Advances in robust signal processing and applications

Xinjue Wang

Aalto University

Aalto University publication series
Doctoral Theses 249/2025

Advances in robust signal processing and applications

Xinjue Wang

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall T1 of the school on 19 December 2025 at 12:00.

Aalto University
School of Electrical Engineering
Department of Information and Communications Engineering

Supervising professor
Prof. Esa Ollila, Aalto University, Finland

Thesis advisor
Prof. Sergiy A. Vorobyov, Aalto University, Finland

Preliminary examiners
Prof. Paolo Di Lorenzo, Sapienza University of Rome, Italy
Prof. Marc Castella, Telecom SudParis, France

Opponents
Prof. Paolo Di Lorenzo, Sapienza University of Rome, Italy
Prof. Bhavani Shankar Mysore, University of Luxembourg, Luxembourg

Aalto University publication series
Doctoral Theses 249/2025

ISBN 978-952-64-2879-6 (soft cover)
ISBN 978-952-64-2878-9 (PDF)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (PDF)

https://urn.fi/URN:ISBN:978-952-64-2878-9
Unigrafia Oy
Helsinki 2025

Aalto University

ABSTRACT

Author Xinjue Wang

Name of the doctoral thesis Advances in Robust Signal Processing and Applications

Article-based thesis

Number of pages 151

Keywords Robust Signal Processing, Graph Convolutional Neural Networks, massive Machine-Type Communications, Activity Detection, Tensor Decomposition

Robust signal processing and machine learning methodologies are critical for the reliable operation of modern technological systems, particularly in dynamic and uncertain environments such as the Internet of Things (IoT).

However, system performance is often compromised by pervasive challenges, including structural perturbations in graph-based models, complex non-Gaussian noise in communication systems, and the structural heterogeneity of high-dimensional tensor data. This thesis addresses these critical challenges by developing a suite of robust methodologies grounded in distinct yet complementary perspectives on robustness.

First, this research establishes a comprehensive analytical framework to quantify the sensitivity of Graph Convolutional Neural Networks (GCNNs) to probabilistic graph perturbations. Tight, expected bounds for Graph Shift Operator (GSO) errors are derived without requiring eigendecomposition, and a linear relationship between GSO perturbations and GCNN output differences is revealed, providing theoretical stability guarantees for multilayer architectures.

Second, novel robust device activity detection (AD) algorithms are developed for massive random access systems operating under challenging non-Gaussian noise. By formulating AD objectives based on robust loss functions (e.g., Huber's loss) and proving the geodesic convexity of the conditional objective, efficient fixed-point, coordinate-wise, and matching pursuit algorithms with proven convergence are proposed. These methods significantly outperform traditional Gaussian-based approaches in heavy-tailed noise environments.

Third, a generalized Nonnegative Structured Kruskal Tensor Regression (NS-KTR) framework is introduced for the effective and interpretable modeling of high-dimensional tensor data. This framework integrates non-negativity constraints with mode-specific hybrid regularizations (e.g., LASSO, total variation, ridge), accommodates both linear and logistic regression, and is solved via an efficient ADMM-based algorithm.

Collectively, this thesis advances the theory and practice of robust signal processing by providing novel tools for ensuring stability, resilience to distributional deviations, and robust modeling through structural priors. The developed frameworks and algorithms contribute to the design of more reliable and efficient signal processing systems for real-world applications.

Preface

This research was carried out at the Department of Information and Communications Engineering, School of Electrical Engineering, Aalto University.

I have been exceptionally fortunate to have Prof. Esa Ollila and Prof. Sergiy A. Vorobyov as my supervisors. Their mentorship was pivotal in shaping my academic trajectory. I am deeply grateful for their unwavering guidance, encouragement, and trust, which was both comprehensive and meticulous. They patiently guided me through complex technical details while also teaching me how to approach research on a conceptual level and cultivate a refined research taste. Their unceasing support, along with the opportunities they provided for networking and professional development, has contributed significantly to my intellectual growth.

I sincerely thank my pre-examiners, Prof. Paolo Di Lorenzo and Prof. Marc Castella, for taking the time to carefully review my thesis and provide constructive feedback. I extend further gratitude to Prof. Paolo Di Lorenzo for also serving as my opponent, alongside Prof. Bhavani Shankar Mysore. Their insightful questions and rigorous evaluation were invaluable.

I am indebted to the many outstanding scholars I have been fortunate to learn from and work with. Special thanks go to my co-authors, Dr. Xiuheng Wang and Prof. Ammar Mian, for fruitful research collaborations. I would also like to thank my colleagues at Aalto University for their friendship, intellectual companionship, and moral support throughout this journey.

This dissertation would not have been possible without generous financial support. I wish to thank the School of Electrical Engineering at Aalto University for providing one of the best research environments in the world. Many thanks also go to the Nokia Foundation, the Walter Ahlström Foundation, the French Institute in Finland, the Finnish Society of Sciences and Letters, and the Research Council of Finland.

Last but not least, my heartfelt thanks go to my friends, both within the Aalto community and those I met at international conferences. Your companionship provided not only refreshing distractions from academic life but also a vital source of new energy and motivation to keep moving

forward.

Most importantly, I would like to thank my mother for her unconditional love and for always believing in me.

Espoo, November 17, 2025,

Xinjue Wang

Contents

Preface **1**

Contents **3**

List of Publications **5**

Author's Contribution **7**

Abbreviations **9**

Symbols **11**

1. Introduction **1**
1.1 Background and Motivation 1
1.1.1 Ensuring Stability and Interpretability of Graph-based Learning Models 1
1.1.2 Achieving Robust Device Activity Detection in Complex IoT Environments 2
1.1.3 Modeling of High-Dimensional Structured Tensor Data 2
1.1.4 Objectives 3
1.1.5 Perspectives on robustness 4
1.1.6 Summary of publications and their contributions . 6
1.1.7 Thesis structure 8

2. Graph Convolutional Neural Networks Sensitivity Under Probabilistic Error Model **11**
2.1 Preliminaries on GCNN 13
2.2 Probabilistic graph error model 14
2.3 Expected bound for GSO error 16
2.4 Expected bound for GF and GCNN 19
2.4.1 GF sensitivity analysis 19
2.4.2 GCNN sensitivity analysis 20
2.5 Specifications for GCNN variants 22

2.5.1 Specification for GIN 22
2.5.2 Specification for SGCN 23
2.6 Conclusion and discussion 23

3. Robust Activity Detection for Massive Random Access 25
3.1 Problem formulation 26
3.2 Theory of proposed robust method 29
3.2.1 Solving the Conditional Objective 32
3.2.2 FP Algorithm 33
3.3 Proposed methods 35
3.3.1 RCWO 35
3.3.2 RCL-MP 35
3.3.3 Computational complexity 37
3.4 Numerical experiment 38
3.5 Conclusion and discussion 40

4. Generalized Nonnegative Structured Kruskal Tensor Regression for Effective and Interpretable Data Modeling 43
4.1 Background of tensor regression and motivation 44
4.2 Preliminaries on tensor 45
4.3 Proposed method 47
4.3.1 Structured regularization 47
4.3.2 Alternating optimization framework 49
4.4 ADMM algorithm for subproblem 50
4.5 Conclusion and discussion 52

5. Conclusion 55
5.1 Future research directions 56

References 59

Publications 65

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I X. Wang, E. Ollila, and S. A. Vorobyov. Graph neural network sensitivity under probabilistic error model. In *Proc. 30th European Signal Processing Conference (EUSIPCO)*, Belgrade, Serbia, pp. 2146– 2150, August 2022.

II X. Wang, E. Ollila, and S. A. Vorobyov. Graph convolutional neural networks sensitivity under probabilistic error model. *IEEE Transactions on Signal and Information Processing over Networks*, vol. 10, pp. 788-803, October 2024.

III X. Wang, E. Ollila, S. A. Vorobyov. Robust activity detection for massive access using covariance-based matching pursuit. In *Proc. 50th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, pp. 1-5, April 2025.

IV X. Wang, E. Ollila, and S. A. Vorobyov. Robust activity detection for massive random access. *IEEE Transactions on Signal Processing*, vol. 73, pp. 3513-3527, Aug 2025.

V X. Wang, E. Ollila, and S. A. Vorobyov. Nonnegative structured Kruskal tensor regression. In *Proc. 9th Workshop Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Herradura, Costa Rica, pp. 441-445, December 2023.

VI X. Wang, E. Ollila, S. A. Vorobyov, and A. Mian. Generalized nonnegative structured Kruskal tensor regression. *Signal Processing*, 110338, Mar. 2026.

Author's Contribution

Publication I: "Graph neural network sensitivity under probabilistic error model"

The author developed the core analysis, performed the numerical experiments, interpreted the results, and wrote the article, integrating comments from the co-authors.

Publication II: "Graph convolutional neural networks sensitivity under probabilistic error model"

The author developed the core analysis, performed the numerical experiments, interpreted the results, and wrote the article, integrating comments from the co-authors.

Publication III: "Robust activity detection for massive access using covariance-based matching pursuit"

The initial idea was proposed by Prof. E. Ollila. The author then formulated the problem, derived the algorithm, performed the numerical simulations and was the main writer of the article while also incorporating comments from other co-authors.

Publication IV: "Robust activity detection for massive random access"

The initial idea was proposed by Prof. E. Ollila. The author then formulated the problem, derived the proofs and algorithm, performed the numerical

simulations and was the main writer of the article while also incorporating comments from other co-authors.

Publication V: "Nonnegative structured Kruskal tensor regression"

The initial idea was proposed by Prof. E. Ollila, and then further developed by the author in terms of the detailed algorithm implementation. The author performed the numerical simulations and was the main writer of the article while also incorporating comments from other co-authors.

Publication VI: "Generalized nonnegative structured Kruskal tensor regression"

The author developed the core algorithm design, performed the numerical experiments, interpreted the results, and wrote the article, integrating comments from the co-authors.

Abbreviations

AD Activity Detection

ADMM Alternating Direction Method of Multipliers

AMP Approximate Message Passing

BS Base Station

CL Covariance Learning

CL-MP Covariance Learning-based Matching Pursuit

CPD Canonical Polyadic Decomposition

CS Compressive Sensing

CWO Coordinate-wise Optimization

EM Expectation-Maximization

FC-HADM Fully Combined Hadamard Matrix

FP Fixed-Point

GAT Graph Attention Network

GCNN Graph Convolutional Neural Network

GF Graph Filter

GIN Graph Isomorphism Network

GNN Graph Neural Network

GSO Graph Shift Operator

HSI Hyperspectral Imaging

IHT Iterative Hard Thresholding

i.i.d. Independent and Identically Distributed

IoT Internet of Things

KTR Kruskal Tensor Regression

LLF Log-Likelihood Function

LSFC Large-Scale Fading Component

MIMO Multiple-Input Multiple-Output

MLE Maximum Likelihood Estimation

MLP Multilayer Perceptron

mMTC massive Machine-Type Communications

MMV Multiple Measurement Vector

MP Matching Pursuit

MPNN Message Passing Neural Network

MTD Machine-Type Device

NS-KTR Nonnegative Structured Kruskal Tensor Regression

OMP Orthogonal Matching Pursuit

PDH Positive Definite Hermitian

PMF Probability Mass Function

RC-HADM Restrictively Combined Hadamard Matrix

RCL-MP Robust Covariance Learning-based Matching Pursuit

RCWO Robust Coordinate-wise Optimization

RIP Restricted Isometry Property

SBL Sparse Bayesian Learning

SGCN Simple Graph Convolution Network

SMV Single Measurement Vector

SNR Signal-to-Noise Ratio

SOMP Simultaneous Orthogonal Matching Pursuit

TR Tensor Regression

TT Tensor Train

TV Total Variation

WL Weisfeiler-Lehman

Symbols

Scalars

c Tuning constant (e.g., in Huber's loss)

d_u Degree of node u

K Number of active users; Taps in a graph filter

L Length of pilot sequences; Number of GCNN layers

M Number of antennas at the base station

N Total number of potential users/devices

q Quantile parameter for Huber's loss

R Rank of a tensor decomposition

α_i Binary activity indicator for user i

β_i Large-scale fading component (LSFC)

γ_i Combined large-scale channel power for user i

δ_u^-, δ_u^+ Number of deleted and added edges for node u

ϵ_1, ϵ_2 Probabilities of edge deletion and addition

λ Regularization parameter

$\rho(t)$ Robust loss function

ρ_i Uplink transmission power for user i

σ^2 Noise variance

τ_u Minimum degree among neighbors of node u

Vectors

$\boldsymbol{a}_i$ Pilot sequence (signature) for user i

$\boldsymbol{h}_i$ Small-scale fading channel vector for user i

$\boldsymbol{r}$ Residual vector

$\boldsymbol{x}$ Sparse signal vector

$\boldsymbol{x}_i$ Channel vector for user i (row of matrix $\boldsymbol{X}$)

$\boldsymbol{y}$ Received signal vector

$\boldsymbol{\gamma}$ Vector of all channel powers $(\gamma_1, \ldots, \gamma_N)^\top$

$\mathbf{1}$ Vector of all ones

Matrices

$\boldsymbol{A}$ Pilot matrix; Sensing matrix

$\boldsymbol{B}_d$ Factor matrix for mode d in a Canonical polyadic decomposition (CPD)

$\boldsymbol{D}$ Diagonal degree matrix

$\boldsymbol{E}$ GSO error matrix; Noise matrix

$\boldsymbol{H}$ Filter coefficient matrices in a GCNN

$\boldsymbol{I}$ Identity matrix

$\boldsymbol{L}$ Graph Laplacian matrix

$\boldsymbol{S}$ Graph Shift Operator (GSO)

$\hat{\boldsymbol{S}}$ Perturbed GSO

$\boldsymbol{X}$ Channel matrix; Graph signal feature matrix

$\boldsymbol{Y}$ Received signal matrix

$\boldsymbol{\Gamma}$ Diagonal matrix of channel powers, $\text{diag}(\boldsymbol{\gamma})$

$\boldsymbol{\Sigma}$ Covariance matrix of the received signal

$\boldsymbol{\Phi}(\cdot)$ GCNN model function

$\boldsymbol{\Delta}_\epsilon$ Random matrix for probabilistic error model

Tensors and Sets (Calligraphic)

$\mathscr{B}$ D-way tensor (e.g., parameter tensor)

$\mathscr{G}$ Graph; Core tensor in Tucker decomposition

$\mathscr{X}$ D-way tensor (e.g., covariate tensor)

$\mathscr{E}$ Set of edges in a graph

$\mathcal{M}$ Support set of active users

$\mathcal{N}_u$ Set of neighbors of node u

$\mathcal{V}$ Set of nodes in a graph

Operators and Other Notations

argmin Argument of the minimum

argmax Argument of the maximum

cov$(\cdot)$ Covariance operator

diag$(\cdot)$ Diagonalization operator

$\mathbb{E}(\cdot)$ Expectation operator

Pr$(\cdot)$ Probability operator

supp$(\cdot)$ Support of a vector or matrix (set of non-zero indices)

tr$(\cdot)$ Trace of a matrix

$\circ$ Outer product of vectors; Hadamard (element-wise) product of matrices

$\langle \cdot, \cdot \rangle$ Inner product

$\|\cdot\|$ **or** $\|\cdot\|_2$ ℓ_2-norm (vector) or spectral norm (matrix)

$\|\cdot\|_1$ ℓ_1-norm

$(\cdot)^\top$ Transpose

$(\cdot)^\mathsf{H}$ Hermitian transpose

1. Introduction

1.1 Background and Motivation

Robust signal processing and machine learning methodologies are fundamental to diverse technological systems operating in increasingly dynamic environments [84, 23]. Applications span from graph-based analysis of complex network data [4] and signal detection in massive internet of things (IoT) communication systems [43], to the modeling of high-dimensional multi-aspect data in areas like imaging analysis [81]. These systems consistently encounter unpredictable noise, interference, and environmental or structural perturbations, which can compromise performance. The ability to reliably process imperfect signals and extract meaningful features from corrupted data is therefore critical. Consequently, the development of robust methodologies ensuring performance reliability under such conditions has become a paramount research focus.

This doctoral research specifically addresses three critical areas where data irregularities and model uncertainties pose significant challenges to achieving reliable and efficient signal processing.

1.1.1 Ensuring Stability and Interpretability of Graph-based Learning Models

Graph-based learning models, particularly Graph Convolutional Neural Networks (GCNNs), have emerged as powerful tools for analyzing data with network structures [12, 73]. Such data is ubiquitous, arising in domains from social network analysis, recommendation systems, bioinformatics and graph signal processing applications [60]. A key characteristic across these applications is the interdependency between entities. GCNNs can model this complex relationship effectively. However, a fundamental challenge lies in their sensitivity/stability to perturbations in the underlying graph structure [52, 18, 16]. In real-world scenarios, the graph defining

data relationships is usually not perfectly known or is subject to dynamic changes. These changes include link additions/deletions or noise in node features. Such structural uncertainties can significantly impact the graph shift operator (GSO), through which GCNNs propagate information. Without understanding how these perturbations affect GCNN outputs, their stability and reliability in safety-critical applications remain a crucial concern. This necessitates developing an analytical framework to quantify GCNN sensitivity and establish theoretical guarantees for their stability. Such frameworks can also guide the design of resilient graph learning architectures.

1.1.2 Achieving Robust Device Activity Detection in Complex IoT Environments

In Internet of Things (IoT) framework, the massive machine-type communications (mMTC) paradigm envisions massive deployment of interconnected devices, often characterized by sporadic data transmission patterns [42]. Efficiently identifying which devices are active at any given time, a task known as device Activity Detection (AD), is fundamental for effective resource allocation, interference management, and overall network performance [42, 8]. The Gaussian noise assumption is a cornerstone of modern wireless communication systems, yet certain operational environments may exhibit deviations from this model. The presence of outliers or impulsive noise can lead to heavy-tailed statistical distributions [83, 84, 22]. Such non-Gaussian characteristics can be attributed to a variety of sources, including electromagnetic interference in industrial settings [17, 44], impulsive noise in mobile channels [51], and heavy-tailed interference over wireless links [10]. These scenarios represent cases where methodologies offering additional robustness can provide benefits. For mMTC systems deployed in diverse settings, methods that maintain consistent performance are desirable. AD frameworks based on Gaussian noise assumption may experience performance degradation when encountering heavy-tailed noise or outliers. This could potentially affect the reliability of IoT networks where devices operate under such varying conditions. Consequently, the design of accurate and efficient AD frameworks that are robust to these noise characteristics remains a main research focus.

