps improve the performance of all the sensors, with the accuracy of the one with the lowest capacity increasing the most, while still offering significant improvement for the most powerful one.

We further note that a significant reduction in transmission rate with sensor fusion for both unimodal and multi-modal wireless inference on ModelNet dataset is also observed. The input image size is $224 \times 224 \times 3 = 150528$ pixels, which would need to be transmitted separately by $M = 12$ and $M = 5$ different sensors for unimodal and multi-modal structures, respectively. However, with the proposed approach, the transmitted vector size $\mathbf{F}_m$ is reduced to 23328 and 2048, respectively, representing a decrease of 98.71% and 99.73% compared to transmitting $150528 \times M$ pixel values. While this approach provides a highly efficient transmission for multi-sensor wireless inference, we emphasize that the reduction in transmission rates depends on various parameters such as the network structure, the number of sensors, and the input size, and it may differ for other setups.

5.4 Chapter Summary

We study a multi-sensor wireless inference network that utilizes transformation-invariant over-the-air feature fusion during intermediate feature transmission. Due to the limited computational capabilities of simple sensor devices, we split the network branches between a multi-view sensor system and a central device, and propose LogSumExp and L_p-norm inspired approximations for maximum operation in an over-the-air manner to implement a transformation-invariant sensor fusion. Our numerical experiments on rotated MNIST and ModelNet dataset validate that the newly proposed approach achieves nearly the same inference accuracy level as that of the exact maximum combining, while reducing the communication load and computations at the sensor-side with the help of over-the-air transmission over wireless channels.

Chapter 6

Learnable Sensor Fusion for Multi-Sensor Wireless Inference Networks

Due to their superior performance in inference and classification, deep neural networks have garnered significant attention. However, computationally limited simple IoT sensors may not be capable of performing all the required operations for inference. Therefore, as a solution, one may divide the network into two parts: the front-end and the back-end. The data is initially sensed by a sensor and processed through the front-end, after which it is transmitted to a central processing device to complete the inference. As an additional improvement, one can also consider a system where multiple sensors collect data for the same object, inherently introducing data augmentation. In multi-sensor wireless inference setups, sensor fusion plays a crucial role as it affects both the accuracy and the transmission and computational costs of the inference process.

In this chapter, we investigate sensor networks in which multiple sensors gather data from overlapping regions and perform inference on a shared phenomenon, as in Chapter 5, which exploits transformation-invariant sensor fusion. However, practical implementation of transformation-invariant features encounters challenges, such as the need to position sensors throughout the entire scene and to identify a set of transformations that satisfy the necessary properties. These limitations impose constraints on wireless inference systems, and even with transformation-invariant features, achieving the highest accuracy is not guaranteed. To address these issues, we propose to use a trainable sensor fusion function to optimize feature fusion and enhance the overall inference accuracy. This function incorporates a trainable parameter that provides flexibility in sensor fusion based on system properties. By adjusting this parameter, a range of behaviors can be captured, from feature averaging to approximating the maximum operation.

The chapter is organized as follows. Section 6.1 introduces the system model. Learnable sensor fusion for multi-sensor wireless inference is introduced in Section 6.2. Performance of the proposed approach is studied via simulations in Section 6.3, and the chapter is concluded in Section 6.4.

6.1 System Model

We investigate a wireless sensor network comprising M sensors, each capturing observations from a common object. These observations can take various modalities, such as text, visual data, or audio. Although it is feasible to explore multi-modal sensor networks that handle data with different modalities, for the sake of simplicity, we concentrate on single-modal data. The multi-sensor architecture inherently introduces data augmentation to the network, and integrating the sensor data can enhance the reliability of inference.

We introduce an over-the-air learnable sensor fusion method, illustrated in Fig. 6.1. Each sensor, denoted as $m \in \{1, \cdots, M\}$, acquires an intermediate feature vector, $\mathbf{F}_m$, by processing its own sensed data using an embedded front-end network. Our approach draws inspiration from a sensor fusion method similar to the one presented in [12], where an L_p-norm inspired fusion technique is employed. However, we innovate by introducing a learnable parameter, p, into the preprocessing function $\phi(\cdot)$. This parameterization enables the optimization of

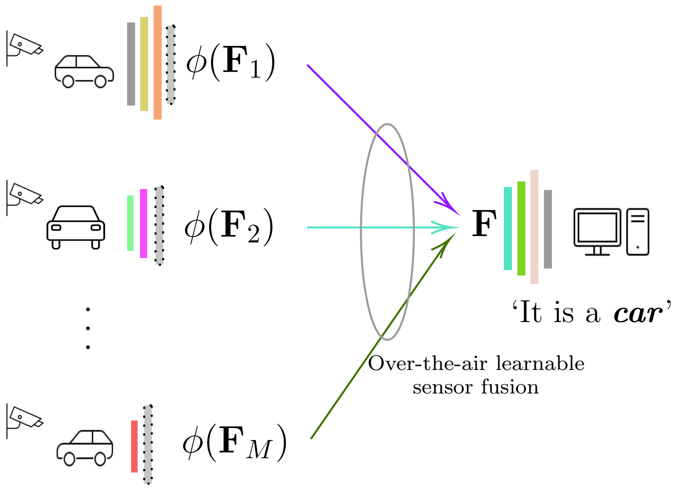


Figure 6.1: System model for the proposed learnable sensor fusion for multi-sensor wireless inference.

the sensor fusion method to better suit the characteristics of the network and sensor distribution.

For inference on the data collected by the sensors, each sensor transmits its processed features via the front-end network to a central processing device to complete the inference, utilizing an AWGN MAC. Leveraging the superposition property of the MAC channel, these transmissions occur concurrently within a shared time/frequency slot, allowing for over-the-air transmission. This approach ensures the system's transmission efficiency. We further introduce a learnable parameter to the sensor fusion method to enhance the adaptability of the system by customizing it to the specific network and sensor distribution.

6.2 Learnable Sensor Fusion for Multi-Sensor Networks

In [126, 127], it is suggested that transformation-invariant features can be derived from multiple data sources through a maximum operation, provided that all potential input transformations form a group adhering to fundamental properties such as closure, associativity, invertibility, and identity. However, practically situating sensors throughout the entire scene to meet these conditions may be infeasible. Furthermore, identifying a set of transformations that fulfills all the required properties can be a daunting task, especially given the potentially vast number of possible transformations. These limitations impose practical constraints on wireless inference setups. Moreover, it is essential to note that even with transformation-invariant features, achieving the highest possible inference accuracy is not guaranteed.