1.1.3 Modeling of High-Dimensional Structured Tensor Data

In contemporary machine learning, encompassing both regression and classification tasks, datasets are frequently encountered in the form of high-dimensional tensors or multi-way arrays [62, 9]. A common initial step is to vectorize such data for use with traditional algorithms. However, this vectorization can disrupt crucial inherent structures and interdepen-

dencies. Utilizing these tensors directly as covariates is therefore a more natural and effective approach, preserving the rich, multi-aspect information they encapsulate [62, 9]. Furthermore, while high-dimensional, much of this tensor-structured data exhibits an underlying low-rank property. Exploiting this low-rank assumption through tensor methods can significantly reduce the number of model parameters compared to vectorized approaches. These include hyperspectral imaging (HSI), biomedical signal analysis, and other domains where data exhibits multi-dimensional structure and complex interdependencies [77, 27, 81].

While tensor methods provide powerful tools for analyzing such data, several challenges impede their optimal application. Real-world tensor data frequently exhibits inherent structural properties such as sparsity in certain modes, smoothness across others, piecewise constancy, or physical constraints like non-negativity. For instance, hyperspectral images typically display smooth variations across spectral bands while maintaining locally constant profiles in spatial dimensions. Conventional tensor regression or decomposition models may fail to adequately capture this structural heterogeneity or enforce critical domain-specific constraints, leading to suboptimal model performance or loss of interpretability. Addressing these challenges requires developing tensor regression models that can incorporate flexible mode-specific structural regularizations, enforce necessary constraints like non-negativity, and ensure computational tractability.

1.1.4 Objectives

To address these critical challenges of data irregularities and model uncertainties, this thesis pursues the following objectives:

- Develop an analytical framework to quantify the sensitivity of GCNNs to structural perturbations. This objective involves establishing tight theoretical upper bounds on the impact of GSO errors, analyzing their propagation through GCNN layers to provide guarantees for network stability under perturbation.

- Design and validate robust device AD algorithms specifically tailored for mMTC systems possibly operating under complex non-Gaussian noise. This involves: *(i)* Formulating AD objective functions based on robust loss functions to mitigate the impact of outliers, *(ii)* Developing efficient algorithms with proven convergence guarantees, and *(iii)* Guaranteeing the performance of robust algorithms comparable to their Gaussian-based counterparts when Gaussianity is valid.

- Construct structured and interpretable tensor regression models for high-dimensional multi-aspect data. This objective focuses on designing

models that integrate non-negativity and mode-specific hybrid regularizations for both linear and logistic regression, and developing efficient optimization techniques for computational scalability.

1.1.5 Perspectives on robustness

The concept of robustness, central to this doctoral research, is multifaceted and manifests differently depending on the specific problem domain, the nature of the encountered uncertainties, and the desired system properties. Distinct yet related perspectives on robustness adopted within the three core research pillars of this thesis are elaborated.

1. Robustness as Output Stability under Structural Perturbations
This perspective is particularly relevant for models like Graph Convolutional Neural Networks (GCNNs) that rely on an underlying data structure, such as a graph topology. Here, robustness primarily refers to stability: the system's ability to maintain a bounded output when its input structure (e.g., the Graph Shift Operator, GSO) experiences bounded perturbations. Such stability ensures predictable model behavior despite uncertainties in the graph. Sensitivity analysis then serves as a quantitative tool. It measures the dependency of output variations on input perturbations and their parameters. A system with low sensitivity is generally considered robust. Figure 1.1 conceptually illustrates this scenario. This thesis (Chapter 2) investigates this form of robustness by establishing GSO error bounds and analyzing error propagation in GCNNs, providing theoretical underpinnings for reliable graph-based learning.

2. Robustness as Resilience to Data Distributional Deviations
In signal processing tasks as device AD in the IoT, data may not strictly adhere to idealized statistical assumptions (e.g., Gaussian noise). Robustness, in this context, signifies the algorithm's insensitivity to such distributional deviations. Real-world noise and interference often exhibit non-Gaussian characteristics like heavy tails or outliers, which can severely degrade traditional methods. Robust statistical techniques address this by employing robust loss functions (e.g., Huber's, t-loss). These functions mitigate the undue influence of atypical data points by assigning them lower weights or bounded influence. Figure 1.2 conceptually illustrates this scenario. This research (Chapter 3) advances this view of robustness by designing AD algorithms based on robust loss functions, thereby enhancing detection performance in challenging non-Gaussian environments.

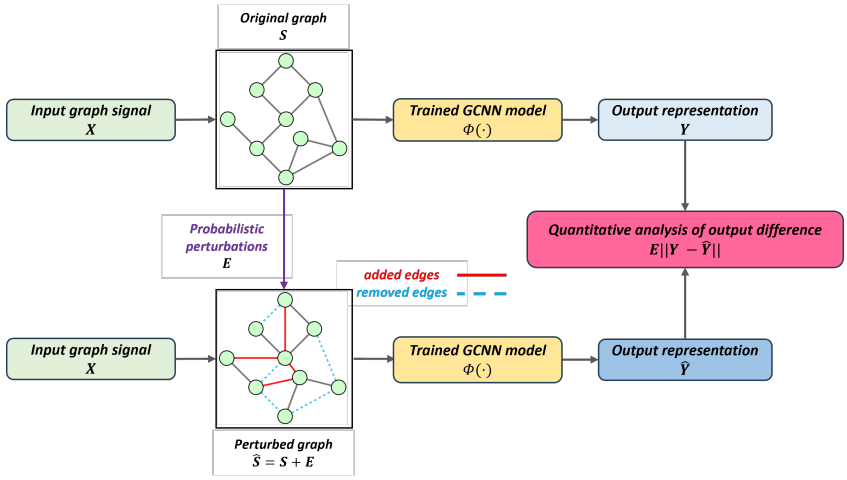


Figure 1.1. Conceptual framework for analyzing GCNN output sensitivity to probabilistic graph perturbations. The same input graph signal $\boldsymbol{X}$ is processed by an identical trained GCNN model $\boldsymbol{\Phi}(\cdot)$ under two scenarios: first, using the original graph structure $\boldsymbol{S}$ to produce output $\boldsymbol{Y}$, and second, using a perturbed graph structure $\hat{\boldsymbol{S}} = \boldsymbol{S} + \boldsymbol{E}$ resulting from probabilistic edge additions (red solid lines) or removals (blue dashed lines) due to perturbations $\boldsymbol{E}$. The core analysis focuses on the quantitative assessment of the expected difference between these outputs, $\mathbb{E}\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|$, to understand and bound the impact of structural GSO perturbations on GCNN performance, thereby evaluating network stability.

3. Robustness through Structural Priors and Regularization.
When dealing with high-dimensional data (e.g., tensors) or ill-posed problems, robustness can be achieved by incorporating structural prior knowledge about the desired solution. This is typically implemented via regularization. Regularization techniques add penalty terms to the objective function, guiding the solution towards possessing specific properties like sparsity, smoothness, piecewise constancy, or non-negativity. This constrains the solution space, making the model less sensitive to noise or minor variations in the training data, thus improving generalization and stability. From a Bayesian perspective, regularization terms correspond to prior distributions over the model parameters. A well-chosen prior, reflecting true data structures or problem constraints, leads to more stable and physically meaningful solutions, even with limited or corrupted data. Figure 1.3 conceptually illustrates this scenario. This thesis (Chapter 4) explores this aspect by developing tensor regression models with mode-specific hybrid regularizations and non-negativity constraints, implicitly enhancing model robustness while capturing complex data structures.

These distinct perspectives on robustness are not isolated challenges but are, in fact, interconnected facets of the overarching goal of this dissertation: to develop reliable and interpretable signal processing systems for the modern era. The stability of a graph-based model under structural perturbations, the resilience of a detection algorithm to distributional devi-

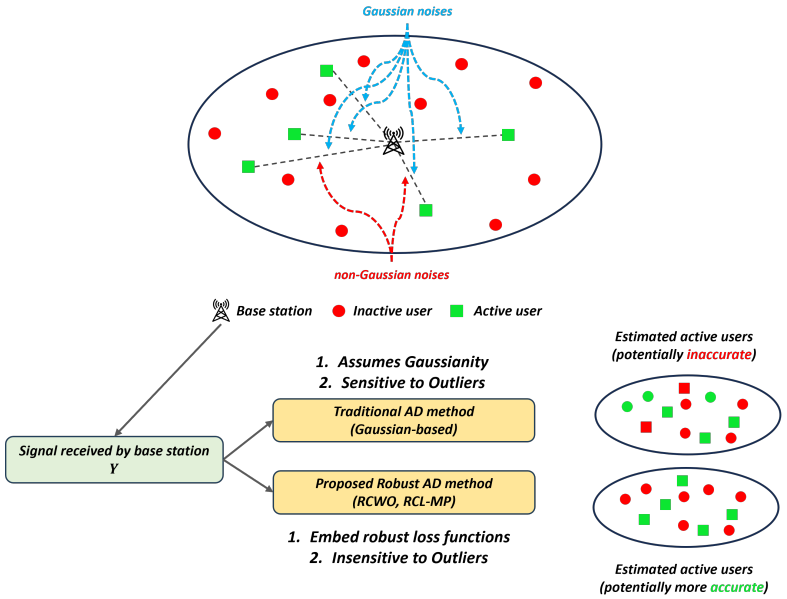


Figure 1.2. Conceptual illustration of robust device AD in an IoT scenario with mixed noise conditions. The top panel depicts a base station receiving signals from multiple users (active users: green squares; inactive users: red circles) over communication links (black dashed lines) affected by ubiquitous Gaussian noises (blue dashed arrows) and sporadic, potentially strong non-Gaussian noises (red dashed arrows). The bottom panel illustrates the AD process: the signal received by the base station $\boldsymbol{Y}$ is processed by either a traditional Gaussian-based AD method, which assumes Gaussianity and is sensitive to outliers, or the proposed robust AD method (e.g., RCWO, RCL-MP), which embeds robust loss functions and is insensitive to outliers. Consequently, the robust method yields more accurate estimation of active users, especially under non-Gaussian noise conditions, demonstrating its resilience to data distributional deviations.

ations in data, and the ability to instill robustness in a high-dimensional model through structural priors are complementary pillars. A truly robust system must be resilient to uncertainty in its underlying structure, its input data, and its own parameterization. This thesis systematically investigates these pillars to provide a holistic contribution to the theory and practice of robust engineering.

1.1.6 Summary of publications and their contributions

- In Publication I, an analytical framework is initiated to assess GCNN sensitivity to a probabilistic graph error model. Upper bounds for GSO (adjacency matrix and its augmented normalized version) errors based on spectral analysis are derived, and the sensitivity of a simple GCNN is studied under this model.

- In Publication II, a comprehensive sensitivity analysis framework for

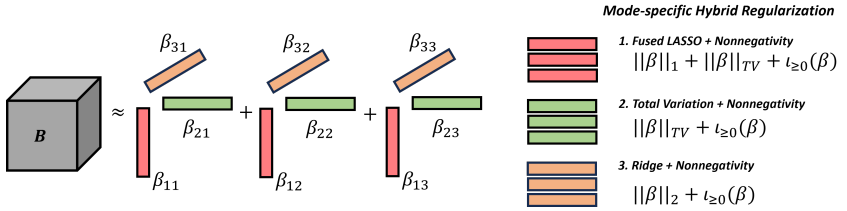


Figure 1.3. Conceptual illustration of the NS-KTR parameter tensor $\boldsymbol{B}$ modeling. The parameter tensor is represented via a Canonical polyadic decomposition (CPD) as a sum of R rank-1 tensors ($R = 3$ shown for illustration, top right), each formed by the outer product of factor vectors corresponding to different modes (e.g., red, green, and orange vectors for a 3-mode tensor). The NS-KTR framework then applies mode-specific hybrid regularization to the collection of factors for each mode (bottom right). For example, Fused LASSO might be applied to the factors of the first mode (red), Total Variation to the second mode (green), and a Ridge regularizer to the third mode (orange). Crucially, a nonnegativity regularizer (constraint) is applied to all factor vectors, ensuring physical interpretability and contributing to model robustness by incorporating structural priors.

GCNNs under probabilistic graph perturbations was established. This work introduced tight expected GSO error bounds explicitly linked to a generic stochastic graph error model, which parametrizes the deletion and addition of edges of a graph. Furthermore, a linear relationship between GSO perturbations and GCNN layer output differences was revealed, demonstrating stability via a recursion of linearity for multilayer GCNNs such as GIN and SGCN. The proposed model is free of limited perturbation budget, and is generic to GCNN models, with specific adjustments for GSO, graph shifts count, network layer count, and activation functions.

- In Publication III, a robust activity detection algorithm, termed Robust Covariance Learning-based Matching Pursuit (RCL-MP), was proposed for grant-free massive access systems. The method proposes uses a general robust loss function instead of the non-robust Gaussian loss in the Gaussian negative log-likelihood objective and utilizes a matching pursuit framework for greedily selecting active users, demonstrating superior robustness in non-Gaussian noise conditions compared to non-robust counterparts.

- In Publication IV, robust statistical techniques for device AD in massive random access scenarios were systematically developed, extending beyond Gaussianity of the measurement vectors. A key theoretical contribution is proving the geodesic convexity of the coordinate-wise robust AD objective function, which leads to an efficient fixed-point (FP) algorithm. The corresponding convergence guarantee is also established. Based on this FP algorithm, two robust AD algorithms are proposed: a coordinate-wise optimization algorithm (RCWO) and an enhanced RCL-

MP. Two different experiments are performed: a conventional simulation study describing the performance of the algorithms under different non-Gaussian measurement conditions and a more realistic uplink simulation for mMTC device AD. Two algorithms achieve high accuracy and close computational efficiency to their Gaussian-based counterparts.

- In Publication V, a nonnegative structured Kruskal tensor regression (KTR) model was investigated, incorporating sparsity and smoothness-inducing regularizations alongside non-negativity constraints on the coefficient tensor. This work proposed a penalized nonnegative KTR problem, motivated by the inherent positive-valued nature of many tensor covariates and responses, and the structural properties (e.g., spectral smoothness, spatial piecewise constancy) observed in applications like hyperspectral imaging. An efficient block-wise alternating minimization method was employed for its solution, with simulations demonstrating the approach's efficacy.

- In Publication VI, a generalized Nonnegative Structured Kruskal Tensor Regression (NS-KTR) framework was introduced. This novel framework integrates mode-specific hybrid regularization (fused LASSO, total variation, and ridge regularizations) with non-negativity constraints on the coefficient tensor. The NS-KTR allows both linear and logistic regression formulations for multidimensional tensor data. An alternating optimization algorithm based on the Alternating Direction Method of Multipliers (ADMM) is developed for efficient parameter estimation. The performance and practicality of NS-KTR framework are demonstrated through synthetic experiments. The framework is also validated on a real-world hyperspectral imaging (HSI) datasets of wheat flag leaves. NS-KTR consistently outperforms over conventional tensor regression methods in terms of estimation accuracy in both regression and classification tasks (e.g., predicting agricultural traits and classifying wheat cultivars).

1.1.7 Thesis structure

The remainder of this thesis is organized as follows. Chapter 2 presents an analytical framework for quantifying the sensitivity of GCNNs to probabilistic structural perturbations in the GSO. Theoretical bounds on GSO errors and their propagation through GCNN layers are established, providing stability guarantees. Chapter 3 details the design and validation of robust device AD algorithms tailored for mMTC systems operating under complex non-Gaussian noise. Novel AD objective functions based on robust loss functions and efficient, scalable algorithms with proven convergence are introduced. Chapter 4 introduces a generalized NS-KTR framework

for the effective and interpretable modeling of high-dimensional multi-aspect data. This chapter details models integrating non-negativity and mode-specific hybrid regularizations, alongside efficient ADMM-based optimization techniques. Chapter 5 provides a synthesis of the research contributions, discusses their broader implications, and outlines potential directions for future work.

2. Graph Convolutional Neural Networks Sensitivity Under Probabilistic Error Model

As established in Chapter 1, a significant challenge in deploying GCNNs is their sensitivity to graph perturbations. GCNNs have demonstrated significant efficacy in learning from data characterized by network structures, as highlighted in Chapter 1. However, a critical impediment to their widespread and reliable deployment, particularly in dynamic or uncertain environments, is their inherent instability to perturbations in the underlying graph structure [18, 16, 15]. In practical scenarios, the graph topology, represented by the GSO, is often an estimate or subject to alterations such as edge additions, deletions, or node feature noise [52, 74]. These structural uncertainties can propagate through the GCNN layers, potentially leading to unpredictable or degraded performance.

This chapter directly addresses this challenge by developing a rigorous analytical framework to quantify the sensitivity of GCNNs to such probabilistic structural perturbations.[1] The primary objective is to establish theoretical underpinnings for GCNN stability, providing guarantees on how parameterized GSO errors impact the network's output. Specifically, this involves deriving tight expected bounds for GSO errors under a generic stochastic graph error model and subsequently analyzing the propagation of these errors through the layers of various GCNN architectures.

The contributions presented in this chapter are primarily based on the work in Publication II, which extends and deepens the preliminary investigations initiated in Publication I. The main contributions are summarized as follows.

- **General error model.** We use a general probabilistic graph error model [52] in the analysis framework. This error model is practically appealing because it is grounded in stochastic block models, supports both deletion and addition of edges, and permits a broader perturbation scale.

[1] Note that the detailed derivations used to establish the Lemmas and Theorems presented in the rest of the Thesis are provided as part of individual contributions.

- **Tight GSO error bound.** We first provide an ℓ_2 norm-based deterministic GSO error bound in Publication I, and further give a tighter expected GSO error bound. This expected bound is interpretable, as it directly monitors node degrees that are varying. The varying node degrees are explicitly connected to error model parameters, i.e., probabilities of deleting and adding edges. Additionally, our bound avoids eigendecomposition of GSO, which is computationally expensive for large graphs.

- **Sensitivity analysis framework.** *(i)* We remove the constraint on perturbation scale and allow for a large perturbation budget. Our analysis demonstrates empirically to be valid even under such perturbation. *(ii)* We provide probabilistic expected bounds, treating deterministic perturbations as particular cases of our analysis. *(iii)* This framework applies to general GCNN models, with appropriate adjustments for GSO, graph shifts count, network layer count, and activation functions.

These advancements collectively provide a robust analytical toolkit for assessing GCNN performance under graph perturbations. The developed general error model, tight GSO error bounds, and generic sensitivity analysis framework directly contribute to understanding and guaranteeing GCNN stability, a key aspect of robustness highlighted as a challenge in Chapter 1. This work therefore offers insights for the reliable deployment of GCNNs in uncertain real-world settings.

This chapter's analytical framework advances GCNN stability analysis beyond prior work in several key ways. Unlike previous studies constrained to edge deletions [18], specific rewiring schemes [31], or small norm-bounded errors [16], we employ a general probabilistic error model supporting both edge deletion and addition. We derive tight expected GSO error bounds that improve upon preliminary deterministic bounds [69], directly linking GSO error to perturbation probabilities through node degree changes for enhanced interpretability. Our bounds avoid computationally intensive GSO eigendecomposition required by spectral approaches [16, 18], improving scalability for large graphs."

This chapter's work has gained recognition and citations within the GNN community. Studies have acknowledged our contributions to GNN stability analysis against perturbations, particularly alongside eigenvalue/eigenvector perturbation analyses [19, 79]. Our demonstration of GCNN sensitivity to sparse graph perturbations has been cited in applied domains, including unstructured mesh-based GNNs for environmental modeling [32]. The probabilistic framework has been referenced for its analysis of embedding perturbations from stochastic edge addition/deletion [78]. This recognition highlights our framework's impact across theoretical and practical GNN research.

2.1 Preliminaries on GCNN

Graph Basics. An undirected, unweighted graph is denoted as $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$, where $\mathscr{V} = \{1, \ldots, N\}$ is the set of N nodes, $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ is the edge set, and $\mathscr{W} : \mathscr{V} \times \mathscr{V} \to \{0, 1\}$ is the edge weighting function assigning binary edges. We assume undirected and unweighted graphs, and thus $\mathscr{W}(i, j) = \mathscr{W}(j, i) = 1$. The 1-hop neighboring set of a node i is defined as $\mathscr{N}_i = \{j \in \mathscr{V} : (i, j) \in \mathscr{E}\}$. The degree of node i is d_i, and $\tau_i = \min_{j \in \mathscr{N}_i} d_j$ represents the minimum degree among its neighbors.

GSO. The GSO $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ represents the graph structure and dictates signal propagation between adjacent nodes. Common GSOs include the adjacency matrix $\boldsymbol{A}$, the Laplacian $\boldsymbol{L}$, or their normalized forms. These operators are crucial for analyzing data on graphs by capturing connectivity patterns. The adjacency matrix $\boldsymbol{A}$ defines connections: $[\boldsymbol{A}]_{ij} = 1$ if an edge exists between nodes i and j, and 0 otherwise. The Laplacian matrix $\boldsymbol{L}$ is defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A}\boldsymbol{1}_N)$ is a diagonal matrix, $[\boldsymbol{D}]_{ii} = d_i$, and $d_i = \sum_{j \in \mathscr{N}_i} [\boldsymbol{A}]_{ij}$ is the degree of node i. Normalized versions of the adjacency and Laplacian matrices are defined as $\boldsymbol{A}_{\mathrm{n}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ and $\boldsymbol{L}_{\mathrm{n}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$, and often used to manage data scaling and ensure consistency.

Graph Convolutional Filter. Graph signals, denoted by $\boldsymbol{x} \in \mathbb{R}^N$ (where $[\boldsymbol{x}]_i = x_i$ is the signal value at node v_i), are processed via the GSO. A K-tap graph convolutional filter $\boldsymbol{h}(\boldsymbol{S})$ is a polynomial of GSO with filter weights $\boldsymbol{h} = \{h_k\}_{k=0}^K$. The filter's output is obtained by the graph convolution

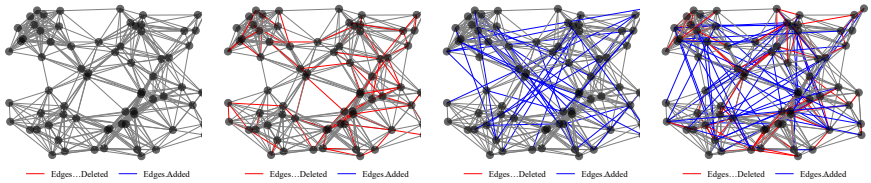
$$\boldsymbol{y} = h_0 \boldsymbol{S}^0 \boldsymbol{x} + \cdots + h_K \boldsymbol{S}^K \boldsymbol{x} = \sum_{k=0}^{K} h_k \boldsymbol{S}^k \boldsymbol{x} = \boldsymbol{h}(\boldsymbol{S})\boldsymbol{x}. \tag{2.1}$$

Here, $\boldsymbol{h}(\boldsymbol{S}) = \sum_{k=0}^K h_k \boldsymbol{S}^k$ is a shift-invariant graph filter, weighting information from up to K-hop neighborhoods. This filtered signal is typically then passed through a nonlinear activation function, forming a core GCNN component.

Graph Perceptron and GCNN. A Graph Perceptron [29] is a simple unit of transformation in the GCNN. For a multi-feature graph signals, represented as $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d] \in \mathbb{R}^{N \times d}$, where d is the number of features. An L-layer GCNN cascades multiple graph perceptrons. The output of layer $\ell - 1$ serves as the input to layer ℓ. Denoting the initial input as $\boldsymbol{X}_0 = \boldsymbol{X}$, the operation at layer ℓ is

$$\boldsymbol{Y}_\ell = \sum_{k=1}^{K} \boldsymbol{S}^k \boldsymbol{X}_{\ell-1} \boldsymbol{H}_{\ell k}, \quad \boldsymbol{X}_\ell = \sigma_\ell(\boldsymbol{Y}_\ell). \tag{2.2}$$

In this formulation, $\boldsymbol{Y}_\ell$ is the intermediate graph filter output at layer ℓ, $\sigma_\ell(\cdot)$ is the layer's nonlinear activation function, and $\boldsymbol{X}_\ell \in \mathbb{R}^{N \times F_\ell}$ is the graph signal with F_ℓ features. The set of filter coefficient matrices is



(a) $\epsilon_1 = 0, \epsilon_2 = 0$ **(b)** $\epsilon_1 = 0.3, \epsilon_2 = 0$ **(c)** $\epsilon_1 = 0, \epsilon_2 = 0.1$ **(d)** $\epsilon_1 = 0.3, \epsilon_2 = 0.1$

Figure 2.1. Visual representation of the probabilistic graph error model applied to a random geometric graph. From left to right: (a) Original graph; (b) Graph after edge deletions ($\epsilon_1 = 0.3, \epsilon_2 = 0$); (c) Graph after edge additions ($\epsilon_1 = 0, \epsilon_2 = 0.1$); (d) Graph after both edge deletions and additions ($\epsilon_1 = 0.3, \epsilon_2 = 0.1$). Deleted edges are marked in red and added edges are marked in blue. The transformations effectively illustrate the impact of perturbations modeled by (2.6).

$\boldsymbol{H} = \{\boldsymbol{H}_{\ell k}\}_{\ell=1,\dots,L;k=1,\dots,K}$, where $\boldsymbol{H}_{\ell k} \in \mathbb{R}^{F_{\ell-1} \times F_\ell}$. Recursively applying (2.2) up to the final layer $\ell = L$ defines the GCNN output

$$\boldsymbol{\Phi}(\boldsymbol{X}; \boldsymbol{H}, \boldsymbol{S}) = \boldsymbol{X}_L = \sigma\Big(\sum_{k=1}^{K} \boldsymbol{S}\boldsymbol{X}_{L-1}\boldsymbol{H}_{Lk}\Big). \tag{2.3}$$

This hierarchical structure allows GCNNs to learn complex representations from graph-structured data by iteratively transforming features based on graph topology.

2.2 Probabilistic graph error model

A general GSO error model can be presented as

$$\hat{\boldsymbol{S}} = \boldsymbol{S} + \boldsymbol{E}, \tag{2.4}$$

where $\hat{\boldsymbol{S}}$ is the perturbed GSO, $\boldsymbol{S}$ is the original GSO, and $\boldsymbol{E}$ is the error term. We define the GSO distance by the spectral norm of this error term.

$$d(\hat{\boldsymbol{S}}, \boldsymbol{S}) = \|\hat{\boldsymbol{S}} - \boldsymbol{S}\| = \|\boldsymbol{E}\|. \tag{2.5}$$

When a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is perturbed, it becomes $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}}, \hat{\mathcal{W}})$, with the node set $\mathcal{V}$ unaffected. we focus on the alterations within the neighboring nodes of a node $u \in \mathcal{V}$. The perturbed neighborhood may encompass added nodes ($\mathcal{A}_u$), deleted nodes ($\mathcal{D}_u$), and remaining nodes ($\mathcal{R}_u$), leading to changes in node degree and changes to the adjacency matrix. We denote degrees of node $u \in \mathcal{V}$ in original and perturbed graphs as $d_u = \sum_j |[\boldsymbol{A}]_{u,j}|$ and $\hat{d}_u = \sum_j |[\hat{\boldsymbol{A}}]_{u,j}| = d_u + \delta_u$, respectively. Here, $\hat{\boldsymbol{A}}$ denotes the adjacency matrix of the perturbed graph $\hat{\mathcal{G}}$, and $\delta_u = \delta_u^+ - \delta_u^-$ is the degree change at node u, with $\delta_u^+ = |\mathcal{A}_u|$ and $\delta_u^- = |\mathcal{D}_u|$ corresponding to the number of edges added and deleted, respectively.

We use an Erdös-Rényi (ER) graph-based model for perturbations on the graph adjacency matrix [52]. The adjacency matrix of an ER graph

is characterized by a random $N \times N$ matrix Δ_ϵ, where each element of the matrix is generated independently, satisfying $\Pr([\Delta_\epsilon]_{i,j} = 1) = \epsilon$ and $\Pr([\Delta_\epsilon]_{i,j} = 0) = 1 - \epsilon$ for all $i \neq j$. The diagonal elements are zero, i.e., $[\Delta_\epsilon]_{i,i} = 0$ for $i = 1, \ldots, N$, eliminating the possibility of self-loops. For our analysis, we also assume that the perturbed graph $\hat{\mathcal{G}}$ does not contain any isolated nodes, meaning that for all $u \in \mathcal{V}$, $\hat{d}_u \geq 1$[2]. The model can be adapted by employing the lower triangular matrix $\mathbf{\Delta}_\epsilon^l$, and then defining $\mathbf{\Delta}_\epsilon = \mathbf{\Delta}_\epsilon^l + (\mathbf{\Delta}_\epsilon^l)^\top$. Consequently, by specifying the error term in (2.4), the perturbed adjacency matrix of a graph signal can be expressed as

$$\hat{\boldsymbol{A}} = \boldsymbol{A} - \mathbf{\Delta}_{\epsilon_1} \circ \boldsymbol{A} + \mathbf{\Delta}_{\epsilon_2} \circ (\mathbf{1}_{N \times N} - \boldsymbol{A}), \tag{2.6}$$

where the first term is responsible for edge deletion with probability ϵ_1, and the second term accounts for edge addition with probability ϵ_2. This error model can be conceptualized as superimposing two ER graphs on top of the original graph. To better illustrate this model, we utilize visual aids based on a random geometric graph [57, 24]. Fig. 2.1 represents the transition from the original graph to perturbed versions, which include the graph with only edge deletions ($\epsilon_1 = 0.3, \epsilon_2 = 0$), the graph with only edge additions ($\epsilon_1 = 0, \epsilon_2 = 0.1$), and the graph with both edge deletions and additions ($\epsilon_1 = 0.3, \epsilon_2 = 0.1$). Each state depicts the progressive impacts of the perturbations.

In this context, the impact of the perturbation on the degree of a given node $u \in \mathcal{V}$ can be computed as follows. The effect of edge deletion is represented by $(-\mathbf{\Delta}_{\epsilon_1} \circ \boldsymbol{A})_u$, where each non-zero element in $\boldsymbol{A}_u$ has a probability of ϵ_1 being deleted. Thus, the total number of deleted edges δ_u^- is the sum of d_u independent and identically distributed (i.i.d.) Bernoulli random variables, each with a probability of ϵ_1. Similarly, the effect of edge addition is denoted by $\left(\mathbf{\Delta}_{\epsilon_2} \circ (\mathbf{1}_{N \times N} - \boldsymbol{A})\right)_u$, and the total number of added edges δ_u^+ is the sum of d_u^* i.i.d. Bernoulli random variables, each with a probability of ϵ_2, where $d_u^* = N - d_u - 1$. Hence, we can express the number of deleted edges δ_u^- and the number of added edges δ_u^+ as following binomial distributions:

$$\delta_u^- \sim \mathrm{Bin}(d_u, \epsilon_1),\ \delta_u^+ \sim \mathrm{Bin}(d_u^*, \epsilon_2), \tag{2.7}$$

where $\mathrm{Bin}(n, p)$ represents a binomial distribution with parameters n and p.

[2] We note that this is a subsequent assumption made for the purpose of the analysis, as the random error model itself could theoretically produce isolated nodes with a small probability. This condition is particularly important for the tractability of the analysis involving normalized GSOs, which are undefined if any node has a degree of zero.

2.3 Expected bound for GSO error

Based on the probabilistic error model, we first quantify the sensitivity of the GSO to graph structure perturbations. We examine the case where the adjacency matrix serves as the GSO, implying $\hat{\boldsymbol{S}} = \hat{\boldsymbol{A}}$ and $\boldsymbol{S} = \boldsymbol{A}$. Based on (2.6), the error model can be specified as

$$\boldsymbol{E} = \hat{\boldsymbol{A}} - \boldsymbol{A} = -\boldsymbol{\Delta}_{\epsilon_1} \circ \boldsymbol{A} + \boldsymbol{\Delta}_{\epsilon_2} \circ (\boldsymbol{1}_{N\times N} - \boldsymbol{A}). \tag{2.8}$$

The ℓ_1 norm of a matrix $\boldsymbol{E}$ is represented as $\|\boldsymbol{E}\|_1 = \max_j \sum_i |\boldsymbol{E}_{i,j}|$. Linking the change in degree with the ℓ_1 norm of (2.8), we have

$$\|\boldsymbol{E}\|_1 = \max_{u\in\mathcal{V}} \|\boldsymbol{E}_u\|_1, \tag{2.9}$$

where $\boldsymbol{E}_u$ denotes the uth row of matrix $\boldsymbol{E}$. Let

$$Y_u \triangleq \|\boldsymbol{E}_u\|_1 = |\mathscr{D}_u| + |\mathscr{A}_u| = \delta_u^- + \delta_u^+, \tag{2.10}$$

and

$$Y \triangleq \max_{u\in\mathcal{V}} Y_u. \tag{2.11}$$

Since δ_u^- and δ_u^+ are independent random variables, we derive expected value bounds, which are tighter and more informative than deterministic bounds. Bounds on expected error are better suited for analyzing the degree changes of nodes given the probabilistic nature of the model. Thus we derive a closed-form expression for the expectation of the maximum node degree error, i.e.,

$$\mathbb{E}[\|\boldsymbol{E}\|_1] = \mathbb{E}[\max_{u\in\mathcal{V}} \|\boldsymbol{E}_u\|_1]. \tag{2.12}$$

The probability mass function (PMF) of Y_u can be found by convolving the PMFs of δ_u^- and δ_u^+, which are independent random variables. Following binomial distributions in (2.7), we can obtain the following PMFs

$$\Pr_{\delta_u^-}(k) = \binom{d_u}{k} \epsilon_1^k (1-\epsilon_1)^{d_u-k},\ k = 0,\ldots,d_u, \tag{2.13}$$

$$\Pr_{\delta_u^+}(k) = \binom{d_u^*}{k} \epsilon_2^k (1-\epsilon_2)^{d_u^*-k},\ k = 0,\ldots,d_u^*, \tag{2.14}$$

where $d_u^* = N - d_u - 1$, $\Pr_{\delta_u^-}(k)$ and $\Pr_{\delta_u^+}(k)$ represent the probabilities of δ_u^- and δ_u^+ taking the value k, respectively. Then, the PMF of Y_u can be computed as

$$\Pr_{Y_u}(k) = \sum_{i=\max\{0,k-d_u^*\}}^{\min\{k,d_u\}} \Pr_{\delta_u^-,\delta_u^+}(i,k-i) = \sum_{i=\max\{0,k-d_u^*\}}^{\min\{k,d_u\}} \Pr_{\delta_u^-}(i)\Pr_{\delta_u^+}(k-i), \tag{2.15}$$

where $k = 0,\dots,N-1$. Using (2.15), the cumulative distribution function (CDF) of Y is computed as

$$\begin{aligned} \mathrm{F}_Y(k) &= \Pr(Y \le k) = \Pr(\max(Y_1,\dots,Y_N) \le k) \\ &= \Pr(Y_1 \le k,\dots,Y_N \le k) = \prod_{u=1}^{N} \Pr(Y_u \le k). \end{aligned} \tag{2.16}$$

Given that Y_u for $u \in \mathcal{V}$ are i.i.d. and for $k = 1,\dots,N-1$, the CDFs for Y and Y_u are as follows

$$\mathrm{F}_Y(k) = \prod_{u=1}^{N} \mathrm{F}_{Y_u}(k), \quad \mathrm{F}_{Y_u}(k) = \sum_{j=0}^{k} \mathrm{Pr}_{Y_u}(j). \tag{2.17}$$

With the PMF of Y taking on a specific value k being $\mathrm{Pr}_Y(k) = \mathrm{F}_Y(k) - \mathrm{F}_Y(k-1)$, the expectation of Y can be represented as

$$\mathbb{E}[Y] = \sum_{k=1}^{N-1} k \mathrm{Pr}_Y(k) = \sum_{k=1}^{N-1} k \left[\mathrm{F}_Y(k) - \mathrm{F}_Y(k-1)\right], \tag{2.18}$$

which provides a closed-form expression for $\mathbb{E}[Y] = \mathbb{E}[\|\boldsymbol{E}\|_1]$. The variance of Y can also be given as

$$\mathrm{Var}[Y] = \mathrm{Var}[\|\boldsymbol{E}\|_1] = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2, \tag{2.19}$$

where $\mathbb{E}[Y^2] = \sum_{k=1}^{N-1} k^2 \mathrm{Pr}_Y(k)$.

ℓ_1 norm-based bound for GSO error

In the analysis of graph-structured data, the spectral norm (ℓ_2 norm), is often used to quantify the graph spectral error. Assuming an undirected graph and perturbation, we have $\boldsymbol{E} = \boldsymbol{E}^\top$. Using inequalities $\|\boldsymbol{E}\|^2 \le \|\boldsymbol{E}\|_1 \|\boldsymbol{E}\|_\infty$ [20, Section 2.3.3] and the fact that in our case $\|\boldsymbol{E}\|_1 = \|\boldsymbol{E}\|_\infty$, the ℓ_2 norm can be bounded by the ℓ_1 norm

$$\|\boldsymbol{E}\| \le \|\boldsymbol{E}\|_1 = \max_{u \in \mathcal{V}} \|\boldsymbol{E}_u\|_1. \tag{2.20}$$

The entries in the error matrix $\boldsymbol{E}$ of equation (2.8) are random variables. We give an expected bound of (2.20)

$$\mathbb{E}[\|\boldsymbol{E}\|] \le \mathbb{E}[\|\boldsymbol{E}\|_1] = \mathbb{E}[\max_{u \in \mathcal{V}} \|\boldsymbol{E}_u\|_1], \tag{2.21}$$

which takes into account the distribution of the random variables, as well as the structural changes of the perturbed graph. Thus, we have the following theorem.