Given these considerations, we propose employing a trainable sensor fusion to optimize the feature fusion step and enhance the overall inference accuracy. In this step, we utilize an L_p-norm inspired sensor fusion method, similar to the one introduced in [12], where sensor m transmits its feature vector $\mathbf{F}_m$ as follows:

$$\mathbf{x}_m = \phi(\mathbf{F}_m) = \mathbf{F}_m^p, \tag{6.1}$$

where $\mathbf{F}_m$ is the feature vector which can be 1D vectors, 2D matrices or 3D tensors depending on the network structure and layers. For simplicity, we assume $\mathbf{F}_m \in \mathbb{R}^d$.

With over-the-air transmission, the received signal will be superposition of sensor data from M sensors, which is given by

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{x}_m + \mathbf{n} = \sum_{m=1}^{M} \mathbf{F}_m^p + \mathbf{n}, \tag{6.2}$$

where $\mathbf{y} \in \mathbb{R}^d$, and $\mathbf{n} \in \mathbb{R}^d$ is the AWGN noise vector with variance σ_n^2.

At the central processing device, this signal will be used utilizing a preprocessing function

$$y(i)^{\frac{1}{p}} = \left(\sum_{m=1}^{M} F_m(i)^p + n(i) \right)^{\frac{1}{p}}, \tag{6.3}$$

for the i-th element of the received signal. Note that for constant $p \geq 0$, this preprocessing function is as discussed in [12]. However, by introducing the trainable parameter p instead of a constant one, we are able to capture a spectrum of behaviors by ensuring flexibility for sensor fusion based on the system properties. Specifically, when $p = 1$, the function corresponds to feature summation, which conveys the same information as averaging, while for sufficiently large values of p, it approximates the maximum operation. During training, this parameter can converge to an optimal value that maximizes the overall inference accuracy based on the specific setup.

Remark 5. *We note that, due to the computational limitations of sensors, we consider offline training, and the network parameters are shared with the sensors and the central processing device after training. These parameters are used during the online wireless inference process for real-time applications.*

Remark 6. *During the offline training, we introduce the parameter p of the sensor fusion method as a trainable network parameter. Consequently, this parameter, in conjunction with the network parameters, undergoes adjustments to minimize a suitable loss function, thereby fine-tuning the entire network. This optimization process depend on various aspects, including the number of sensors, their quality, placement, and the sensor fusion method, all adjusted for the specific characteristics of the sensor network. After the offline training process, the resulting value of parameter p is fixed, and this fixed value is used during the real-time online inference process.*

6.3 Numerical Examples

6.3.1 Dataset Descriptions

In our numerical examples, we utilize two datasets: 1) a custom-made Car Learning to Act (CARLA) dataset, and 2) the ModelNet dataset to evaluate the performance of learnable sensor fusion for multi-sensor wireless inference.

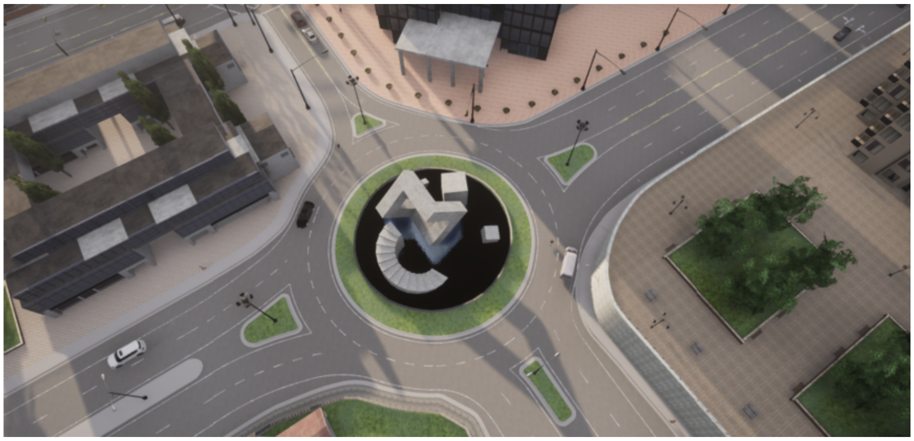


Figure 6.2: Roundabout simulation environment in CARLA.

CARLA: Wireless inference systems have gained traction in various domains, from autonomous driving to surveillance. These systems rely on sensor data fusion for accurate environmental interpretation. However, evaluating these systems in real-world scenarios is challenging due to complexity and cost. To address this, we use the CARLA platform [133] to create a simulation environment. Our setup includes a roundabout scenario with eight strategically placed cameras (as in Fig. 6.2). We use data from these cameras to construct a robust dataset that simulates wireless communication challenges like occlusions, signal variations, and interference. This dataset is crucial for training and evaluating wireless inference algorithms, enabling accuracy, robustness, and efficiency analysis in a simulated, yet highly realistic, environment. It also allows for algorithm comparisons, suitability assessments, and the development of new approaches.

In this environment, we collect training and test data for five classes: pedestrians, small vehicles, large vehicles, bicycles, and motorcycles. Table 6.1 summarizes the dataset properties. It is important to note that although there are eight cameras positioned around the roundabout, some detected objects may only be sensed by a subset of the cameras. Consequently, the number of collected images for a particular object class could be less than eight.

ModelNet: The Princeton ModelNet dataset, specifically its 40-class subset [132], originally provides 3D CAD models of objects. These models have been

Table 6.1: The dataset properties for the custom-made CARLA dataset.

Class	Train sample size	Test sample size
Bicycle	220	108
Motorcycle	170	64
Large vehicle	250	131
Small vehicle	260	118
Pedestrian	180	58

processed using the method proposed in [128] to obtain 2D samples from a 12-view camera setup. In our simulations, we utilize this 2D dataset [128] to assess the performance of our proposed approach.

6.3.2 Numerical Results

Table 6.2: Network architecture for the learnable sensor fusion for multi-sensor wireless inference.

Front-end (Sensor side)	Image: $224 \times 224 \times 3$
	3×3 convolutional layer, 32 channels, ReLU, stride:4, padding: 2
	2D maxpooling with kernelsize=3, stride=2
	dropout with probability 0.5
Back-end (Device-size)	fully connected layer (23328, 8196) ReLU
	dropout with probability 0.5
	fully connected layer (8196, 2048) ReLU
	Output layer: fully connected layer (2048, 40)

In the numerical examples, we adopt the network structure outlined in Table 6.2 and employ $M = 5$ sensors for both the custom-made CARLA and ModelNet datasets. During offline training, we utilize a mini-batch size of 10. The training process utilizes the Adam optimizer [118] with an initial learning rate of 10^{-4} and runs for 80 iterations. For the results presented in this study, we conduct training for 10 models and report the average test accuracy along with the associated one-standard-deviation interval. It is worth noting that both offline training and real-time inference involve a wireless channel connecting the sensors to the central processing device, modeled as an AWGN MAC.