Theorem 1 (Publication II: Theorem 1)**.** *In the context of the probabilistic error model* (2.8)*, let GSO be adjacency matrix $\boldsymbol{S} = \boldsymbol{A}$, and perturbed GSO be*

$\hat{\boldsymbol{S}} = \hat{\boldsymbol{A}}$, then, a closed-form expression for the upper bound on the expectation of the GSO distance is given by

$$\mathbb{E}\left[d(\hat{\boldsymbol{S}}, \boldsymbol{S})\right] \leq \mathbb{E}[Y], \tag{2.22}$$

where $\mathbb{E}[Y]$ is computed using (2.18)*,* (2.17)*, and* (2.15)*.*

Theorem 1 provides a closed-form expression for the upper bound, which are explicitly dependent on the parameters (ϵ_1, ϵ_2) of the probabilistic error model in (2.8). Using a loose upper bound proposed in [1], we can bound (2.22) as

$$\mathbb{E}[Y] \leq \max_{1 \leq u \leq N} (d_u \epsilon_1 + d_u^* \epsilon_2) + \sqrt{\frac{N-1}{N} \sum_{u=1}^{N} \left(d_u \epsilon_1 (1-\epsilon_1) + d_u^* \epsilon_2 (1-\epsilon_2) \right)}. \tag{2.23}$$

Note that, (2.23) showcases how our bound in Theorem 1 is parameterized by the probabilities of adding and deleting edges. Thus, Theorem 1 precisely captures the resulting structural changes induced by the probabilistic error model.

ℓ_1 norm-based bound for normalized case

Next, the GSO is considered as the normalized version of the adjacency matrix, i.e., $\boldsymbol{S} = \boldsymbol{A}_{\mathrm{n}}$. The entries of the normalized adjacency matrix are as follows, $[\boldsymbol{A}_{\mathrm{n}}]_{u,v} = \frac{1}{\sqrt{d_u d_v}}$ if $(u,v) \in \mathscr{E}$, and $[\boldsymbol{A}_{\mathrm{n}}]_{u,v} = 0$ if $(u,v) \notin \mathscr{E}$. In [30], a closed form for $\|\boldsymbol{E}_u\|_1$ is proposed

$$\|\boldsymbol{E}_u\|_1 = \sum_{v \in \mathscr{D}_u} \frac{1}{\sqrt{d_u d_v}} + \sum_{v \in \mathscr{A}_u} \frac{1}{\sqrt{\hat{d}_u \hat{d}_v}} + \sum_{v \in \mathscr{R}_u} \left| \frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{\hat{d}_u \hat{d}_v}} \right|, \tag{2.24}$$

where $\hat{d}_u$ and $\hat{d}_v$ denote the degrees of node u and v after perturbation. However, the assumption in [30] states that the degree alteration $\hat{d}_v$ should not exceed twice the initial degree, i.e., $\hat{d}_v \leq 2d_v, v \in \{\mathscr{N}_u \cup u\}$. We remove this restriction by allowing an increased probability of edge addition ϵ_2 in error model (2.6).

We start with the following lemma.

Lemma 1 (Publication II: Lemma 1)**.** *Let $\boldsymbol{E}_u$ be defined as in* (2.24)*, then its ℓ_1 norm is bounded by a random variable Z_u*

$$\|\boldsymbol{E}_u\|_1 \leq Z_u = Z_{u,1} + Z_{u,2}, \tag{2.25}$$

where Z_u is defined as the sum of $Z_{u,1} = \sqrt{d_u/\tau_u}$ and

$$Z_{u,2} = \sum_{v \in \mathscr{A}_u \cup \mathscr{R}_u} \frac{1}{\sqrt{(d_u + \delta_u^+ - \delta_u^-)(d_v + \delta_v^+ - \delta_v^-)}},$$

d_u is the degree of node u, τ_u is the minimum degree of neighboring nodes of u, and $\delta_u^-, \delta_u^+, \delta_v^-, \delta_v^+$ are random variables with binomial distributions as $\delta_u^- \sim \mathrm{Bin}(d_u, \epsilon_1), \delta_u^+ \sim \mathrm{Bin}(d_u^*, \epsilon_2), \delta_v^- \sim \mathrm{Bin}(d_v, \epsilon_1), \delta_v^+ \sim \mathrm{Bin}(d_v^*, \epsilon_2)$ for $u \in \mathcal{V}$ and $v \in \mathscr{A}_u \cup \mathscr{R}_u$, where $d_u^* = N - d_u - 1$ and $d_v^* = N - d_v - 1$.

Let

$$Z \triangleq \max_{u \in \mathcal{V}} Z_u, \tag{2.26}$$

and note that Z_u and Z are discrete random variables. While the binomial random variables and degrees in the expression for Z are assumed to be i.i.d., the inherent nonlinearity and high-dimensionality in the function, along with the complexity introduced by the maximization operation over all nodes, pose challenges for deriving an analytical expression for $\mathbb{E}[Z]$. Furthermore, the expectation of a maximum of random variables often lacks a simple closed form with only bounds often being derivable, not the exact value. Monte Carlo simulations provide an efficient alternative for estimating $\mathbb{E}[Z]$

$$\mu_Z \triangleq \mathbb{E}[Z] \approx \frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} Z_{(i)} = \hat{\mu}_Z, \tag{2.27}$$

where $Z_{(i)}$ is the outcome from the i-th Monte Carlo trial. Thus, for the normalized GSO, the following proposition is the counterpart of Theorem 1.

Proposition 1 (Publication II: Proposition 1)**.** *In the context of the probabilistic error model $\boldsymbol{E} = \hat{\boldsymbol{S}} - \boldsymbol{S}$, let GSO be normalized adjacency matrix $\boldsymbol{S} = \boldsymbol{A}_n$, and perturbed GSO being $\hat{\boldsymbol{S}} = \hat{\boldsymbol{A}}_n$. Then, an upper bound on the expectation of the GSO distance is given by*

$$\mathbb{E}\left[d(\hat{\boldsymbol{S}}, \boldsymbol{S})\right] \leq \mathbb{E}[Z], \tag{2.28}$$

where the approximation of $\mathbb{E}[Z]$ is computed using (2.27)*,* (2.26)*, and Lemma 1.*

The upperbound provided in Proposition 1 focuses specifically on normalized adjacency matrices. This result complements the analysis for the unnormalized case. The bound for normalized GSO is a theoretical upperbound, instead of an approximation or an empirical estimation. The only difference between the bound in Proposition 1 and the bound in Theorem 1 is the computation. As for the bound in Theorem 1 (unnormalized case), $\mathbb{E}[Y]$ has a closed-form expression; while for computing the bound in Proposition 1 (normalized case) $\mathbb{E}[Z]$, we use Monte Carlo simulations (2.27).

2.4 Expected bound for GF and GCNN

2.4.1 GF sensitivity analysis

The sensitivity of graph filters is a critical topic that follows logically from the preceding discussion on the expected bounds of GSO errors. Graph

filters, being polynomials of GSOs, inherit the perturbations in the graph structure, manifesting as variations in filter responses.

The sensitivity of a graph filter to perturbations in the GSO is captured by the theorem below, which establishes a bound on the error in the graph filter response due to perturbations in the GSO and the filter coefficients.

Theorem 2 (Graph filter sensitivity; Publication II: Theorem 2)**.** *Let $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$ be the GSO for the true graph $\mathcal{G}$ and the perturbed graph $\hat{\mathcal{G}}$, respectively. The distance between polynomial graph filters $\boldsymbol{h}(\boldsymbol{S}) = \sum_{k=0}^{K} h_k \boldsymbol{S}^k$ and $\boldsymbol{h}(\hat{\boldsymbol{S}}) = \sum_{k=0}^{K} h_k \hat{\boldsymbol{S}}^k$ is defined as*

$$d\left(\boldsymbol{h}(\hat{\boldsymbol{S}}), \boldsymbol{h}(\boldsymbol{S})\right) = \|\boldsymbol{h}(\hat{\boldsymbol{S}}) - \boldsymbol{h}(\boldsymbol{S})\|. \tag{2.29}$$

The expectation of filter distance (2.29) *is bounded as*

$$\mathbb{E}\left[d\left(\boldsymbol{h}(\hat{\boldsymbol{S}}), \boldsymbol{h}(\boldsymbol{S})\right)\right] \leq \sum_{k=1}^{K} k|h_k|(\lambda_k \mathbb{E}[\|\mathbf{E}\|] + \zeta_k), \tag{2.30}$$

where $\lambda_k \triangleq \mathbb{E}[\lambda^{k-1}]$, $\zeta_k \triangleq \mathrm{Cov}[\|\boldsymbol{E}\|, \lambda^{k-1}]$, and $\lambda = \max\{\|\hat{\boldsymbol{S}}\|, \|\boldsymbol{S}\|\}$ denotes the largest of the maximum singular values of two GSOs.

Theorem 2 reveals that the expected graph filter distance is linearly bounded by the expected GSO distance, $\mathbb{E}[\|\boldsymbol{E}\|]$, if the sufficient condition $\lambda = \|\boldsymbol{S}\|$ is met. This bound is influenced by: the filter degree K, the maximum singular value λ of GSOs, and the filter coefficients $\{h_k\}_{k=1}^{K}$. The theorem indicates that higher order graph filters are likely to exhibit greater instability.

2.4.2 GCNN sensitivity analysis

Based on the sensitivity analysis of graph filter, we extend this study to the sensitivity analysis of the general GCNN. We present the following theorem to exemplify this approach, encapsulating the sensitivity of a general GCNN to GSO perturbations.

Theorem 3 (GCNN Sensitivity; Publication II: Theorem 3)**.** *For a general GCNN under the probabilistic error model* (2.8)*, the expected difference of outputs at the final layer L is given as*

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{X}}_L - \boldsymbol{X}_L\right\|\right] \leq C_{\sigma_L} B_L \mathbb{E}[\|\boldsymbol{E}\|] + C_{\sigma_L} D_L, \tag{2.31}$$

where C_{σ_ℓ} represents the Lipschitz constant for the nonlinear activation function used at layer ℓ, for $\ell = 1, \ldots, L$, B_ℓ and D_ℓ for $\ell = 1$ and then for

$\ell = 2,\ldots,L$ are defined as follows

$$B_1 = \sum_{k=1}^{K} k\lambda_k \|\boldsymbol{X}_0\| \|\boldsymbol{H}_{1k}\|, D_1 = \sum_{k=1}^{K} k\zeta_k \|\boldsymbol{X}_0\| \|\boldsymbol{H}_{1k}\|,$$

$$B_\ell = \sum_{k=1}^{K} \left(\lambda_{k+1} C_{\sigma_{\ell-1}} B_{\ell-1} + k\lambda_k \|\boldsymbol{X}_{\ell-1}\|\right) \|\boldsymbol{H}_{\ell k}\|, \tag{2.32}$$

$$D_\ell = \sum_{k=1}^{K} \left(\mu_{k,\ell-1} + \lambda_k C_{\sigma_{\ell-1}} D_{\ell-1} + k\zeta_k \|\boldsymbol{X}_{\ell-1}\|\right) \|\boldsymbol{H}_{\ell k}\|,$$

with constant $\mu_{k,\ell-1} \triangleq \sqrt{\mathrm{Var}[\|\hat{\boldsymbol{X}}_{\ell-1} - \boldsymbol{X}_{\ell-1}\|]\mathrm{Var}[\lambda^k]}$, and λ_k and ζ_k in Theorem 2, for $k = 1,\ldots,K$.

In Theorem 3, recursive bounds containing inter-layer features are used to simplify the formulation. Note that these inter-layer features $\{\boldsymbol{X}_{\ell-1}, \hat{\boldsymbol{X}}_{\ell-1}\}_{\ell=2}^{L}$ can be explicitly computed by the initial input feature $\boldsymbol{X}_0$, both original and perturbed GSOs $(\boldsymbol{S}, \hat{\boldsymbol{S}})$, GCNN parameters (number of layers L and graph shift K, network's learned weights $\{\boldsymbol{H}_{\ell k}\}$, and activation functions $\sigma(\cdot)$). The derivation process employs induction. For the first layer $\ell = 1$, we have $\boldsymbol{X}_1 = \sigma_1(\sum_{k=1}^{K} \boldsymbol{S}^k \boldsymbol{X}_0 \boldsymbol{H}_{1k})$ and $\hat{\boldsymbol{X}}_1 = \sigma_1(\sum_{k=1}^{K} \hat{\boldsymbol{S}}^k \boldsymbol{X}_0 \boldsymbol{H}_{1k})$; for the second layer $\ell = 2$, the features are $\boldsymbol{X}_2 = \sigma_2(\sum_{k=1}^{K} \boldsymbol{S}^k \boldsymbol{X}_1 \boldsymbol{H}_{2k})$ and $\hat{\boldsymbol{X}}_2 = \sigma_2(\sum_{k=1}^{K} \hat{\boldsymbol{S}}^k \hat{\boldsymbol{X}}_1 \boldsymbol{H}_{2k})$; by induction, for the $(\ell-1)$th layer, we have

$$\boldsymbol{X}_{\ell-1} = \sigma_\ell \left(\sum_{k=1}^{K} \boldsymbol{S}^k \boldsymbol{X}_{\ell-2} \boldsymbol{H}_{\ell-1,k} \right), \quad \hat{\boldsymbol{X}}_{\ell-1} = \sigma_\ell \left(\sum_{k=1}^{K} \hat{\boldsymbol{S}}^k \hat{\boldsymbol{X}}_{\ell-2} \boldsymbol{H}_{\ell-1,k} \right). \tag{2.33}$$

Theorem 3 forms the bedrock of our analysis, quantifying how GCNNs respond to graph perturbations, which is described by a linear relationship at each layer. The sensitivity of multilayer GCNN to perturbations can be represented by a recursion of linearity. For multilayer GCNN, its expected output difference is controlled by: *(i)* the input feature, *(ii)* the GSO, error model parameters, *(iii)* Lipschitz constants of activation functions, and *(iv)* GCNN weights. We note that, choosing activation functions with more conservative Lipschitz constants can possibly improve the stability of GCNNs by imposing more constraints on the recursion. However, this may suppress the performance of a neural network, as noted in [53]. Our sensitivity analysis framework is generic, allowing for simplifications such as assuming a unit Lipschitz constant and normalized input features, as suggested in [31]. However, these simplifications do not indicate that the GCNN sensitivity is unaffected by the Lipschitz constant or input features. This layered analysis also enables an understanding of how perturbations propagate through GCNN layers, impacting the overall performance. Additionally, Theorem 3 does not restrict the scale of graph perturbations, which is a typical restriction in the existing literature.

Given that the GSO error is bounded as in Theorem 1 and Proposition 1, the linear bound of each layer of GCNN permits the network's stability against perturbation as long as the graph error remains within the bound.

2.5 Specifications for GCNN variants

Building upon sensitivity analysis of Theorem 3, our discussion now evolves towards two specific GCNN variants - GIN [75] and SGCN [38, 72]. They apply different GSOs for feature propagation. In GIN, the GSO for each layer is chosen as a partially augmented unnormalized adjacency matrix; in SGCN, the GSO is chosen as a normalized augmented adjacency matrix. By GIN and SGCN, we are essentially extending our theoretical understanding to practical and real-world applications.

2.5.1 Specification for GIN

The GIN is designed to capture the node features and the graph structure simultaneously. The primary intuition behind GIN is to learn a function of the feature information from both the target node and its neighbors, which is related to the Weisfeiler-Lehman (WL) graph isomorphism test [70]. The chosen GSO for GIN is $\boldsymbol{S} = \boldsymbol{A} + (1+\varepsilon)\boldsymbol{I}$, where the learnable parameter ε preserves the distinction between nodes in the graph that are connected differently, and prevents GIN from reducing to a WL isomorphism test.

Given the GSO above, only the first order term with $K = 1$ in (2.1) is kept, and the intermediate output of such graph filter is $\boldsymbol{y} = \boldsymbol{S}\boldsymbol{x}$. A node Multilayer Perceptron (MLP) $\boldsymbol{h}_{\boldsymbol{\Theta}}$ is then applied to the filter's output as $\boldsymbol{h}_{\boldsymbol{\Theta}}(\boldsymbol{y})$. Assuming the inner MLP has two layers in each GIN layer, a single-layer GIN ($L = 1$) can be represented as

$$\boldsymbol{X}_L = \sigma_{L2}(\sigma_{L1}(\boldsymbol{S}\boldsymbol{X}_{L-1}\boldsymbol{W}_{L1} + \boldsymbol{B}_{L1})\boldsymbol{W}_{L2} + \boldsymbol{B}_{L2}), \tag{2.34}$$

where $(\boldsymbol{W}_{L1}, \boldsymbol{B}_{L1}, \sigma_{L1}(\cdot))$ are weight matrix, bias matrix, and nonlinearity function in the first layer of the MLP, and $(\boldsymbol{W}_{L2}, \boldsymbol{B}_{L2}, \sigma_{L2}(\cdot))$ are weight matrix, bias matrix, and nonlinearity function in the second layer of the MLP. Then, we provide the following corollary.

Corollary 1 (The sensitivity of single-layer GIN; Publication II: Corollary 1)**.** *For the single-layer GIN ($L = 1$) in* (2.34) *under the probabilistic error model* (2.8)*, the expected difference of outputs because of GSO perturbations is given as*

$$\mathbb{E}\left[\|\hat{\boldsymbol{X}}_L - \boldsymbol{X}_L\|\right] \leq \xi \mathbb{E}[\|\boldsymbol{E}\|], \tag{2.35}$$

with constant

$$\xi = C_{\sigma_{L2}} C_{\sigma_{L1}} \|\boldsymbol{W}_{L2}\| \|\boldsymbol{W}_{L1}\| \|\boldsymbol{X}_{L-1}\|, \tag{2.36}$$

where $\boldsymbol{X}_{L-1} = \boldsymbol{X}_0$ is the input feature, and $C_{\sigma_{L1}}$ and $C_{\sigma_{L2}}$ are Lipschitz constants of the two-layer MLP embedded in the GIN.

Corollary 1 shows a linear dependency between the output difference of a single-layer GIN and GSO perturbations. In GIN, node vector transformations by MLP contribute significantly to network's expressivity. Under evasion attacks, with Corollary 1, the analysis of these transformed node representations is straightforward.

2.5.2 Specification for SGCN

The SGCN is a streamlined model, developed by aiming to simplify a multilayered GCNN through the utilization of an affine approximation of graph convolution filter and the elimination of intermediate layer activation functions. The GSO chosen for SGCN is $\boldsymbol{S} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$, where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the augmented adjacency matrix and $\tilde{\boldsymbol{D}}$ is the corresponding degree matrix of the augmented graph.

Given the normalized augmented GSO, the node degrees $d_u, u = 1,\ldots,N$ are redefined based on the augmented GSO, specifically, they are incremented by 1 compared to their values in the non-augmented version. This streamlined model simplifies the structure of a vanilla GCN [33] by retaining a single layer and the Kth order GSO in (2.1), so the output of the filter is $\boldsymbol{y} = h_K\boldsymbol{S}^K\boldsymbol{x}$. Note that for a SGCN, the maximum number of layers is $L = 1$. Consequently, the output of a single-layer SGCN using a linear logistic regression layer is represented as

$$\boldsymbol{X}_L = \sigma_L(\boldsymbol{S}^K\boldsymbol{X}\boldsymbol{H}_K), \tag{2.37}$$

and thus, we can easily give the following corollary.

Corollary 2 (The sensitivity of SGCN)**.** *For the SGCN in* (2.37) *under the probabilistic error model* (2.8)*, the expected difference of outputs because of GSO perturbations is given as*

$$\mathbb{E}\left[\|\hat{\boldsymbol{X}}_L - \boldsymbol{X}_L\|\right] \leq C_{\sigma_L}B_L\mathbb{E}[\|\boldsymbol{E}\|] + C_{\sigma_L}D_L, \tag{2.38}$$

where $B_L = \lambda_K\|\boldsymbol{X}\|\|\boldsymbol{H}_K\|$, $D_L = K\zeta_K\|\boldsymbol{X}\|\|\boldsymbol{H}_K\|$, λ_K and ζ_K are defined in Theorem 3.

With Corollary 2, we conclude that the sensitivity analysis for SGCN is a specification for the general form of a multilayer GCNN.

2.6 Conclusion and discussion

This chapter presented an analytical framework to quantify the sensitivity of GCNNs to probabilistic structural perturbations. A general probabilistic

error model was employed to represent edge additions and deletions, upon which tight expected upper bounds for GSO errors were mathematically derived. The subsequent analysis focused on the propagation of these GSO errors through the network's layers.

The primary utility of these results is the establishment of quantitative stability guarantees. The derived expected GSO error bounds are explicit functions of the error model's parameters, allowing for a direct assessment of GSO perturbation. A key methodological advantage is that these bounds are computed without requiring the eigendecomposition of the GSO, thus ensuring computational tractability for large-scale graphs. The analysis of error propagation further reveals a linear dependency between the expected GSO error and the expected output difference at each GCNN layer. For multilayer networks, this relationship represents as a recursion of linearity, which provides a formal mechanism to demonstrate output sensitivity.

In conclusion, this chapter contributes to the theme of robustness as output stability under structural perturbations. The derived theoretical bounds and the analysis of their propagation provide the necessary tools to certify the stability of GCNNs in the presence of graph uncertainties. The focus now shifts in Chapter 3 to an orthogonal perspective on robustness: resilience to distributional deviations in measurement data.

3. Robust Activity Detection for Massive Random Access

As discussed in Chapter 1, accurately and efficiently identifying active devices in mMTC systems, the task of device AD, is fundamental for the successful operation of IoT applications. However, the performance of traditional AD methods, which often rely on Gaussian noise assumptions, can be severely compromised in realistic IoT environments. These environments are frequently characterized by complex non-Gaussian noise, including impulsive interference and heavy-tailed distributions, stemming from diverse sources such as industrial machinery or channel impairments [84, 22, 17, 44, 51, 10]. Such challenging noise conditions can lead to significant degradation in detection accuracy and overall system reliability.

This chapter confronts these challenges by developing novel and robust statistical frameworks for device AD specifically tailored for mMTC systems operating under such adverse non-Gaussian noise conditions. The primary objective is to design AD algorithms that not only achieve high accuracy and computational efficiency but also exhibit resilience to outliers and heavy-tailed interference. This involves a systematic approach, from formulating new AD objective functions based on robust loss principles to developing scalable algorithms with proven convergence and validating their superior performance. A key desideratum is that these robust algorithms maintain performance comparable to their Gaussian-based counterparts in benign noise scenarios while offering substantial gains when non-Gaussian characteristics are prevalent.

The methodologies and findings presented in this chapter are from Publication IV, which provides a comprehensive development of robust AD techniques. This work systematically builds upon and significantly extends the initial proposal of a robust matching pursuit algorithm introduced in Publication III.

The main contributions of this chapter include:

- **Robust objective formulation for AD.** Novel AD objective functions are formulated by replacing the conventional Gaussian negative log-likelihood with robust loss functions (e.g., Huber's, t-loss). This approach

directly mitigates the detrimental impact of outliers and heavy-tailed noise distributions, which may appear in practical IoT communication environments but challenging for traditional Gaussian-based AD methods.

- **Convergence analysis for algorithms.** A key theoretical contribution is the proof of geodesic convexity for the coordinate-wise (conditional) robust AD objective function. This allows the derivation of an efficient iterative fixed-point (FP) algorithm for its minimization, for which rigorous convergence guarantees are established. The derived algorithm serves as a core building block for solving the full AD problem.

- **Development of robust AD algorithms.** Building upon the FP algorithm, two distinct and comprehensive robust AD algorithms are proposed:

  – A Robust Coordinate-Wise Optimization (RCWO) algorithm, which cyclically updates device activity estimates.

  – A Robust Covariance Learning-based Matching Pursuit (RCL-MP) algorithm, which greedily selects active users by incorporating robust covariance learning within a matching pursuit framework. This significantly extends the preliminary RCL-MP concept introduced in Publication III.

Collectively, these contributions provide a principled and effective suite of tools for robust device activity detection in massive IoT systems. Addressing the challenges posed by non-Gaussian noise, this work enhances the reliability of AD, a critical component for efficient resource management and overall network performance in mMTC. This directly contributes to the goal of achieving *robustness as resilience to data distributional deviations*, a key perspective highlighted in Chapter 1.

3.1 Problem formulation

We consider an uplink single-cell massive random access scenario with N single-antenna Machine-Type Devices (MTDs) communicating with a BS equipped with M antennas. Each device n is preassigned a unique signature sequence $\boldsymbol{a}_n = (a_{n1}, \ldots, a_{nL})^\top$ of length L, which is known at the BS. In mMTC scenarios, the number of devices is greater than the pilot length (i.e., $N > L$ or $N \gg L$). We also assume sporadic user traffic, i.e., $K \ll N$ devices are active during each coherence interval.

The signal model is

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}. \tag{3.1}$$

Here, $\boldsymbol{Y} = (\boldsymbol{y}_1 \ \cdots \ \boldsymbol{y}_M) \in \mathbb{C}^{L\times M}$ denotes the received signal matrix over L signal dimensions (symbols) and M antennas. The columns of $\boldsymbol{Y}$ are independent due to independent and identically distributed (i.i.d.) channel coefficients (Rayleigh fading assumptions) over different antennas. The matrix $\boldsymbol{A} = (\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_N) \in \mathbb{C}^{L\times N}$ is the known pilot matrix. The unobserved channel matrix $\boldsymbol{X} = (\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_N)^\top \in \mathbb{C}^{N\times M}$ contains the user activity information. The i-th row of $\boldsymbol{X}$, $\boldsymbol{x}_i \in \mathbb{C}^{1\times M}$, denoted as

$$\boldsymbol{x}_i = \sqrt{\gamma_i}\boldsymbol{h}_i, \text{ for } i = 1,\ldots,N \tag{3.2}$$

models the channel between the i-th device and the BS. The vector $\boldsymbol{h}_i \sim \mathbb{C}\mathcal{N}_M(\boldsymbol{0},\boldsymbol{I})$ represents the Rayleigh fading component, and γ_i is the scaling component defined as

$$\gamma_i = \alpha_i \varrho_i \beta_i, \tag{3.3}$$

where $\alpha_i \in \{0,1\}$ is an indicator function of device activity (= 1 when device is active and = 0 otherwise), ϱ_i is the device's uplink transmission power, and $\beta_i > 0$ is the large-scale fading component (LSFC) accounting for pathloss and shadowing. For the noise matrix $\boldsymbol{E}$, we assume that its elements are independent and identically circular Gaussian distributed, i.e., $e_{lm} \sim \mathscr{CN}(0,\sigma^2)$, with known variance $\sigma^2 > 0$. Then the received signal, $\boldsymbol{y}_m \sim \mathscr{CN}_L(\boldsymbol{0},\boldsymbol{\Sigma})$ are i.i.d with $L \times L$ positive definite Hermitian (PDH) covariance matrix $\boldsymbol{\Sigma} = \mathrm{cov}(\boldsymbol{y}_m)$ given by [42]

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{H}} + \sigma^2\boldsymbol{I} = \sum_{i=1}^{N} \gamma_i \boldsymbol{a}_i \boldsymbol{a}_i^{\mathsf{H}} + \sigma^2\boldsymbol{I}, \tag{3.4}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ and $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_N)^\top$.

In massive MIMO systems, pilot design constitutes an important research area, and different designs yield varying theoretical performance guarantees [11, 14, 66]. Pilot sequence design directly affects the sample covariance matrix and subsequent AD problem conditioning. In [48, 47], nonorthogonal pilots are constructed from linear combinations of orthogonal sequences, such as Hadamard matrix derivatives. Variations including fully combined Hadamard matrix (FC-HADM) and restrictively combined Hadamard matrix (RC-HADM) have been proposed to improve detection performance. Another structured approach involves designing a sparse pilot matrix, where the non-zero patterns are guided by sparse graph codes [39]. In [14], pilot sequences are drawn uniformly and independently from a complex sphere of a given radius, a method for which strong theoretical recovery guarantees based on the Restricted Isometry Property (RIP)

can be established. While specialized pilot designs can offer performance benefits under specific conditions,this thesis focuses on random Bernoulli pilot sequences, which are are easy to implement for practical pilot design.

A key characteristic of massive mMTC is the sporadic nature of device traffic. At any given time, only a small subset of devices, $K \ll N$, is active. Thus, the signal matrix $\boldsymbol{X}$ is K-rowsparse, i.e., at most K rows of $\boldsymbol{X}$ contain non-zero entries. The rowsupport of $\boldsymbol{X} \in \mathbb{C}^{N \times M}$ is the index set of rows containing non-zero elements

$$\mathscr{M} = \operatorname{supp}(\boldsymbol{X}) = \{i \in [\![N]\!] : x_{ij} \neq 0 \text{ for some } j \in [\![M]\!]\}. \tag{3.5}$$

Thus, $\mathscr{M}$ collects the indices of the active devices, $\mathscr{M} = \{i \in [\![N]\!] : \alpha_i = 1\}$. Note that, $\mathscr{M} = \operatorname{supp}(\boldsymbol{X}) = \operatorname{supp}(\boldsymbol{\gamma})$. The objective of AD is to identify the support set $\mathscr{M}$, given the received signal $\boldsymbol{Y}$, the pilot matrix $\boldsymbol{A}$, and the noise level σ^2.

To solve the sparse AD problem, various algorithms have been developed. A prominent algorithm is Sparse Bayesian Learning (SBL) [71]. In the SBL framework, the sparse vector to be recovered is assumed to follow a zero-mean Gaussian distribution with with an unknown variance [71]. These variances are then estimated by maximizing the marginal likelihood of the observations via an Expectation-Maximization (EM) framework. This process effectively drives the variances of inactive users' channels towards zero, thereby identifying the sparse support set. Another influential algorithm is Approximate Message Passing (AMP) [13, 42], which employs iterative message passing techniques to approximate Bayesian inference. While powerful under certain random matrix assumptions, AMP's performance can be affected by specific structure of the pilot matrix $\boldsymbol{A}$. The AMP can achieve good performance for large i.i.d. random pilot matrices, particularly Gaussian ones [42]. Other widely-used approaches include greedy methods and iterative thresholding algorithms. An important greedy method is Simultaneous Orthogonal Matching Pursuit (SOMP), which iteratively builds the support set by identifying the pilot sequence most correlated with the current signal residual and then projecting the signal onto the subspace spanned by the selected sequences to update the estimate [65]. In parallel, Iterative Hard Thresholding (IHT), combines gradient descent steps on the least-squares objective with a hard thresholding operation that projects the current estimate onto the set of K-row-sparse matrices in each iteration [3].

Different from the introduced algorithms, this thesis focus on *covariance learning* (CL)-based support recovery algorithms. (CL)-based support recovery algorithms can be constructed by minimizing the Gaussian negative log-likelihood function (LLF) of the data $\boldsymbol{Y}$ defined by (after scaling by $1/M$

and ignoring additive constants)

$$\mathscr{L}(\boldsymbol{\gamma}) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_m + \log|\boldsymbol{\Sigma}|. \tag{3.6}$$

CL-based methods treat $\boldsymbol{\gamma}$ as a set of deterministic but unknown parameters and model the received signal $\boldsymbol{Y}$ based on the Gaussianity of both the source signal $\boldsymbol{X}$ and the noise $\boldsymbol{E}$. This distribution assumption formulates the problem as a maximum likelihood estimation (MLE) problem, making it tractable for analysis and implementation. Two notable CL-based algorithms are Coordinate-Wise Optimization (CWO), which iteratively updates the power estimate for each user in a cyclic manner [14], and Covariance Learning-based Matching Pursuit (CL-MP), a greedy method that sequentially identifies active users based on their contribution to the covariance structure [49], which is inspired by the Covariance Learning-based orthogonal matching pursuit (CL-OMP) algorithm in [54].

However, practical scenarios may differ from Gaussian assumptions due to outliers (leading to heavy-tailed distributions of received signals), imperfect or outdated channel state information or variations in device mobility. Such outliers can be caused by sporadic, strong interference from other systems, unexpected channel conditions, or faulty sensor behavior. In such cases, the traditional Gaussian assumption may become invalid, leading to potential issues in estimation accuracy. The Gaussian negative LLF is non-robust because it is based on the squared error loss, which assigns equal weights for all samples. Consequently, outliers can dominate the objective function, skewing the estimation of the covariance matrix $\boldsymbol{\Sigma}$ and leading to inaccurate AD. Therefore, robust methods that are less sensitive to these deviations are required to effectively address situations where the Gaussianity assumption is not valid.

3.2 Theory of proposed robust method

Our goal is robust detection of active devices. Robust methods use robust loss functions to downweight the (squared) Mahalanobis distances $\{\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_m\}_{m=1}^{M}$ to reduce the impact of outliers [50, 84]. Recall that a function $\rho(t)$ is said to be geodesically convex in $t \in \mathbb{R}_{\geq 0}$ if $r(x) = \rho(e^x)$ is convex in $x \in \mathbb{R}$. We define robust loss function as follows.

Definition 1 (Loss function; Publication IV, Definition 1)**.** *A geodesically convex function $\rho(t)$ that is continuous in $t \in \mathbb{R}_{>0}$, non-decreasing and differentiable, is called a loss function, and its first derivative*

$$u(t) = \rho'(t) \geq 0 \tag{3.7}$$

is called a weight function.

Then, a more general version of (3.6) is defined by

$$\mathscr{L}(\boldsymbol{\gamma}) = \frac{1}{bM}\sum_{m=1}^{M}\rho\left(\boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y}_m\right) + \log|\boldsymbol{\Sigma}|, \tag{3.8}$$

where ρ is a loss function in sense of Definition 1. This robust loss function generalizes the Gaussian MLE in (3.6) by applying $\rho(\cdot)$ to the Mahalanobis distance, reducing the impact of outliers and handling non-Gaussian noise effectively. The term b is a consistency factor defined as

$$b = \mathsf{E}[\psi(\boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y})]/L,\ \boldsymbol{y} \sim \mathbb{C}\mathscr{N}_L(\boldsymbol{0},\boldsymbol{\Sigma}), \tag{3.9}$$

where $\psi(t) = t\rho'(t)$. This consistency factor is used in M-estimation for obtaining consistency of the obtained estimator to covariance matrix when data is Gaussian-distributed [50, 84].

The most widely used loss function ρ that can be used in (3.8) is *Gaussian loss* $\rho(t) = t$. Its weight function is $u(t) = 1$, and consistency factor is $b = 1$. Hence the negative LLF (3.8) reduces to (3.6). The non-robustness of the Gaussian negative log-likelihood function (3.6) stems from its implicit use of linear loss on the Mahalanobis distance, which can be sensitive to outliers. To address this, we replace the Gaussian loss with a robust loss function, a central concept in robust statistics [84].

A robust loss function is characterized by its ability to mitigate the influence of atypical data points. This is typically achieved through the weight function $u(t)$, that is bounded or descends towards zero as its argument t (representing the squared Mahalanobis distance) increases. This property ensures that observations with large deviations from the model, i.e., outliers, are assigned a lower weight or have a bounded influence on the overall objective function, preventing them from dominating the parameter estimation.

A widely-used robust loss function is Huber's loss [28]. Huber's loss is based on a weight function

$$u(t;c) = \begin{cases} 1, & \text{for } t \le c^2, \\ c^2/t, & \text{for } t > c^2, \end{cases} \tag{3.10}$$

where $c > 0$ is a tuning constant that controls robustness (how heavily one down weights large distances). The loss function ρ corresponding to Huber's M-estimator is [55]:

$$\rho(t;c) = \begin{cases} t & \text{for } t \leqslant c^2, \\ c^2\left(\log(t/c^2)+1\right) & \text{for } t > c^2. \end{cases} \tag{3.11}$$

As $t = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} \sim (1/2)\chi^2_{2L}$ when $\boldsymbol{y} \sim \mathscr{CN}_L(\boldsymbol{0},\boldsymbol{\Sigma})$, it is common to choose c^2 as the qth quantile of $(1/2)\chi^2_{2L}$-distribution, i.e., verifying $2c^2 = F^{-1}(q;\chi^2_{2L})$ for some $q \in (0,1)$, where $F(\cdot;\chi^2_{2L})$ designates the cumulative density function

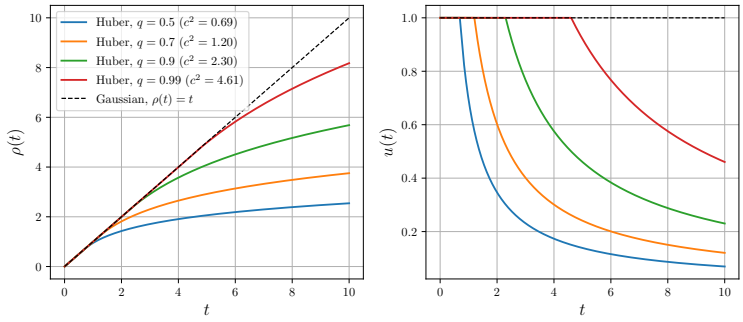


Figure 3.1. Comparison of Huber's loss function with the standard Gaussian loss. *Left*: Loss function $\rho(t)$ versus the squared Mahalanobis distance t. *Right*: Corresponding weight function $u(t)$ versus t. Huber's loss is set up with various quantiles $q \in \{0.5, 0.7, 0.9, 0.99\}$ with $L = 1$. For $t \leq c^2$, Huber's loss behaves like Gaussian loss, transitioning to logarithmic growth for $t > c^2$, effectively down-weighting outliers.