For the sensor fusion using the L_p-norm inspired function, we initialize the learnable parameter p as $0.95 + U[0, 0.1]$, where $U[0, 0.1]$ represents a uniform distribution between 0 and 0.1. This proposed approach is compared with three baselines: 1) sensor fusion by taking the exact maximum of all sensor's transmitted features, 2) feature averaging, 3) using only one sensor without sensor fusion. Taking the exact maximum requires recovering all transmitted features from different sensors at the receiver side, necessitating orthogonal transmission and making the process costly and undesirable. On the other hand, both the learnable sensor fusion with the L_p-norm inspired function and feature averaging can be performed in an over-the-air manner with concurrent transmission, making them transmission efficient. Furthermore, it is worth noting that with the help of the trainable parameter p in learnable sensor fusion, one can cover both ends of the spectrum and approximate both averaging and exact maximum by adjusting p values. Hence, the sensor fusion can be optimized for the given sensor network and data structure.

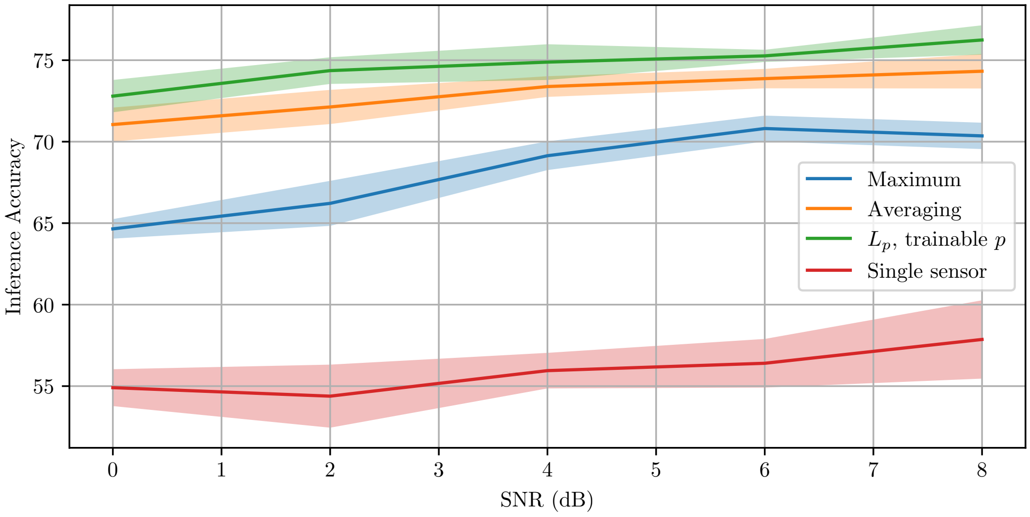


Figure 6.3: Inference accuracy for learnable sensor fusion using CARLA dataset with $M = 5$ sensors with perfect (fixed) SNR training.

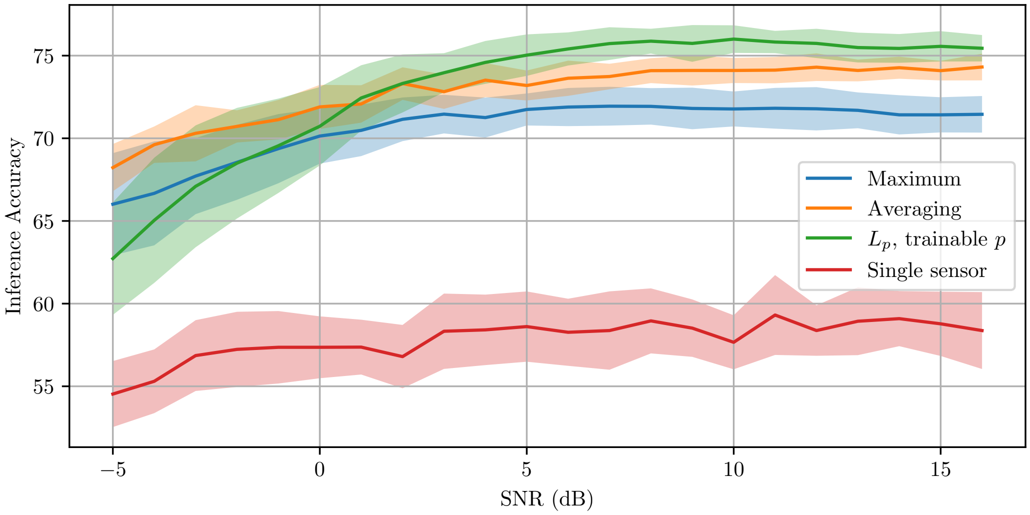


Figure 6.4: Inference accuracy for learnable sensor fusion using CARLA dataset with $M = 5$ sensors with SNR-robust training.

In Fig. 6.3, we examine the perfectly known SNR scenario, where we assume that the channel SNR is available both during offline training and real-time processing. Consequently, we train and test the proposed approach and the baselines with exactly the same SNR settings. Specifically, we train multiple models with $\{0, 2, 4, 6, 8\}$ dB SNRs and test them with matching SNRs. As anticipated, using only one sensor without sensor fusion has the poorest performance, since it cannot exploit multiple data sources for the detected object. Moreover, the L_p-norm inspired sensor fusion achieves superior performance compared to both averaging and exact maximum across the entire SNR range. This observation highlights the benefits of introducing the learnable parameter p, which enables a sensor fusion method capable of generalizing better than averaging or exact maximum operations for the given setup. In this setup, it is noteworthy that the learnable parameter p converges to approximately 0.88, an average taken over 10 models. Consequently, despite the final fusion method exhibiting different characteristics from a simple averaging, it is more closer to the averaging function (not to the maximum operation).

In Fig. 6.4, we investigate SNR-robust training for the same system. In this scenario, during both offline training and real-time inference, the only available information about the AWGN channel is the range of the SNR. The exact statistics of the channel is unknown. To address this, during each iteration of the training process, we sample an SNR value between -5 dB and 15 dB (uniformly sampled in linear scale) and perform the training accordingly. Consequently, even though we lack perfect knowledge of the SNR, we can train a single network to accommodate the wide SNR range. As depicted in Fig. 6.4, especially for moderate to high SNR values, our previous observation holds, confirming that the L_p-norm inspired sensor fusion with the trainable parameter remains a superior method compared to averaging and exact maximum. However, for low SNR values, e.g., for SNRs lower than 1 dB, averaging exhibits a slightly better performance compared to the proposed approach. Furthermore, compared to the previous results, one does not need to train multiple models for each specific channel quality when employing the SNR-robust training approach. Instead, a single model is trained to adapt to the entire range of SNR values, making the system more versatile and efficient.