(cdf) of χ^2_{2L}-distribution. We regard $q \in (0,1)$ as a loss parameter which can be chosen by design, and use $q = 0.9$ as our default choice. The optimal selection of this parameter, which often involves a trade-off between robustness and statistical efficiency. For $q \to 1$, Huber's loss equals Gaussian loss. The scaling constant $b > 0$ can be expressed in closed form as

$$b = F_{\chi^2_{2(L+1)}}(2c^2) + c^2(1 - F_{\chi^2_{2L}}(2c^2))/L. \tag{3.12}$$

The behavior of Huber's loss compared to the Gaussian loss is illustrated in Fig. 3.1. For small squared distances ($t \leq c^2$), Huber's loss is identical to the Gaussian loss, treating these observations as inliers. However, for large distances ($t > c^2$), where the Gaussian loss continues to grow quadratically, Huber's loss transitions to a logarithmic growth. This effectively down-weights the influence of outliers. As the tuning parameter c^2 increases, the point at which this down-weighting begins is pushed further out, and as $c \to \infty$, Huber's loss converges to the standard Gaussian loss. Conversely, a smaller c leads to more aggressive down-weighting, providing greater robustness at the potential cost of some efficiency if the data is purely Gaussian.

The behavior of Huber's loss compared to the Gaussian loss is illustrated in Fig. 3.1. As shown in the left panel, for small squared distances ($t \leq c^2$), Huber's loss is identical to the Gaussian loss, treating these observations as inliers. However, for large distances ($t > c^2$), where the Gaussian loss continues to grow linearly, Huber's loss transitions to a much slower logarithmic growth. This effectively down-weights the influence of outliers. The right panel of Fig. 3.1 demonstrates this through the corresponding weight functions. While the Gaussian loss has a constant weight of 1, the weight function of Huber's loss descends as c^2/t for $t > c^2$, assigning less weights to observations with larger distances. The tuning parameter c^2, determined by the quantile q, controls the point at which this down-

weighting begins. As $q \to 1$ (and thus $c^2 \to \infty$), Huber's loss converges to the standard Gaussian loss, reducing robustness but increasing efficiency on clean Gaussian data. Conversely, a smaller q (and a smaller c^2) leads to more aggressive down-weighting, providing greater robustness against outliers, potentially at the cost of some statistical efficiency if the data is purely Gaussian. This trade-off between robustness and efficiency is important in robust statistics. This adaptability allows the proposed robust algorithms to achieve performance nearly identical to their Gaussian-based counterparts in Gaussian noise environments by selecting a quantile q close to 1 (see the red curve with $q = 0.99$ in Fig. 3.1).

3.2.1 Solving the Conditional Objective

We first consider the simpler case, the conditional objective function where all device powers $\{\gamma_j\}$ for $j \neq i$ are known. The conditional objective function for the single unknown i-th source power γ is

$$\mathscr{L}_i(\gamma \mid \boldsymbol{\Sigma}_{\setminus i}) = \frac{1}{bM} \sum_{m=1}^{M} \rho(s_{m,i}(\gamma)) + \log|\boldsymbol{\Sigma}_{\setminus i} + \gamma \boldsymbol{a}_i \boldsymbol{a}_i^{\mathsf{H}}|, \tag{3.13}$$

where

$$\boldsymbol{\Sigma}_{\setminus i} = \sum_{j \neq i} \gamma_j \boldsymbol{a}_j \boldsymbol{a}_j^{\mathsf{H}} + \sigma^2 \boldsymbol{I} = \boldsymbol{\Sigma} - \gamma_i \boldsymbol{a}_i \boldsymbol{a}_i^{\mathsf{H}} \tag{3.14}$$

is the covariance matrix of $\boldsymbol{y}_m$-s after the contribution from the i-th device is removed, and

$$s_{m,i}(\gamma) = \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_m. \tag{3.15}$$

Denoting

$$\boldsymbol{b}_i = \boldsymbol{\Sigma}_{\setminus i}^{-1} \boldsymbol{a}_i,\ d_i = \boldsymbol{a}_i^{\mathsf{H}} \boldsymbol{b}_i,\ i = 1, \ldots, N, \tag{3.16}$$

(3.15) can be written as

$$s_{m,i}(\gamma) = \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}_{\setminus i}^{-1} \boldsymbol{y}_m - \frac{\gamma}{1 + \gamma d_i} |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2. \tag{3.17}$$

Setting the first derivative of $\mathscr{L}_i$ with respect to γ to zero shows that the minimizer of (3.13) must verify the FP equation

$$\gamma = \mathscr{H}_i(\gamma) = \frac{\hat{\sigma}_{\gamma,i}^2 - d_i}{d_i^2}, \tag{3.18}$$

where

$$\hat{\sigma}_{\gamma,i}^2 = \frac{1}{bM} \sum_{m=1}^{M} u_{m,i}(\gamma) |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2. \tag{3.19}$$

Note that we use a shorthand notation $u_{m,i}(\gamma) = u(s_{m,i}(\gamma))$. The detailed derivation of (3.18) can be found in Publication IV.

3.2.2 FP Algorithm

We propose a FP algorithm for solving the minimizer of $\mathscr{L}_i(\gamma) \equiv \mathscr{L}_i(\gamma \mid \boldsymbol{\Sigma}_{\setminus i})$ for each coordinate/device $i = 1,\ldots,N$. We derive a FP algorithm using (3.18) that proceeds as follows. Start with an initial guess $\gamma^{(0)} \geq 0$ and perform the iterative procedure

$$\gamma^{(\ell+1)} = \mathscr{H}_i(\gamma^{(\ell)}), \tag{3.20}$$

which implies computing the following steps

$$s_{m,i}^{(\ell)} = \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}_{\setminus i}^{-1} \boldsymbol{y}_m - \frac{\gamma^{(\ell)}}{1+\gamma^{(\ell)} d_i} |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2, \tag{3.21a}$$

$$\hat{\sigma}_{\gamma,i}^{2(\ell)} = \frac{1}{bM} \sum_{m=1}^{M} u(s_{m,i}^{(\ell)}) |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2, \tag{3.21b}$$

$$\gamma^{(\ell+1)} = \frac{\hat{\sigma}_{\gamma,i}^{2(\ell)} - d_i}{d_i^2}. \tag{3.21c}$$

The iterative procedure given by (3.21a)-(3.21c) defines a sequence $\{\gamma^{(\ell)}\}$. Next, we establish geodesic convexity of the conditional objective function.

Theorem 4 (Geodesic convexity; Publication IV, Theorem 1)**.** *If $\rho(x)$ is a loss function in sense of Definition 1, then $\mathscr{L}_i(\gamma)$ is geodesically convex in $\gamma \in \mathbb{R}_{\geq 0}$.*

Following the geodesical convexity established in Theorem 4, we now characterize the global minimizer properties of the conditional objective function $\mathscr{L}_i(\gamma)$.

Theorem 5 (Existence and uniqueness; Publication IV, Theorem 2)**.** *If $\rho(\cdot)$ is a loss function in Definition 1, then:*

(a) Any local minimum of $\mathscr{L}_i(\gamma)$ is a global minimum.

(b) If $\rho(\cdot)$ is bounded below, then $\mathscr{L}_i(\gamma)$ has at least one global minimizer.

(c) If, in addition, $\rho(\cdot)$ is strictly geodesically convex, then the global minimizer of $\mathscr{L}_i(\gamma)$ is unique.

Theorem 5 has established the existence and uniqueness properties of the global minimizer for the conditional objective function $\mathscr{L}_i(\gamma)$. Next we show that for any $\gamma^{(\ell)}$ in the sequence $\{\gamma^{(\ell)}\}$ for $\ell = 0,1,2,\ldots$, that is not a stationary point of the objective function $\mathscr{L}_i(\gamma)$, the objective function decreases at successive iterations, i.e., the $\{\mathscr{L}_i(\gamma^{(\ell)})\}$ forms a monotone decreasing sequence in $\mathbb{R}$.

Algorithm 1 FP-CW: **FP** algorithm for **c**oordinate-**w**ise objective function

Input: $(\boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{\Sigma}_{\setminus i}^{-1}\boldsymbol{y}_m)_{M\times 1}$, $(|\boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{b}_i|^2)_{M\times 1}$, $d_i > 0$, weight function $u(\cdot)$, initial device power $\hat{\gamma}_{\text{init}} \geq 0$
Initialize: convergence threshold: $\delta = 5\cdot 10^{-3}$; a number close to 0: $\epsilon = 10^{-3}$; maximum number of iterations: $I_{\max} = 10$; $\hat{\gamma} = \hat{\gamma}_{\text{init}}$.
if $\mathscr{L}_i(0) < \mathscr{L}_i(\epsilon)$ **then**
    $\hat{\gamma} \leftarrow 0$
else ▷ Minimum is not on boundary, so use FP algorithm
    $\hat{\gamma}^{\text{old}} \leftarrow \hat{\gamma}_{\text{init}}$
    for $\ell = 1, \ldots, I_{\max}$ **do**
        $s_{m,i} \leftarrow \boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{\Sigma}_{\setminus i}^{-1}\boldsymbol{y}_m - \dfrac{\hat{\gamma}}{1+\hat{\gamma}d_i}|\boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{b}_i|^2$
        $\hat{\sigma}_{\gamma,i}^2 \leftarrow \dfrac{1}{bM}\displaystyle\sum_{m=1}^{M} u(s_{m,i})|\boldsymbol{y}_m^{\mathsf{H}}\boldsymbol{b}_i|^2$
        $\hat{\gamma} \leftarrow \dfrac{\hat{\sigma}_{\gamma,i}^2 - d_i}{d_i^2}$
        if $\ell = I_{\max}$ **or** $|\hat{\gamma} - \hat{\gamma}^{\text{old}}|/\hat{\gamma}^{\text{old}} < \delta$ **then**
            Break
        end if
        $\hat{\gamma}^{\text{old}} \leftarrow \hat{\gamma}$
    end for
end if
Return: optimal $\hat{\gamma} \in \mathbb{R}_{\geq 0}$

Theorem 6 (Monotonic decrease; Publication IV, Theorem 3)**.** *Assume that* $\rho(\cdot)$ *has second derivative with* $\rho''(\cdot) \leq 0$*. Let* $\gamma^{(\ell+1)} > 0$*, and* $\gamma^{(\ell)} > 0$ *be two successive iterations obtained from the FP mapping* (3.20)*. If* $\gamma^{(\ell)}$ *is not a stationary point of* $\mathscr{L}_i(\gamma)$*, then*

$$\mathscr{L}_i(\gamma^{(\ell+1)}) \leq \mathscr{L}_i(\gamma^{(\ell)}). \tag{3.22}$$

The inequality in (3.22) *is strict if* $\rho(\cdot)$ *is increasing.*

Theorem 6 shows that the conditional negative LLF decreases at each step of the FP algorithm and the negative LLF's values $\{\mathscr{L}_i(\gamma^{(\ell)})\}_\ell$ form a monotonic sequence.

- Interior case ($\gamma > 0$): The geodesic convexity ensures that any local minimum is a global minimum. When $\rho(\cdot)$ is bounded below, the function $\mathscr{L}_i(\gamma)$ is bounded below and coercive, guaranteeing that the FP iterations remain bounded. In practice, $\rho(\cdot)$ is selected as an increasing function (see for example (3.11)). Thus, (3.22) holds as strict inequality and FP iterations converge to the global minimum.

- Boundary case ($\gamma = 0$): Algorithm 1 (FP-CW) efficiently identifies this

case by comparing $\mathscr{L}_i(0)$ with $\mathscr{L}_i(\epsilon)$ for a small positive ϵ. If $\mathscr{L}_i(0) < \mathscr{L}_i(\epsilon)$, the algorithm confirms $\hat{\gamma} = 0$ as the minimizer without further iterations.

Algorithm 1 summarizes the FP mapping given by (3.21a)-(3.21c) which fits the coordinate-wise optimization in the AD problem. The major computational complexity of the FP-CW algorithm lies in the input stage, where the term $\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}_{\setminus i}^{-1} \boldsymbol{y}_m$ is computed for $m = 1, \ldots, M$, resulting in a complexity of $\mathcal{O}(ML^2)$.

3.3 Proposed methods

Based on Algorithm 1, we propose two robust algorithms for AD task. The first is the Robust coordinatewise optimization (RCWO) algorithm and the second is the Robust CL-based matching pursuit (RCL-MP) algorithm.

3.3.1 RCWO

The RCWO algorithm is based on the principle of coordinate-wise optimization, a powerful iterative strategy for solving complex multi-variable optimization problems. The core idea of this approach, also known as coordinate descent, is to decompose a optimization problem over many variables into a sequence of simpler block-wise subproblems. In each step of the iterative process, the algorithm focuses on optimizing the objective function with respect to a single variable, while keeping all other variables fixed at their current values. This process is then repeated, cycling through all variables until a convergence criterion is met. The coordinate-wise methods are effective when single-variable subproblems are easier to solve than the original joint problem.

The RCWO algorithm updates the parameters $\gamma_1, \ldots, \gamma_N$ cyclically by solving

$$\min_{\gamma} \mathscr{L}_i(\gamma)$$

for $i = 1, \ldots, N$. In RCWO, each device power γ_i is updated one at a time via the FP-CW algorithm (Algorithm 1), and afterwards the estimation of covariance matrix $\boldsymbol{\Sigma}$ is updated by the new γ_i. The algorithm is presented in Algorithm. 2.

3.3.2 RCL-MP

The RCL-MP algorithm is conceptually based on the principles of matching pursuit, a class of greedy iterative algorithms widely used for sparse signal recovery in the conventional sparse linear model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$ [65]. The fundamental idea of MP is to approximate a signal by iteratively selecting

Algorithm 2 RCWO: **R**obust **c**oordinate**w**ise **o**ptimization algorithm

Input: $\boldsymbol{A}$, $\boldsymbol{Y}$, σ^2, K, function $u(\cdot)$.
Initialize: $\boldsymbol{\Sigma}^{-1} = \sigma^{-2}\boldsymbol{I}$, $\hat{\boldsymbol{\gamma}} = \boldsymbol{0}$, number of cycles $I_{\text{cyc}} = 50$, and stopping threshold for CWO algorithm $\delta_{\text{CWO}} = 5 \cdot 10^{-3}$.
for $i_{\text{cyc}} = 1, \ldots, I_{\text{cyc}}$ **do**
  for $i = 1, \ldots, N$ **do**
    $\boldsymbol{c}_i \leftarrow \boldsymbol{\Sigma}^{-1}\boldsymbol{a}_i$
    $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Sigma}^{-1} + \dfrac{\hat{\gamma}_i \boldsymbol{c}_i \boldsymbol{c}_i^{\mathsf{H}}}{1 - \hat{\gamma}_i \boldsymbol{a}_i^{\mathsf{H}} \boldsymbol{c}_i}$
    $\boldsymbol{B} = \begin{pmatrix} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_N \end{pmatrix} \leftarrow \boldsymbol{\Theta}\boldsymbol{A}$
    $\tilde{\boldsymbol{s}} = (\tilde{s}_m)$, where $\tilde{s}_m \leftarrow \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Theta} \boldsymbol{y}_m$, $m = 1, \ldots, M$.
    $\boldsymbol{v} = (v_m)$, where $v_m \leftarrow |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2$, $m = 1, \ldots, M$.
    $d_i \leftarrow \boldsymbol{a}_i^{\mathsf{H}} \boldsymbol{b}_i$
    $\hat{\gamma}_i^{\text{new}} \leftarrow$ **FP-CW**$(\tilde{\boldsymbol{s}}, \boldsymbol{v}, d_i, u(\cdot), \hat{\gamma}_i)$
    $\delta_i \leftarrow \hat{\gamma}_i^{\text{new}} - \hat{\gamma}_i$
    $\boldsymbol{\Sigma}^{-1} \leftarrow \boldsymbol{\Sigma}^{-1} - \dfrac{\delta_i \boldsymbol{c}_i \boldsymbol{c}_i^{\mathsf{H}}}{1 + \delta_i \boldsymbol{a}_i^{\mathsf{H}} \boldsymbol{c}_i}$
    $\hat{\gamma}_i \leftarrow \hat{\gamma}_i^{\text{new}}$
  end for
  if $\|\hat{\boldsymbol{\gamma}} - \hat{\boldsymbol{\gamma}}^{\text{new}}\|_\infty / \|\hat{\boldsymbol{\gamma}}^{\text{new}}\|_\infty < \delta_{\text{CWO}}$ **then**
    Break
  end if
end for
$\mathscr{M} \leftarrow \text{supp}_K\left(\hat{\boldsymbol{\gamma}}\right)$
Return: Set $\mathscr{M}$ of indices for active devices ($|\mathscr{M}| = K$)

atoms (columns) from a dictionary (the matrix $\boldsymbol{A}$) that best match the current signal residual. In each iteration k, the algorithm identifies the atom $\boldsymbol{A}_i$ (the ith column of $\boldsymbol{A}$) that is most correlated with the current residual $\mathbf{r}^{(k-1)} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(k-1)}$, where $\boldsymbol{x}^{(k-1)}$ is the sparse estimate from the previous iteration. The selection criterion is thus

$$i_k = \underset{i \notin \mathscr{M}^{(k-1)}}{\text{argmax}} |\langle \boldsymbol{r}^{(k-1)}, \boldsymbol{A}_i \rangle|.$$

The index i_k is then added to the support set, the estimate $\boldsymbol{x}^{(k)}$ is updated, and the residual is recomputed for the next iteration. This process greedily builds up a sparse representation of the signal.

While traditional MP methods operate on the signal residual, the RCL-MP algorithm adapts this greedy selection philosophy to the CL domain. An equivalence can be drawn: instead of finding the atom $\boldsymbol{A}_i$ that maximizes the correlation with the signal residual, RCL-MP seeks to find the most influential user index i by identifying the user who gives the minimum value of the robust negative log-likelihood objective function (3.13). This is achieved by solving the one-dimensional optimization problem for

Algorithm 3 RCL-MP: **R**obust **CL**-based **m**atching **p**ursuit algorithm

1: **Input:** $\boldsymbol{A}$, $\boldsymbol{Y}$, σ^2, K, and function $u(\cdot)$.
2: **Initialize:** $\boldsymbol{\Theta} = \sigma^{-2}\boldsymbol{I}$, $\mathscr{M} = \emptyset$, $\hat{\boldsymbol{\gamma}} = \boldsymbol{0}$.
3: **for** $k = 1, \ldots, K$ **do**
4: $\quad \boldsymbol{B} = \begin{pmatrix} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_N \end{pmatrix} \leftarrow \boldsymbol{\Theta}\boldsymbol{A}$
5: $\quad \tilde{\boldsymbol{s}} = (\tilde{s}_m)$, where $\tilde{s}_m \leftarrow \boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Theta} \boldsymbol{y}_m$, $m = 1, \ldots, M$.
6: $\quad$ **for** $i \in \mathscr{M}^{\complement}$ **do**
7: $\quad\quad \boldsymbol{v} = (v_m)$, where $v_m \leftarrow |\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{b}_i|^2$, $m = 1, \ldots, M$.
8: $\quad\quad d_i \leftarrow \boldsymbol{a}_i^{\mathsf{H}} \boldsymbol{b}_i$
9: $\quad\quad \hat{\gamma}_i \leftarrow \textbf{FP-CW}(\tilde{\boldsymbol{s}}, \boldsymbol{v}, d_i, u(\cdot), \hat{\gamma}_i)$
10: $\quad\quad s_{m,i} \leftarrow \tilde{s}_m - \dfrac{\hat{\gamma}_i}{1 + \hat{\gamma}_i d_i} v_m,\ m = 1, \ldots, M$
11: $\quad\quad \epsilon_i \leftarrow \frac{1}{bM} \sum_{m=1}^{M} \rho(s_{m,i}) - \log|\boldsymbol{\Theta}| + \log(1 + \hat{\gamma}_i d_i)$
12: $\quad$ **end for**
13: $\quad \mathscr{M} \leftarrow \mathscr{M} \cup \{i_k\}$ with $i_k \leftarrow \operatorname{argmin}_{i \in \mathscr{M}^{\complement}} \epsilon_i$
14: $\quad \boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \dfrac{\hat{\gamma}_{i_k}}{1 + \hat{\gamma}_{i_k} d_{i_k}} \boldsymbol{b}_{i_k} \boldsymbol{b}_{i_k}^{\mathsf{H}}$
15: **end for**
16: **Return:** Set $\mathscr{M}$ of indices for active devices ($|\mathscr{M}| = K$)

each inactive user i,

$$\epsilon_i = \min_{\gamma_i \geq 0} \mathscr{L}_i(\gamma). \tag{3.23}$$

The active user can be identified by $i = \operatorname{argmin}_j \epsilon_j$. The user corresponding to the minimum value of this objective, $\operatorname{argmin}_i \epsilon_i$, is then greedily added to the support set, as this choice yields the greatest decrease in the overall LLF. The algorithm is presented in Algorithm. 3.

3.3.3 Computational complexity

The major complexity of the RCWO algorithm depends on the FP-CW algorithm in line 11 of Algorithm 2, which needs the complexity of $\mathscr{O}(TML^2)$. Hence, for each cycle, the complexity of RCWO algorithm is $\mathscr{O}(TMNL^2)$, where N indicates the total number of devices. For the Gaussian-based CWO algorithm introduced in [25, 7, 14], its complexity is $\mathscr{O}(NL^2)$, which is less than that of the proposed RCWO algorithm. Still, the increased complexity $\mathscr{O}(TMNL^2)$ remains reasonable and can be justified by the improved robustness and performance.

The major complexity of the RCL-MP algorithm depends on the FP-CW algorithm in line 9 of Algorithm 3, which needs the complexity of $\mathscr{O}(TML^2)$. For each increment in k by 1, the sweeping range $|(\mathscr{M}^{(k-1)})^{\complement}|$ reduces by 1, where $|(\mathscr{M}^{(0)})^{\complement}| = N$ for $k = 1$. Hence, for each iteration of RCL-MP algorithm, the worst-case complexity is $\mathscr{O}(TMNL^2)$. The computational complexity of RCLMP algorithm (Algorithm 3) is the same as for RCWO

Table 3.1. Computational complexity of AD algorithms.

Algorithm	Complexity (per iteration)
CL-MP [49]	$\mathcal{O}(NL^2)$
CWO [14]	$\mathcal{O}(NL^2)$
VAMP [43]	$\mathcal{O}(N^2M)$
SBL [71]	$\mathcal{O}(N^2L)$
SOMP [65]	$\mathcal{O}(L\cdot\max(NM,k^2)+k^3)$
RCWO (proposed)	$\mathcal{O}(TMNL^2)$
RCL-MP (proposed)	$\mathcal{O}(TMNL^2)$

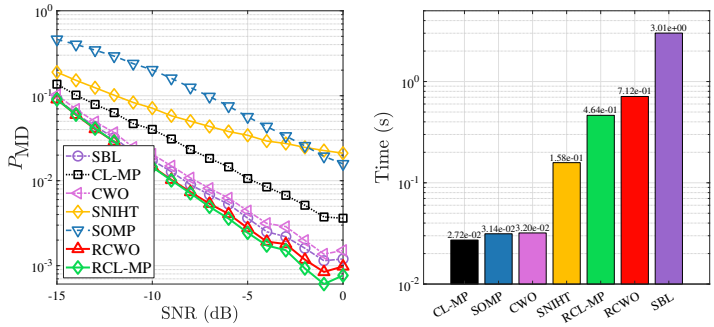


Figure 3.2. *Left*: Probability of miss detection versus SNR. *Right*: Average running time of compared algorithms. The noise is impulsive t-noise with $\nu = 2.5$, and the system configuration is as: $N = 1000$, $K = 25$, $M = 40$ and $L = 40$.

algorithm (Algorithm 2). Note that, in practical executions, the FP algorithm typically converges within $T = 5$ steps. RCWO method can execute for maximum I_{cyc} rounds, while RCL-MP stops after K rounds. In mMTC scenarios, it is common that $K < I_{\text{cyc}}$. Regarding other methods commonly applied in the AD problem, the SBL [71] has complexity of $\mathcal{O}(N^2L)$, the VAMP [43] has complexity of $\mathcal{O}(N^2M)$, and the SOMP [65] has complexity of $\mathcal{O}(L\cdot\max(NM,k^2)+k^3)$, where k represents the index of current iteration. Table 3.1 outlines the complexity of algorithms discussed above.

3.4 Numerical experiment

We assess the proposed RCWO and RCL-MP algorithms' performance within a cellular massive MIMO uplink mMTC network. Within a range of $[0.05\text{ km}, 1\text{ km}]$, $N = 1000$ devices are randomly positioned and served by the BS. The path loss model from [42] is adopted for the LSFCs, given as

$$\beta_i = -128.1 - 36.7\log_{10}(d_i), \tag{3.24}$$

with d_i indicating the distance in kilometers between the BS and the i-th device. The power control scheme from [61] is implemented as follows

$$\varrho_i = \frac{\varrho_{\max}(\beta_i)_{\min}}{\beta_i}, \tag{3.25}$$

with $\varrho_{\max}$ being the maximum transmission power of the devices, and $(\beta_i)_{\min}$ indicating the minimum LSFC. Under this scheme, maximum power transmission occurs for the device with the lowest LSFC, while other devices' transmission powers scale inversely with their LSFCs. The maximum transmission power is configured as $\varrho_{\max} = 0.2$W. With bandwidth and coherence time configured at 1 MHz and 1 ms respectively, 1000 symbols can be transmitted through the wireless channel. Pilot sequences occupy 10% of available symbols, leading to a maximum pilot length of $L_{\max} = 100$.

We adopt the Bernoulli pilot. Given the finite set of unique pilot sequences, identical sequences may occur between two devices with non-zero probability, termed the collision probability [61]. The calculation from [61, Eq. (2)] shows that with $N = 1000$ devices and pilot length $L = 20$, this collision probability reaches 4.54×10^{-7}, remaining negligible in practice.

The performance of the proposed RCWO and RCL-MP algorithms is compared to the following algorithms

1. Coordinatewise Optimization Algorithm (CWO) [14, Algorithm 1]

2. Covariance-based Matching Pursuit (CL-MP) [49]

3. Sparse Bayesian Learning (SBL) [71]

4. Simultaneous Normalized Iterative Hard Thresholding (SNIHT) [2, Algorithm 1]

5. Simultaneous orthogonal matching pursuit (SOMP) [65, Algorithm 3.1]

Note that the SBL algorithm is implemented using the Expectation Maximization (EM) approach as described in [71], rather than the coordinatewise optimization variant.

Each row vector of the noise matrix $\boldsymbol{E} \in \mathbb{C}^{L\times M}$ is generated independently from a circular M-variate t-distribution with $\nu = 2.5$ degrees of freedom and covariance matrix $\sigma^2\boldsymbol{I}_M$. The t-distribution naturally models impulsive noise through its heavier tails compared to Gaussian distributions, with the parameter ν controlling heaviness of the tails (smaller values produce more frequent outliers). For $\nu > 2$, this noise can be generated by scaling an M-variate circular complex Gaussian random vector $\boldsymbol{z} \sim \mathcal{CN}_M(\boldsymbol{0}, \boldsymbol{I})$ by $\sqrt{\sigma^2(\nu-2)/\chi^2_\nu}$, where χ^2_ν represents independent chi-squared random variable with ν degrees of freedom. Our choice of $\nu = 2.5$ represents moderately

heavy-tailed noise that effectively challenges Gaussian-based methods while maintaining finite variance.

Fig. 3.2 (left panel) shows the probability of miss detection (P_{MD}) performance across SNR levels from [-15, 0] dB, while Fig. 3.2 (right panel) shows the average running time of the algorithms. Across all cases, the proposed RCL-MP algorithm delivers the best performance, with the proposed RCWO following closely behind. RCL-MP's running time remains comparable to SNIHT. Despite requiring additional computation, RCL-MP delivers superior performance under non-Gaussian noise while maintaining running time comparable to CWO.

3.5 Conclusion and discussion

This chapter has addressed the critical challenge of device AD in massive IoT systems operating under complex, non-Gaussian noise conditions. Traditional AD algorithms, which often rely on Gaussian noise assumptions, were shown to be susceptible to performance degradation in the presence of heavy-tailed noise and outliers. To overcome this limitation, this chapter introduced a robust statistical framework for AD. The core of this framework is the replacement of the conventional Gaussian negative log-likelihood function with objective functions based on robust loss functions, such as Huber's loss.

A key theoretical contribution was the proof of the geodesic convexity of the conditional robust objective function, which facilitated the development of a provably convergent FP iterative algorithm for its efficient minimization. Building upon this foundation, two novel and practical robust AD algorithms were proposed: the RCWO algorithm and the RCL-MP algorithm.

The importance of these results is most evident when comparing their performance to state-of-the-art methods under challenging conditions. While conventional Gaussian-based AD algorithms like CWO and CL-MP perform well in benign noise, their performance deteriorates in the presence of heavy-tailed noises, as demonstrated in our experimental evaluations. In sharp contrast, the proposed RCWO and RCL-MP algorithms maintain high detection accuracy and reliability across a wide range of non-Gaussian noise scenarios. This demonstrates advancements in robustness over their Gaussian-based counterparts.

Furthermore, this robustness is achieved without compromising practical applicability. The RCL-MP algorithm, in particular, was shown to achieve computational efficiency comparable to the widely-used Gaussian CWO algorithm, making it a viable and powerful alternative for large-scale deployments. The RCWO algorithm also demonstrated superior performance in scenarios with a larger number of active devices. We want to

note that, the proposed framework is also flexible; by tuning the parameters of the robust loss function (e.g., setting the quantile q close to 1 for Huber's loss), our methods can recover performance nearly identical to SOTA Gaussian-based methods when the noise is purely Gaussian.

In the broader context of this thesis, this chapter contributes to the theme of robustness as resilience to data distributional deviations. By moving beyond idealized statistical assumptions and leveraging principles from robust statistics, the developed framework provides a crucial step towards building reliable massive random access systems for real-world IoT applications. Having addressed robustness against both structural perturbations (Chapter 2) and distributional deviations (this Chapter), the thesis will next explore a third perspective of robustness: achieving stable and interpretable models for high-dimensional structured tensor data.

4. Generalized Nonnegative Structured Kruskal Tensor Regression for Effective and Interpretable Data Modeling

The effective modeling of high-dimensional, multi-aspect data, frequently encountered in the form of tensors, presents both significant opportunities and challenges in modern signal processing and machine learning, as outlined in Chapter 1. While tensor representations encapsulate rich structural information and complex interdependencies often lost in vectorized approaches, conventional tensor regression or decomposition methods may struggle to capture inherent structural heterogeneity (such as mode-specific sparsity or smoothness) or enforce critical domain-specific constraints like non-negativity. This can lead to suboptimal model performance and a lack of interpretability, hindering their utility in applications ranging from hyperspectral imaging to biomedical signal analysis. Chapter 1 also introduced the perspective that robustness in such high-dimensional settings can be achieved by incorporating structural prior knowledge, typically implemented via regularization, to guide solutions towards more stable and physically meaningful outcomes.

This chapter directly addresses these challenges by introducing a novel and comprehensive Generalized Nonnegative Structured Kruskal Tensor Regression (NS-KTR) framework. The primary objective is to develop a versatile tensor regression methodology that not only enhances predictive performance but also promotes model interpretability. This is achieved by systematically integrating non-negativity constraints with mode-specific hybrid regularization strategies—tailoring fused LASSO, total variation, and ridge regularizers to the distinct structural characteristics of individual tensor modes. Furthermore, the NS-KTR framework is designed to accommodate both linear and logistic regression formulations, making it applicable to diverse response variable types, and is solved using an efficient parameter estimation algorithm based on the Alternating Direction Method of Multipliers (ADMM). The core methodologies and findings presented herein are principally derived from Publication VI, which details the complete NS-KTR framework and its extensive validation. Publication VI represents generalization and advancement over the preliminary investigation into nonnegative structured KTR presented in Publication V.

The main contributions of this chapter are as follows:

- **Introduction of the NS-KTR approach.** A novel NS-KTR model is introduced, which uniquely integrates non-negativity constraints with mode-specific hybrid regularization. This allows the method to adaptively promote sparsity, piecewise constancy, and smoothness, tailored to the inherent structural characteristics of each dimension of the tensor parameter.

- **Extension to generalized regression models.** The developed NS-KTR framework is versatile, extending beyond standard Gaussian response assumptions by incorporating both linear and logistic tensor regression variants. This enables the effective modeling of continuous and binary response variables through a unified computational approach, enhancing its applicability to a broader range of real-world datasets.

- **Development of an efficient ADMM-based optimization algorithm.** An efficient alternating optimization algorithm, leveraging the ADMM, is developed for parameter estimation in the NS-KTR framework. This algorithm is designed to handle the non-convex objective function, non-negativity constraints, and structured regularizations effectively, demonstrating superior performance and computational tractability on both synthetic and real-world hyperspectral imaging (HSI) datasets.

These contributions collectively establish the NS-KTR framework as a powerful tool for robust and interpretable tensor regression. By enforcing structural priors (non-negativity and mode-specific regularizations), the framework leads to more stable estimations from potentially noisy or complex high-dimensional data, aligning with the pursuit of *robustness through structural priors and regularization* as discussed in Chapter 1.

4.1 Background of tensor regression and motivation

TR models have gained significant attention over the past decade as a powerful tool for relating high-dimensional tensor covariates to a response variable. A variety of approaches have been proposed in the literature, distinguished by the type of low-rank decomposition assumed for the parameter tensor. These include models based on Tucker decomposition, which offers great flexibility through a core tensor and factor matrices [40], and models based on low-rank orthogonally decomposable tensors [58]. For handling more complex tensor structures, methods like tensor ring completion [45] have been developed. Furthermore, the TR framework has been extended to incorporate graphical models in graph-regularized

tensor regression [76] and integrated with deep learning concepts in tensor regression networks [36]. Among these, models based on the Kruskal tensor, i.e., the CPD [37, 5, 26], are particularly prevalent due to their simplicity and interpretability. The Bayesian KTR model, for instance, provides a probabilistic framework for estimation [21]. However, despite this rich landscape of methods, several fundamental challenges remain, motivating the work presented in this chapter.

While the classical KTR model effectively reduces the number of unknown parameters via a rank-R CPD, it often falls short in handling the complex characteristics of real-world tensor data, such as that from HSI or brain connectivity networks. Such data frequently exhibits multi-dimensional structural heterogeneity—for example, smoothness along one mode, sparsity in another, or piecewise constancy in a third—which demands more than just a low-rank assumption. A targeted, mode-specific regularization is required to preserve these vital structural characteristics.

Furthermore, two additional limitations motivate the development of a more generalized framework. First, many applications, particularly in HSI or other physical measurement domains, involve inherently non-negative data. Enforcing non-negativity constraints on the parameter tensor is not only physically meaningful but has also been shown to improve optimization stability and ensure the existence of global solutions for CPD approximation problems [41]. Second, standard TR models often assume Gaussian responses and are optimized via Frobenius norm minimization. This becomes problematic for binary or categorical labels common in classification tasks. These limitations motivate the development of a more structured and generalized framework, as proposed in this chapter.

4.2 Preliminaries on tensor

We use $\mathscr{B} = (b_{i_1 \cdots i_D})$ to denote a D-way tensor of size $I_1 \times \cdots \times I_D$. The mode-$d$ matricization [35] of $\mathscr{B}$ is defined as a matrix $\boldsymbol{B}_{(d)} \in \mathbb{R}^{I_d \times \prod_{d' \neq d} I_{d'}}$, which reshapes tensor $\mathscr{B}$ to a $I_d \times \prod_{d' \neq d} I_{d'}$ matrix such that the tensor's $(i_1, \ldots, i_D)$ element corresponds to the (i_d, j) element of the matrix $\boldsymbol{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d} (i_{d'} - 1) \prod_{d'' < d', d'' \neq d} I_{d''}$. The vectorization operator $\mathrm{vec}(\cdot)$ maps a tensor into a vector by stacking the columns of $\boldsymbol{B}_{(1)}$ on top of each other. The tensor inner product between two tensors of same size are also denoted by the inner product of their vectorized or mode-d matricized counterparts as $\langle \mathscr{A}, \mathscr{B} \rangle = \langle \mathrm{vec}(\mathscr{A}), \mathrm{vec}(\mathscr{B}) \rangle = \langle \boldsymbol{A}_{(d)}, \boldsymbol{B}_{(d)} \rangle$, where the latter inner product for matrices can also be expressed compactly using matrix trace as $\langle \boldsymbol{A}_{(d)}, \boldsymbol{B}_{(d)} \rangle = \mathrm{tr}(\boldsymbol{A}_{(d)} \boldsymbol{B}_{(d)}^\top)$.

The rank-R CPD expresses a tensor as a linear combination of rank-1

tensors

$$\mathscr{B} \equiv [\![\boldsymbol{B}_1,\ldots,\boldsymbol{B}_D]\!] = \sum_{r=1}^{R} \boldsymbol{B}_1(\cdot,r)\circ\cdots\circ\boldsymbol{B}_D(\cdot,r), \tag{4.1}$$

where $\boldsymbol{B}_d \in \mathbb{R}^{I_d\times R}$ for $d = 1,\ldots,D$ are latent factor matrices, $\boldsymbol{B}_d(\cdot,r) \in \mathbb{R}^{I_d}$ denotes the r-th column of matrix, and $\circ$ denotes the outer product. A tensor admitting decomposition (4.1) is also referred to as a Kruskal tensor [35].