In Fig. 6.5, we provide a comparison of different training strategies for the learnable sensor fusion with the L_p norm:

- Strategy 1: Training with a pre-chosen (single) fixed SNR (selected as 4 dB),
- Strategy 2: SNR-robust training with uniformly sampled SNR from 0 to 8 dB during each iteration, as described for the setup in Fig. 6.4,
- Strategy 3: Training with perfect (fixed) SNR, as described for the setup in Fig. 6.3.

Notably, Strategies 1 and 2 only require training a single model, while Strategy 3 requires the training of different models for each SNR value. As expected, Strategy 3 achieves better performance across almost all SNR values since it is specifically trained for each SNR, maximizing performance for each scenario. Conversely, Strategy 1 performs poorly for SNRs different from 4 dB, as it is optimized for that specific SNR only. On the other hand, Strategy 2, with SNR-robust training, demonstrates reasonable performance for the given SNR range,

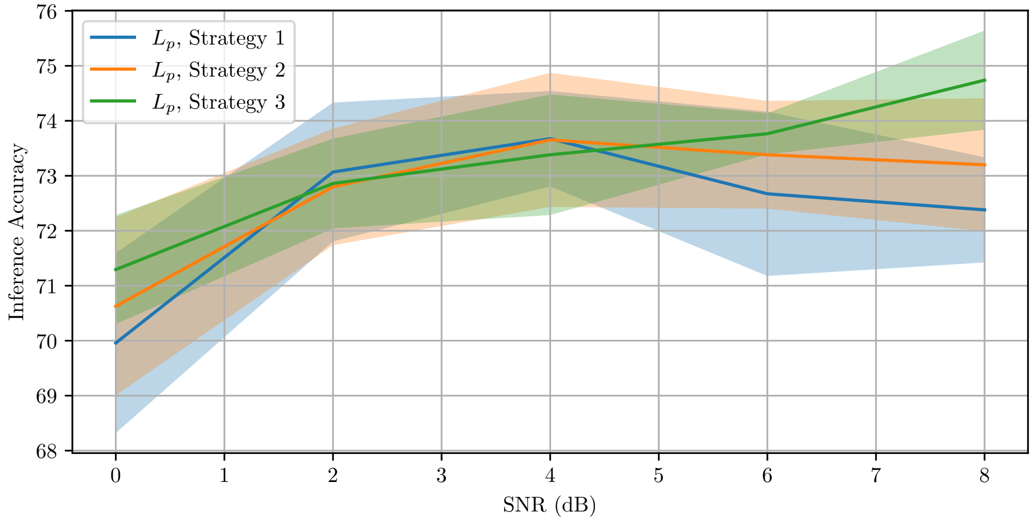


Figure 6.5: Comparison of inference accuracy for learnable sensor fusion using CARLA dataset with $M = 5$ for three training strategies.

although it is slightly inferior compared to Strategy 3 with perfect SNR knowledge. However, when considering the trade-off between training cost and inference accuracy, Strategy 2 with robust training proves to be a desirable approach. The SNR-robust training allows for a single model to be deployed effectively across a range of SNR values, reducing the need for multiple specialized models and optimizing the training cost. Furthermore, with this approach, the memory requirement of the sensor devices are significantly reduced helping to obtain low-cost network design.

In Fig. 6.6, we adopt the robust-training strategy for the ModelNet dataset with $M = 5$ sensors, considering an SNR range between -5 and 15 dB. Unlike the CARLA simulations, in this case, sensor fusion with the exact maximum performs better than averaging for the high SNR range, while averaging outperforms the exact maximum for lower SNRs. However, for SNRs higher than approximately 1 dB, the learnable sensor fusion with the L_p-norm inspired function achieves superior performance. This observation indicates that the order of performance between averaging and exact maximum can vary based on the dataset, sensor distribution, or other system parameters. Nevertheless, with the learnable sensor fusion utilizing the L_p-norm inspired function, one can obtain a generalizable

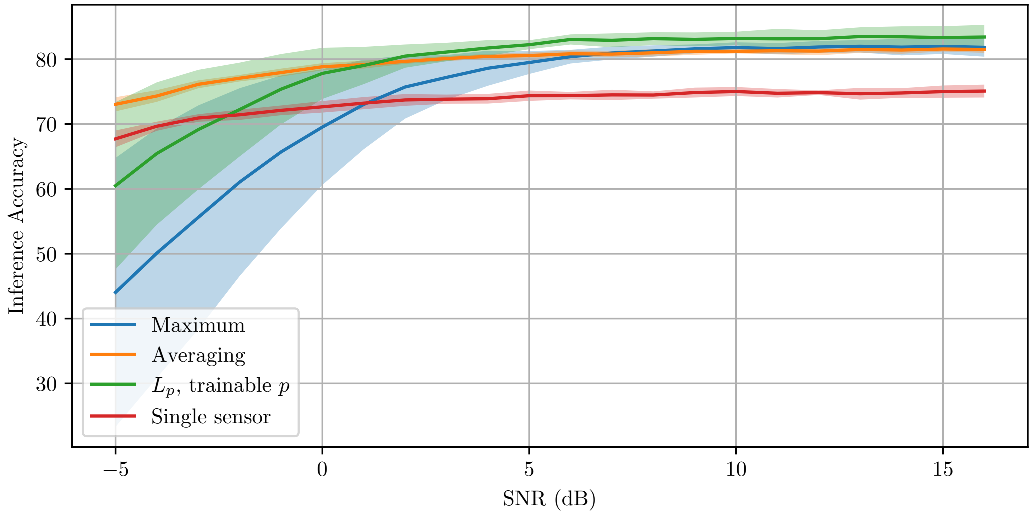


Figure 6.6: Inference accuracy for learnable sensor fusion using ModelNet dataset with $M = 5$ sensors with SNR-robust training.

sensor fusion strategy, particularly beneficial for moderate to high SNR ranges. This demonstrates the adaptability and effectiveness of the proposed approach, making it suitable for diverse scenarios and system configurations.

6.4 Chapter Summary

In this chapter, we have investigated a learnable sensor fusion method for multi-sensor wireless networks. This method employs an adaptable over-the-air combining function resembling the L_p-norm, controlled by a learnable parameter. This parameter enables the deep neural networks to dynamically adjust their sensor fusion method, encompassing a wide range from averaging to maximum, suiting diverse scenarios and system configurations. Additionally, our approach exhibits robustness to SNR fluctuations, thereby reducing the offline training cost and memory demands of the sensor network. Overall, our method offers an efficient and flexible solution for optimizing sensor network performance, tailoring the fusion process to different application requirements.

Chapter 7

Conclusions and Future Work

We conclude the thesis by summarizing our contributions and outlining several avenues for future research.

In Chapter 3, we have addressed practical implementation challenges in the context of federated learning over wireless channels. In our proposed system model, we have characterized the channel as a frequency-selective channel and employed OFDM for transmission. The workers perform local iterations and transmit corresponding gradients using low-resolution DACs, while the PS employes ADCs at the receive antennas. The transmissions occur in an over-the-air manner to maximize bandwidth efficiency. We have demonstrated both theoretically and empirically that the use of low-resolution DACs and ADCs, including one-bit DACs and ADCs, does not impede the performance of federated learning algorithms. Furthermore, we have shown that multipath channel effects diminish when a sufficient number of antennas are employed at the PS.

Addressing another practical concern for FL, Chapter 4 has been dedicated to an exploration of federated learning over time-varying channels. Local model updates are transmitted through time-varying channels, utilizing an OFDM-based approach over multiple access channels. Such channel variations can exist in numerous real-world scenarios, e.g., when there are workers or parameter server mobility, leading to the inter-carrier interference that disrupts the orthogonality of subcarriers. We have conducted an analytical investigation to illustrate that the resulting inter-carrier interference does not impede the convergence of the learning algorithm, particularly in cases of slow to moderately varying channels. These findings have been extensively validated through numerical simulations.

The first part of the thesis has been centered on addressing practical implementation issues within the context of FL for realistic scenarios. We highlight that there is room for further enhancement across various dimensions. Firstly, FL presents solutions with broad applicability, e.g., autonomous driving, medical diagnosis, and virtual assistants. To illustrate the utility of FL in these real-world systems, one can assess the effectiveness of the models employed using more realistic datasets. For instance, one could experiment with the CARLA dataset for FL with autonomous driving applications, providing a more accurate representation of some real-world scenarios.

Moreover, in Chapter 3, we explored hardware impairments resulting from low-resolution DACs and ADCs in the context of FL models. However, it is important to note that other components, e.g., amplifiers, filters, mixers, and antennas, within a practical system can also contribute to impairments and introduce noise or distortion that may degrade the overall learning performance. Therefore, as a complementary research direction, one can also investigate and analyze effects of non-idealities in other components of federated learning systems over wireless channels.

In Chapter 5, we have introduced a multi-sensor wireless inference system in which the sensors observe an overlapping region assigned to perform real-time applications. This setup inherently introduces data augmentation, which improves the overall inference performance, but also requires careful deployment of sensor fusion due to the involvement of multiple sensors. For the introduced setup, we have proposed a bandwidth-efficient over-the-air sensor fusion method by approximating the transformation-invariant maximum operation using L_p-norm inspired and LogSumExp functions. We have demonstrated numerically that the newly proposed solution improves overall performance while significantly reducing transmission costs facilitated by over-the-air transmission.

As an extension and further improvement of the aforementioned multi-sensor setup, in Chapter 6, we have proposed a trainable sensor fusion method that introduces a learnable parameter to the L_p-norm inspired sensor fusion function. This parameter can approximate several sensor fusion methods, e.g., averaging and maximum operations, by optimizing its value during offline training. Additionally, it allows for customization of sensor fusion to match the characteristics of the overall system, taking into account factors like the number of sensors, their placement, data modality, and quality.

Several avenues for improvement and further exploration can be pursued based on our findings. An important direction for future work is to delve deeper into the underlying theoretical aspects of the proposed sensor fusion approach. Conducting a comprehensive convergence analysis can provide insights into the behavior and stability of the fusion function during the training process. This analysis can involve exploring the conditions under which the fusion function converges to an optimal solution, studying the impact of different loss functions or regularization terms on convergence, and understanding the relationship between the trainable parameter and convergence properties.

To assess the generalization capabilities of the proposed multi-sensor wireless inference setup, it is essential to evaluate its performance on diverse datasets beyond the ones used in this thesis. This can involve collecting or acquiring datasets from different environments, varying sensor configurations, and distinct scenarios. Evaluating the performance of the fusion function on these datasets will provide insights into its robustness, adaptability, and ability to handle varying conditions, thereby contributing to its applicability in real-world scenarios.

Another important aspect for future work is the incorporation of multi-modal data, such as radar, into the multi-sensor wireless inference setup. In our work, we only focused on the fusion of RGB data from different cameras. However, integrating additional modalities can provide complementary information and enhance the overall inference accuracy.

Lastly, classical information theory, as originally introduced by Shannon in [134], predominantly addresses technical communication problems, often regarding the meaning behind transmitted symbols as irrelevant for engineering purposes. Weaver, in [135], expanded upon this by presenting a three-level communication model: Level A, the technical problem; Level B, the semantic problem, which delves into the meaning behind transmitted symbols; and Level C, the effectiveness problem, which is concerned with the desired recipient behavior. While the technical problem aligns with classical information theory, the semantic problem focuses on understanding the meaning of transmitted sequence. The effectiveness problem seeks to ensure the desired actions on the receiver side. Numerous studies have aimed to enhance semantic information theory and develop models for reliable semantic communication, as highlighted in [136, 137, 138]. More recently, in [139], a novel goal-oriented semantic communication framework has been introduced. This framework is based on a graph-based language and specializes in mapping meanings to a predefined syntactic structure, enabling the definition of suitable performance metrics and the implementation of compression/coding schemes based on the conveyed or inferred meaning of transmissions. The introduced language model employs attribute sets to capture additional properties and features, such as those related to detected objects in computer vision applications.

Building on semantic and goal oriented communications, in the context of wireless inference, intermediate features may convey the semantic content of data sensed by different devices. In this setting, the primary objective is not necessarily the perfect recovery of feature vectors, but rather the retrieval of the underlying meaning of these feature vectors, such as the inference result in a classification task. Consequently, there is an opportunity to design a semantic language for intermediate features or to effectively leverage existing semantic language definitions within the context of wireless inference. We perceive this as a promising line of research particularly for IoT and smart environment applications. Addressing these future research directions will not only refine the learnable sensor fusion approach but also deepen our understanding of its theoretical foundations, validating its efficiency, generalization capabilities across diverse datasets and robustness to hardware impairments. Such advancements will contribute to the development of more sophisticated and reliable multi-sensor wireless inference systems in practical applications.

Bibliography

[1] X. Chen and Q. Qi, *Convergence of Energy, Communication and Computation in B5G Cellular Internet of Things*, pp. 111–122. Singapore: Springer Singapore, Apr. 2020.

[2] L. Fan, S. Jin, C.-K. Wen, and H. Zhang, "Uplink achievable rate for massive MIMO systems with low-resolution ADC," *IEEE Communications Letters*, vol. 19, pp. 2186–2189, Dec. 2015.

[3] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, pp. 7–12, Mar. 1960.

[4] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] J. Karhunen, T. Raiko, and K. Cho, "Unsupervised deep learning: A short review," *Advances in independent component analysis and learning machines*, pp. 125–142, May 2015.

[7] L. Zeng, E. Li, Z. Zhou, and X. Chen, "Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the industrial internet of things," *IEEE Network*, vol. 33, pp. 96–103, Sep–Oct 2019.

[8] B. Tegin and T. M. Duman, “Blind federated learning with low-cost analog-to-digital converters,” in *2021 IEEE Global Communications Conference (GLOBECOM)*, (Madrid, Spain), pp. 01–06, Dec. 2021.

[9] B. Tegin and T. M. Duman, “Blind federated learning at the wireless edge with low-resolution ADC and DAC,” *IEEE Transactions on Wireless Communications*, vol. 20, pp. 7786–7798, Dec. 2021.

[10] B. Tegin and T. M. Duman, “Federated learning over time-varying channels,” in *IEEE Global Communications Conference (GLOBECOM)*, (Madrid, Spain), pp. 01–06, Dec. 2021.

[11] B. Tegin and T. M. Duman, “Federated learning with over-the-air aggregation over time-varying channels,” *IEEE Transactions on Wireless Communications*, vol. 22, pp. 5671–5684, Aug. 2023.

[12] B. Tegin and T. M. Duman, “Transformation-invariant over-the-air combining for multi-sensor wireless inference,” in *IEEE Global Communications Conference (GLOBECOM), accepted for presentation*, (Kuala Lumpur, Malaysia), Dec. 2023.

[13] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, Dec. 1943.

[14] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, (Sardinia, Italy), pp. 249–256, JMLR Workshop and Conference Proceedings, May 2010.

[15] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” *Advances in neural information processing systems*, vol. 27, Dec. 2014.

[16] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, pp. 2295–2329, Dec. 2017.

[17] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: A manifold disentanglement perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, pp. 1997–2008, Oct. 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, June 2016.

[19] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Joint device-edge inference over wireless links with pruning," in *IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, (Atlanta, GA, USA), pp. 1–5, May 2020.

[20] R. C. Buck, *Approximate complexity and functional representation*, vol. 70. University of Wisconsin-Madison. Mathematics Research Center, July 1976.

[21] W. Liu, X. Zang, Y. Li, and B. Vucetic, "Over-the-air computation systems: Optimization, analysis and scaling laws," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 5488–5502, Aug. 2020.

[22] H. V. Poor, *An introduction to signal detection and estimation.* Springer Science & Business Media, 1998.

[23] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project adam: Building an efficient and scalable deep learning training system," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, (Broomfield, CO, USA), pp. 571–582, Oct. 2014.

[24] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 20–22 Apr 2017.

[26] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, Jan. 2021.

[27] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling Up Machine Learning: Parallel and Distributed Approaches*. UK: Cambridge University Press, 2011.

[28] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[29] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 491–506, Jan. 2020.

[30] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 3546–3557, May 2020.

[31] Y.-S. Jeon, M. Mohammadi Amiri, and N. Lee, "Communication-efficient federated learning over mimo multiple access channels," *IEEE Transactions on Communications*, vol. 70, pp. 6547–6562, Oct. 2022.

[32] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, pp. 1935–1949, Nov. 2021.

[33] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, pp. 2457–2471, Apr 2021.

[34] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, (Long Beach, CA, USA), pp. 1709–1720, Dec. 2017.

[35] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *15th Annual Conference of the International Speech Communication Association*, (Singapore), pp. 1058–1062, Sept. 2014.

[36] S.-Y. Zhao, H. Gao, and W.-J. Li, "Quantized Epoch-SGD for communication-efficient distributed learning," *arXiv preprint arXiv:1901.03040*, 2019.

[37] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," *arXiv preprint arXiv:2006.10672*, 2020.

[38] M. M. Amiri, T. M. Duman, and D. Gündüz, "Collaborative machine learning at the wireless edge with blind transmitters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, (Ottowa, ON, Canada), pp. 1–5, Nov. 2019.

[39] M. M. Amiri, T. M. Duman, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Blind federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 20, pp. 5129–5143, Aug. 2021.

[40] S. Li, S. M. M. Kalan, A. S. Avestimehr, and M. Soltanolkotabi, "Near-optimal straggler mitigation for distributed gradient methods," in *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, (Vancouver, BC, Canada), pp. 857–866, May 2018.

[41] P. Blanchard, R. Guerraoui, J. Stainer, *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, (Long Beach, CA, USA), pp. 118–128, Dec. 2017.

[42] Y. Tao, S. Cui, W. Xu, H. Yin, D. Yu, W. Liang, and X. Cheng, "Byzantine-resilient federated learning at edge," *IEEE Transactions on Computers*, vol. 72, pp. 2600–2614, Sept. 2023.

[43] T. Jahani-Nezhad, M. A. Maddah-Ali, and G. Caire, "Byzantine-resistant secure aggregation for federated learning based on coded computing and vector commitment," *arXiv preprint arXiv:2302.09913*, 2023.

[44] S. Huang, Y. Zhou, T. Wang, and Y. Shi, "Byzantine-resilient federated machine learning via over-the-air computation," in *2021 IEEE International*

Conference on Communications Workshops (ICC Workshops), (Montreal, Canada), pp. 1–6, June 2021.

[45] B. Zhao, P. Sun, T. Wang, and K. Jiang, “Fedinv: Byzantine-robust federated learning by inversing local model updates,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 9171–9179, Feb–Mar 2022.

[46] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J.-S. Weng, “BDFL: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle,” *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 8639–8652, Sept. 2021.

[47] R. Jin, J. Hu, G. Min, and H. Lin, “Byzantine-robust and efficient federated learning for the internet of things,” *IEEE Internet of Things Magazine*, vol. 5, pp. 114–118, Mar. 2022.

[48] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, “On the byzantine robustness of clustered federated learning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Barcelona, Spain), pp. 8861–8865, May 2020.

[49] H. Guo, H. Wang, T. Song, Y. Hua, Z. Lv, X. Jin, Z. Xue, R. Ma, and H. Guan, “Siren: Byzantine-robust federated learning via proactive alarming,” in *Proceedings of the ACM Symposium on Cloud Computing*, (New York, NY, USA), p. 47–60, Association for Computing Machinery, 2021.

[50] J. Song, W. Wang, T. R. Gadekallu, J. Cao, and Y. Liu, “EPPDA: An efficient privacy-preserving data aggregation federated learning scheme,” *IEEE Transactions on Network Science and Engineering*, vol. 10, pp. 3047–3057, Sep-Oct 2023.

[51] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, “ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1639–1654, 2022.

[52] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, “An adaptive federated learning scheme with differential privacy preserving,” *Future Generation Computer Systems*, vol. 127, pp. 362–372, Feb. 2022.

[53] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, “A hybrid approach to privacy-preserving federated learning,” in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, (New York, NY, USA), p. 1–11, Association for Computing Machinery, 2019.

[54] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[55] A. E. Ouadrhiri and A. Abdelhadi, “Differential privacy for deep and federated learning: A survey,” *IEEE Access*, vol. 10, pp. 22359–22380, 2022.

[56] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, and H. V. Poor, “Personalized federated learning with differential privacy and convergence guarantee,” *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4488–4503, 2023.

[57] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, “Convergence of federated learning over a noisy downlink,” *IEEE Transactions on Wireless Communications*, vol. 21, pp. 1422–1437, Mar. 2022.

[58] Z. Yan, D. Li, X. Yu, and Z. Zhang, “Latency-efficient wireless federated learning with quantization and scheduling,” *IEEE Communications Letters*, vol. 26, pp. 2621–2625, Nov. 2022.

[59] R. Hönig, Y. Zhao, and R. Mullins, “DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research*, pp. 8852–8866, PMLR, July 2022.

[60] Y. Mao, Z. Zhao, G. Yan, Y. Liu, T. Lan, L. Song, and W. Ding, "Communication-efficient federated learning with adaptive quantization," *ACM Trans. Intell. Syst. Technol.*, vol. 13, pp. 1–26, Aug 2022.

[61] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "Green, quantized federated learning over wireless networks: An energy-efficient design," *IEEE Transactions on Wireless Communications*, early access, 2023.

[62] Y. Liu, S. Rini, S. Salehkalaibar, and J. Chen, "M22: A communication-efficient algorithm for federated learning inspired by rate-distortion," *arXiv preprint arXiv:2301.09269*, 2023.

[63] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Barcelona, Spain), pp. 8851–8855, May 2020.

[64] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031, PMLR, Aug. 2020.

[65] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary compression for communication-efficient federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 1162–1176, Mar. 2022.

[66] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 500–514, 2021.

[67] O. Aygün, M. Kazemi, D. Gündüz, and T. M. Duman, "Over-the-air federated learning with energy harvesting devices," in *IEEE Global Communications Conference*, (Rio de Janeiro, Brazil), pp. 1942–1947, Dec. 2022.

[68] O. Aygün, M. Kazemi, D. Gündüz, and T. M. Duman, "Hierarchical over-the-air federated edge learning," in *IEEE International Conference on Communications (ICC)*, (Seoul, Korea, Republic of), pp. 3376–3381, May 2022.

[69] T. Wang, Y. Liu, X. Zheng, H.-N. Dai, W. Jia, and M. Xie, “Edge-based communication optimization for distributed federated learning,” *IEEE Tran. on Network Science and Engineering*, vol. 9, pp. 2015–2024, July-Aug. 2022.

[70] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, “Hierarchical federated learning across heterogeneous cellular networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Barcelona, Spain), pp. 8866–8870, May 2020.

[71] C. Briggs, Z. Fan, and P. Andras, “Federated learning with hierarchical clustering of local updates to improve training on non-IID data,” in *IEEE International Joint Conference on Neural Networks (IJCNN)*, (Glasgow, UK), pp. 1–9, July 2020.

[72] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, “Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, pp. 536–550, Mar. 2022.

[73] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, “To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices,” in *IEEE International Conference on Computer Communications (INFOCOM’21)*, (Vancouver, BC, Canada), May 2021.

[74] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, “Federated learning over wireless networks: Convergence analysis and resource allocation,” *IEEE/ACM Transactions on Networking*, vol. 29, pp. 398–409, Feb 2021.

[75] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, “Scheduling policies for federated learning in wireless networks,” *IEEE Transactions on Communications*, vol. 68, pp. 317–333, Jan. 2020.

[76] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, “Convergence of update aware device scheduling for federated learning at the wireless edge,”

IEEE Transactions on Wireless Communications, vol. 20, pp. 3643–3658, June 2021.

[77] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *IEEE International Conference on Communications (ICC)*, (Shanghai, China), pp. 1–7, May 2019.

[78] Y. J. Cho, J. Wang, and G. Joshi, “Client selection in federated learning: Convergence analysis and power-of-choice selection strategies,” *arXiv preprint arXiv:2010.01243*, 2020.

[79] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Transactions on Wireless Communications*, vol. 20, pp. 1188–1200, Feb. 2021.

[80] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, “An efficiency-boosting client selection scheme for federated learning with fairness guarantee,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, pp. 1552–1564, July 2021.

[81] H. H. Yang, A. Arafa, T. Q. Quek, and H. V. Poor, “Age-based scheduling policy for federated learning in mobile edge networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Barcelona, Spain), pp. 8743–8747, May 2020.

[82] Y. Zhan, P. Li, and S. Guo, “Experience-driven computational resource allocation of federated learning by deep reinforcement learning,” in *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, (New Orleans, LA, USA), pp. 234–243, May 2020.

[83] M. M. Wadu, S. Samarakoon, and M. Bennis, “Federated learning under channel uncertainty: Joint client scheduling and resource allocation,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, (Seoul, Korea (South)), pp. 1–6, May 2020.

[84] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, “Efficient federated learning algorithm for resource allocation in wireless

IoT networks," *IEEE Internet of Things Journal*, vol. 8, pp. 3394–3409, Mar. 2021.

[85] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1–8, Nov 1990.

[86] B. Hassibi, D. Stork, and G. Wolff, "Optimal brain surgeon: Extensions and performance comparisons," *Advances in Neural Information Processing Systems*, vol. 6, pp. 1–8, Jul 1993.

[87] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[88] J. Shao and J. Zhang, "BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, (Dublin, Ireland), pp. 1–6, Jun 2020.

[89] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 89–100, Jan 2021.

[90] J. Shao, H. Zhang, Y. Mao, and J. Zhang, "Branchy-GNN: A device-edge co-inference framework for efficient point cloud processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Toronto, ON, Canada), pp. 8488–8492, Jun 2021.

[91] M. Lee, G. Yu, and H. Dai, "Decentralized inference with graph neural networks in wireless communication systems," *IEEE Transactions on Mobile Computing*, vol. 22, pp. 2582–2598, May 2023.

[92] M. Lee, G. Yu, and H. Dai, "Privacy-preserving decentralized inference with graph neural networks in wireless networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, Early access, 2023.

[93] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, pp. 197–211, Dec 2022.

[94] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multi-device cooperative edge inference," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, 2023.

[95] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[96] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *IEEE Information Theory Workshop (ITW)*, (Jerusalem, Israel), pp. 1–5, Apr 2015.

[97] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proceedings of the 2018 Workshop on Mobile Edge Communications*, (Budapest Hungary), pp. 31–36, Aug 2018.

[98] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th annual international conference on mobile computing and networking*, (Los Cabos, Mexico), pp. 1–16, Aug 2019.

[99] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Computer Architecture News*, vol. 45, pp. 615–629, Mar 2017.

[100] H. Li, C. Hu, J. Jiang, Z. Wang, Y. Wen, and W. Zhu, "JALAD: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution," in *IEEE 24th international conference on parallel and distributed systems (ICPADS)*, (Singapore), pp. 671–678, Dec 2018.

[101] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *IEEE 37th international conference on distributed computing systems (ICDCS)*, (Atlanta, GA, USA), pp. 328–339, June 2017.

[102] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep over-the-air computation," in *IEEE Global Communications Conference (GLOBECOM)*, (Taipei, Taiwan), pp. 1–6, Dec. 2020.

[103] J. Li, G. Liao, L. Chen, and X. Chen, "Roulette: A semantic privacy-preserving device-edge collaborative inference framework for deep learning classification tasks," *IEEE Transactions on Mobile Computing*, early access, 2023.

[104] B. Tegin, *Distributed caching and learning over wireless channels.* MS thesis, Bilkent Universitesi (Turkey), Jan. 2020.

[105] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selelted Areas in Communications*, vol. 17, pp. 539–550, Apr. 1999.

[106] H.-S. Lee and C. G. Sodini, "Analog-to-digital converters: Digitizing the analog world," *Proceedings of the IEEE*, vol. 96, pp. 323–334, Feb. 2008.

[107] S. Wei, D. L. Goeckel, and P. A. Kelly, "Convergence of the complex envelope of bandlimited OFDM signals," *IEEE Transactions on Information Theory*, vol. 56, pp. 4893–4904, Oct. 2010.

[108] S. Jacobsson, U. Gustavsson, G. Durisi, and C. Studer, "Massive MU-MIMO-OFDM uplink with hardware impairments: Modeling and analysis," in *52nd Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA, USA), pp. 1829–1835, Oct. 2018.

[109] S. R. Aghdam and T. Eriksson, "On the performance of distortion-aware linear receivers in uplink massive MIMO systems," in *16th International Symposium on Wireless Communication Systems (ISWCS)*, (Oulu, Finland), pp. 208–212, Aug. 2019.

[110] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted Gaussian signals," Tech. Rep. 216, Research Laboratory of Electronics, Massachusetts Institute of Technology, MA, USA, Mar. 1952.

[111] O. T. Demir and E. Bjornson, "The Bussgang Decomposition of Nonlinear Systems: Basic Theory and MIMO Extensions [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 38, pp. 131–136, Jan. 2021.

[112] E. Björnson, L. Sanguinetti, and J. Hoydis, "Hardware distortion correlation has negligible impact on UL massive MIMO spectral efficiency," *IEEE Transactions on Communications*, vol. 67, pp. 1085–1098, Feb. 2018.

[113] A. K. Fletcher, S. Rangan, V. K. Goyal, and K. Ramchandran, "Robust predictive quantization: Analysis and design via convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 618–632, Dec. 2007.

[114] O. Orhan, E. Erkip, and S. Rangan, "Low power analog-to-digital conversion in millimeter wave systems: Impact of resolution and bandwidth on performance," in *Information Theory and Applications Workshop (ITA)*, (San Diego, CA, USA), pp. 191–198, Feb. 2015.

[115] J. Zhang, L. Dai, Z. He, B. Ai, and O. A. Dobre, "Mixed-ADC/DAC multipair massive MIMO relaying systems: Performance analysis and power optimization," *IEEE Transactions on Communications*, vol. 67, pp. 140–153, Jan. 2018.

[116] P. Billingsley, *Probability and measure.* John Wiley & Sons, 2008.

[117] Y. LeCun, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

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

[119] W. C. Lee, *Mobile communications engineering: theory and applications.* NY, USA: McGraw-Hill Education, 1998.

[120] M. M. Amiri, T. M. Duman, and D. Gunduz, "Collaborative machine learning at the wireless edge with blind transmitters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, (Ottawa, ON, Canada), Nov. 2019.

[121] W.-S. Hou and B.-S. Chen, "ICI cancellation for OFDM communication systems in time-varying multipath fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 2100–2110, Sept. 2005.

[122] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[123] Z. E. Ankarali, B. Peköz, and H. Arslan, "Flexible radio access beyond 5G: A future projection on waveform, numerology, and frame design principles," *IEEE Access*, vol. 5, pp. 18295–18309, Mar. 2017.

[124] W. C. Jakes and D. C. Cox, *Microwave mobile communications*. Wiley-IEEE press, 1994.

[125] C. Fassino, G. Pistone, and M. P. Rogantin, "Computing the moments of the complex Gaussian: Full and sparse covariance matrix," *Mathematics*, vol. 7, p. 263, Mar. 2019.

[126] D. Laptev and J. M. Buhmann, "Transformation-invariant convolutional jungles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3043–3051, June 2015.

[127] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "TI-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 289–297, June 2016.

[128] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE international conference on computer vision*, (Santiago, Chile), pp. 945–953, Dec. 2015.

[129] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Communications*, vol. 28, pp. 57–65, Aug. 2021.

[130] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Over-the-air computation for IoT networks: Computing multiple functions with antenna arrays," *IEEE Internet of Things Journal*, vol. 5, pp. 5296–5306, Dec. 2018.

[131] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *J. of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[132] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[133] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*, pp. 1–16, PMLR, 2017.

[134] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, pp. 379–423, July 1948.

[135] W. Weaver, "The mathematics of communication," *Scientific American*, vol. 181, pp. 11–15, July 1949.

[136] R. Carnap, Y. Bar-Hillel, *et al.*, "An outline of a theory of semantic information," 1952.

[137] L. Floridi, "Outline of a theory of strongly semantic information," *Minds and machines*, vol. 14, pp. 197–221, May 2004.

[138] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *2011 IEEE Network Science Workshop*, (West Point, NY, USA), pp. 110–117, June 2011.

[139] M. Kalfa, M. Gok, A. Atalik, B. Tegin, T. M. Duman, and O. Arikan, "Towards goal-oriented semantic signal processing: Applications and future challenges," *Digital Signal Processing*, vol. 119, p. 103134, Dec. 2021.