The Tucker decomposition, also known as higher-order SVD, is a highly flexible model that decomposes a tensor into a dense core tensor multiplied by a factor matrix along each mode. For a D-way tensor $\mathscr{B} \in \mathbb{R}^{I_1\times\cdots\times I_D}$, its Tucker decomposition is given by

$$\mathscr{B} = \mathscr{G} \times_1 \boldsymbol{B}^{(1)} \times_2 \boldsymbol{B}^{(2)} \cdots \times_D \boldsymbol{B}^{(D)},$$

where $\mathscr{G} \in \mathbb{R}^{R_1\times\cdots\times R_D}$ is the core tensor, and $\boldsymbol{B}^{(d)} \in \mathbb{R}^{I_d\times R_d}$ is the factor matrix for the d-th mode. The term $\times_d$ denotes the mode-d product of a tensor and a matrix. The dimensions of the core tensor, $(R_1,\ldots,R_D)$, are referred to as the Tucker ranks. The CPD can be viewed as a special case of the Tucker decomposition where the core tensor is superdiagonal. For very high-order tensors, models like the Tensor Train (TT) [56] and Tensor Ring [80] decompositions have gained prominence. These models represent a high-dimensional tensor as a sequence or ring of interconnected lower-dimensional core tensors, mitigating the curse of dimensionality. The selection of a particular decomposition model often depends on the specific structural assumptions and computational constraints of the application. A comprehensive and detailed introduction to tensor decompositions and their applications can be found in [35].

Consider two matrices $\boldsymbol{A} = (\boldsymbol{a}_1 \cdots \boldsymbol{a}_n) \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} = (\boldsymbol{b}_1 \cdots \boldsymbol{b}_q) \in \mathbb{R}^{p\times q}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same number of columns $n = q$, then the Khatri-Rao product is defined as a columnwise Kronecker product

$$\boldsymbol{A}\odot\boldsymbol{B} = (\boldsymbol{a}_1\otimes\boldsymbol{b}_1\ \boldsymbol{a}_2\otimes\boldsymbol{b}_2\ \ldots\ \boldsymbol{a}_n\otimes\boldsymbol{b}_n), \tag{4.2}$$

where $\otimes$ denotes the Kronecker product. If $\mathscr{B} \in \mathbb{R}^{I_1\times\cdots\times I_D}$ is a rank-R Kruskal tensor (4.1), then the inner-product between $\mathscr{B}$ and tensor $\mathscr{X} \in \mathbb{R}^{I_1\times\cdots\times I_D}$ can be found as

$$\begin{aligned}\langle\mathscr{X},\mathscr{B}\rangle &= \langle\boldsymbol{X}_{(d)}\boldsymbol{B}_{(-d)},\boldsymbol{B}_d\rangle = \operatorname{tr}((\boldsymbol{X}_{(d)}\boldsymbol{B}_{(-d)})^\top\boldsymbol{B}_d)\\ &= \operatorname{vec}(\boldsymbol{X}_{(d)}\boldsymbol{B}_{(-d)})^\top\operatorname{vec}(\boldsymbol{B}_d),\end{aligned} \tag{4.3}$$

where $\boldsymbol{B}_{(-d)}$ is a Khatri-Rao product between all d'-th factor matrices $(d' = 1,\ldots,D, d' \neq d)$ of $\mathscr{B}$

$$\boldsymbol{B}_{(-d)} = \boldsymbol{B}_D\odot\cdots\odot\boldsymbol{B}_{d+1}\odot\boldsymbol{B}_{d-1}\odot\cdots\odot\boldsymbol{B}_1, \tag{4.4}$$

and $\boldsymbol{X}_{(d)}\boldsymbol{B}_{(-d)}$ is the mode-d matricized tensor times Khatri-Rao product (MTTKRP), which is the key computational element in CP decomposition [35].

We use

$$\boldsymbol{\beta}_d = \text{vec}(\boldsymbol{B}_d) \in \mathbb{R}^{I_d R} \tag{4.5}$$

to denote the vectorized form of the dth factor matrix, with the inverse operation

$$\boldsymbol{B}_d = \text{unvec}(\boldsymbol{\beta}_d) \in \mathbb{R}^{I_d \times R}. \tag{4.6}$$

Additionally, we use $\boldsymbol{\beta}_{dr} = \boldsymbol{B}_d(\cdot, r) \in \mathbb{R}^{I_d}$ to denote the rth column of the dth factor matrix, and thus, the vectorized $\boldsymbol{\beta}_d$ concatenates all columns of $\boldsymbol{B}_d$.

In tensor regression, a critical challenge is the exponential growth of parameters. For a coefficient tensor $\mathscr{B}$ of size $I_1 \times \cdots \times I_D$, the total number of parameters is $\prod_{d=1}^{D} I_d$, which quickly exceeds the available sample size N as dimensions increase. By assuming a low-rank structure for $\mathscr{B}$ through a rank-R CPD, we reduce the number of parameters to $R\sum_{d=1}^{D} I_d$, substantially mitigating overfitting risk while preserving computational efficiency. The CPD of tensor $\mathscr{B}$ with rank-R, where $R \in \mathbb{N}_0^+$, allows us to separate the d-th factor $\boldsymbol{B}_d$ from the tensor inner product

$$\langle \mathscr{X}_i, \mathscr{B} \rangle = \langle \mathscr{X}_i, [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!] \rangle = \text{vec}(\boldsymbol{X}_{i(d)}\boldsymbol{B}_{(-d)})^\top \boldsymbol{\beta}_d, \tag{4.7}$$

where $\boldsymbol{\beta}_d = \text{vec}(\boldsymbol{B}_d) \in \mathbb{R}^{I_d R}$ is the vectorized d-th factor matrix.

4.3 Proposed method

4.3.1 Structured regularization

Real-world tensor data often exhibit distinctive structural characteristics across their various dimensions. For instance, in HSI, the spectral dimension typically shows smooth variations due to the continuous nature of spectral signatures, while the spatial dimensions are characterized by locally constant regions with meaningful boundaries, reflecting spatial coherence. To effectively capture these multi-dimensional structures, we propose a hybrid regularization framework that extends the Fused LASSO regularization [64] by incorporating non-negativity constraints, explicitly tailoring regularization to the multi-dimensional structural properties of each mode d.

Our overall regularization function $h(\cdot)$ for the joint optimization problem is composed of separate mode-specific regularizers:

$$h(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_D) = \sum_{d=1}^{D} h_d(\boldsymbol{\beta}_d), \tag{4.8}$$

where each $h_d(\cdot)$ acts independently on its corresponding factor vector $\boldsymbol{\beta}_d$. For each mode d, we define a general form of regularization function as

$$h_d(\boldsymbol{\beta}_d) \triangleq \lambda_{d1}\|\boldsymbol{\beta}_d\|_1 + \lambda_{d2}\|\boldsymbol{D}_d\boldsymbol{\beta}_d\|_1 + \frac{\lambda_{d3}}{2}\|\boldsymbol{\beta}_d\|_2^2 + \iota_{\geq 0}(\boldsymbol{\beta}_d), \tag{4.9}$$

where $\boldsymbol{\beta}_d = \text{vec}(\boldsymbol{B}_d) \in \mathbb{R}^{I_d R}$ is the vectorized factor matrix, and (λ_{d1}, λ_{d2}, λ_{d3}) are non-negative regularization parameters that control different structural properties, and $\boldsymbol{D}_d \in \mathbb{R}^{(I_d-1)R \times I_d R}$ is the first-order difference matrix defined as

$$\boldsymbol{D}_d \triangleq \boldsymbol{I}_R \otimes \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ & & & \vdots & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}_{(I_d-1)\times I_d}. \tag{4.10}$$

Each component of (4.9) serves a distinct purpose. *(i)* The ℓ_1 norm term $\|\boldsymbol{\beta}_d\|_1$ (also called LASSO [63]) promotes sparsity, effectively performing feature selection within each mode. *(ii)* The total variation term $\|\boldsymbol{D}_d\boldsymbol{\beta}_d\|_1$ encourages piecewise constant patterns, preserving sharp transitions between regions, as introduced in [59]. *(iii)* The quadratic term $\|\boldsymbol{\beta}_d\|_2^2$ induces smoothness, mitigating noise and ensuring stability in the estimation [46]. *(iv)* The indicator function $\iota_{\geq 0}(\boldsymbol{\beta}_d)$ enforces non-negativity constraint, which is critical for applications like HSI where physical quantities (e.g., reflectance) are inherently non-negative.

By strategically configuring the regularization parameters ($\lambda_{d1}, \lambda_{d2}, \lambda_{d3}$) and selectively enforcing the non-negativity constraint $\iota_{\geq 0}$, our proposed hybrid framework enables flexible, mode-specific regularization tailored to the structural characteristics of each tensor mode. Specifically, several well-established regularization terms can be replicated as follows.

1. LASSO (Sparsity): When $\lambda_{d1} > 0$, $\lambda_{d2} = 0$, $\lambda_{d3} = 0$, the penalty reduces to the LASSO, promoting sparsity in the factor matrix. With the non-negativity constraint $\iota_{\geq 0}(\cdot)$, this becomes nonnegative LASSO, ensuring sparse, non-negative factors.

2. Total Variation (TV): When $\lambda_{d1} = 0$, $\lambda_{d2} > 0$, $\lambda_{d3} = 0$, the penalty corresponds to TV regularization, encouraging piecewise constant solutions.

3. Fused LASSO (Sparsity + Piecewise Constant) [64]: When $\lambda_{d1} > 0$, $\lambda_{d2} > 0$, $\lambda_{d3} = 0$, the penalty becomes the fused LASSO, simultaneously enforcing sparsity and piecewise constant behavior.

4. Elastic Net (Sparsity + Smoothness) [82]: When $\lambda_{d1} > 0$, $\lambda_{d2} = 0$, $\lambda_{d3} > 0$, the penalty aligns with the Elastic Net, balancing sparsity and smoothness.

Each factor matrix preserves distinct structural characteristics while maintaining non-negativity constraints, capturing the heterogeneous patterns across different tensor dimensions.

By integrating these mode-specific penalties into the NS-KTR framework, we can adapt the regularization to the unique characteristics of each tensor dimension. For example, in HSI, the spectral mode may prioritize smoothness (higher λ_{d3}), while the spatial modes emphasize piecewise constant behavior (higher λ_{d2}). This adaptability enhances both the interpretability of the factor matrices and the predictive performance of the model, making it particularly suitable for applications processing tensor data with heterogeneous structural properties across dimensions.

4.3.2 Alternating optimization framework

After establishing the structured regularization approach, we now formulate the complete Nonnegative Structured Kruskal Tensor Regression (NS-KTR) optimization problem:

$$\min_{\{\boldsymbol{\beta}_d\}_{d=1}^D} f(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_D)+h(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_D), \tag{4.11}$$

where $f(\cdot)$ represents the data fidelity term that varies based on the regression model. For linear regression, this term takes the form

$$f(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_D)=\frac{1}{2}\sum_{i=1}^{N}(y_i-\langle\mathcal{X}_i,[\![\boldsymbol{B}_1,\ldots,\boldsymbol{B}_D]\!]\rangle)^2, \tag{4.12}$$

while for logistic regression, it becomes

$$f(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_D)=\sum_{i=1}^{N}\log(1+\exp(-y_i\langle\mathcal{X}_i,[\![\boldsymbol{B}_1,\ldots,\boldsymbol{B}_D]\!]\rangle)). \tag{4.13}$$

This optimization problem (4.11) presents two key challenges: it is nonconvex with respect to all factor vectors jointly, and the regularization terms along with non-negativity constraints add further complexity. However, to address it, we can exploit the problem's structure. Particularly, when all but one factor vector are fixed, the problem becomes convex with respect to the remaining factor vectors. We therefore can build an alternating optimization strategy, updating each factor vector sequentially while keeping others fixed. At the $(t+1)$th iteration, we update the dth factor vector as

$$\boldsymbol{\beta}_d^{t+1}=\operatorname*{argmin}_{\boldsymbol{\beta}_d} f(\boldsymbol{\beta}_1^{t+1},\ldots,\boldsymbol{\beta}_{d-1}^{t+1},\boldsymbol{\beta}_d,\boldsymbol{\beta}_{d+1}^{t},\ldots,\boldsymbol{\beta}_D^{t})+h_d(\boldsymbol{\beta}_d). \tag{4.14}$$

Since each mode-specific subproblem in (4.14) has the same form, we can use compact notation. Define $\boldsymbol{x}=\boldsymbol{\beta}_d$ for the current mode $d\in\{1,\ldots,D\}$, so

Algorithm 4 NS-KTR: Nonnegative structured Kruskal tensor regression

Input: Response $\boldsymbol{y}$, tensor covariates $\mathcal{X}_i, \forall i$, rank R.
Initialization: $\forall d = 1,\ldots,D$, Randomize or warm start $\boldsymbol{B}_d^0$,
for $t = 0,1,\ldots,t_{\text{iter}}$ **do**
  $\boldsymbol{B}_{(-d)}^{t+1} \leftarrow \boldsymbol{B}_D^t \odot \cdots \odot \boldsymbol{B}_{d+1}^t \odot \boldsymbol{B}_{d-1}^t \odot \cdots \odot \boldsymbol{B}_1^t, \forall d \in \{1,\ldots,D\}$
  for $d = 1,\ldots,D$ **do**
    $\boldsymbol{A} \leftarrow \left(\text{vec}(\boldsymbol{X}_{1(d)}\boldsymbol{B}_{(-d)}^{t+1}) \quad \cdots \quad \text{vec}(\boldsymbol{X}_{N(d)}\boldsymbol{B}_{(-d)}^{t+1})\right)^\top$
    $\boldsymbol{\beta}_d^{t+1} \leftarrow \boldsymbol{x}^{t+1}$ by solving (4.15) with ADMM (Algorithm 5)
    $\boldsymbol{B}_d^{t+1} \leftarrow \text{unvec}(\boldsymbol{\beta}_d^{t+1})$
  end for
end for
Return: $(\boldsymbol{B}_1^{t+1},\ldots,\boldsymbol{B}_D^{t+1})$

each subproblem becomes

$$\boldsymbol{x} = \underset{\boldsymbol{x}}{\text{argmin}}\, g(\boldsymbol{x}) + h_d(\boldsymbol{x}), \tag{4.15}$$

where $g(\cdot)$ takes either the linear regression form

$$g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \tag{4.16}$$

or the logistic regression form

$$g(\boldsymbol{x}) = \sum_{i=1}^{N} \log(1 + \exp(-y_i(\boldsymbol{A}\boldsymbol{x})_i). \tag{4.17}$$

Here, matrix $\boldsymbol{A} \in \mathbb{R}^{N \times I_d R}$ represents the concatenation of vectorized MTTKRP operations across all samples

$$\boldsymbol{A} = \left(\text{vec}(\boldsymbol{X}_{1(d)}\boldsymbol{B}_{(-d)}) \quad \cdots \quad \text{vec}(\boldsymbol{X}_{N(d)}\boldsymbol{B}_{(-d)})\right)^\top, \tag{4.18}$$

and $\boldsymbol{a}_i = \text{vec}(\boldsymbol{X}_{i(d)}\boldsymbol{B}_{(-d)}) \in \mathbb{R}^{I_d R}$ denotes the vectorized MTTKRP for the ith sample.

The complete alternating optimization procedure is outlined in Algorithm 4. This approach effectively decomposes the original high-dimensional non-convex problem into a sequence of manageable convex subproblems. However, each subproblem still involves complex regularization terms and constraints that require specialized solvers.

4.4 ADMM algorithm for subproblem

To efficiently solve each subproblem in our alternating optimization framework, we develop an ADMM approach. ADMM is well-suited for our

Algorithm 5 ADMM for subproblem

Input: Response $\boldsymbol{y}$, matrix $\boldsymbol{A}$ and $\boldsymbol{D}_d$, hyper-parameters $(\lambda_1, \lambda_2, \lambda_3)$.
Initialize: Randomize $(\boldsymbol{x}^0, \boldsymbol{z}_1^0, \boldsymbol{z}_2^0, \boldsymbol{u}_1^0, \boldsymbol{u}_2^0)$, $\rho > 0$, $\epsilon = 10^{-5}$
for $k = 0, 1, 2, \ldots$ **do**
  if Linear regression **then**
    $\boldsymbol{M} \leftarrow \boldsymbol{A}^\top \boldsymbol{A} + (\rho + \lambda_3)\boldsymbol{I} + \rho \boldsymbol{D}_d^\top \boldsymbol{D}_d$ ▷ Compute only once
    $\boldsymbol{b}^k \leftarrow \boldsymbol{A}^\top \boldsymbol{y} + \rho(\boldsymbol{z}_1^k - \boldsymbol{u}_1^k/\rho) + \rho \boldsymbol{D}_d^\top (\boldsymbol{z}_2^k - \boldsymbol{u}_2^k/\rho)$
    $\boldsymbol{x}^{k+1} \leftarrow \left[\boldsymbol{M}^{-1}\boldsymbol{b}^k\right]_+$
  else if Logistic regression **then**
    Solve $\boldsymbol{x}^{k+1}$ using the Newton's method in [34]
  end if
  $\boldsymbol{z}_1^{k+1} \leftarrow \operatorname{prox}_{\frac{\lambda_1}{\rho}\|\cdot\|_1}\left(\boldsymbol{x}^{k+1} + \frac{1}{\rho}\boldsymbol{u}_1^k\right)$
  $\boldsymbol{z}_2^{k+1} \leftarrow \operatorname{prox}_{\frac{\lambda_2}{\rho}\|\cdot\|_1}\left(\boldsymbol{D}_d\boldsymbol{x}^{k+1} + \frac{1}{\rho}\boldsymbol{u}_2^k\right)$
  $\boldsymbol{u}_1^{k+1} \leftarrow \boldsymbol{u}_1^k + \rho(\boldsymbol{x}^{k+1} - \boldsymbol{z}_1^{k+1})$
  $\boldsymbol{u}_2^{k+1} \leftarrow \boldsymbol{u}_2^k + \rho(\boldsymbol{D}_d\boldsymbol{x}^{k+1} - \boldsymbol{z}_2^{k+1})$
  $\boldsymbol{r}_1^{k+1} \leftarrow \boldsymbol{x}^{k+1} - \boldsymbol{z}_1^{k+1}$, $\boldsymbol{r}_2^{k+1} \leftarrow \boldsymbol{D}_d\boldsymbol{x}^{k+1} - \boldsymbol{z}_2^{k+1}$, $\boldsymbol{s}_1^{k+1} \leftarrow \rho(\boldsymbol{z}_1^{k+1} - \boldsymbol{z}_1^k)$, $\boldsymbol{s}_2^{k+1} \leftarrow \rho(\boldsymbol{z}_2^{k+1} - \boldsymbol{z}_2^k)$
  If $\max\left(\|\boldsymbol{r}_1^{k+1}\|_2, \|\boldsymbol{r}_2^{k+1}\|_2, \|\boldsymbol{s}_1^{k+1}\|_2, \|\boldsymbol{s}_2^{k+1}\|_2\right) \le \epsilon$ **then** break
end for

scenario as it effectively handles the composite structure of our objective with multiple regularization terms and constraints.

We start by reformulating the d-th subproblem (4.15) by introducing auxiliary variables that separate the objective into more manageable components as

$$\min_{\boldsymbol{x}} g(\boldsymbol{x}) + \lambda_1\|\boldsymbol{z}_1\|_1 + \lambda_2\|\boldsymbol{z}_2\|_1 + \frac{\lambda_3}{2}\|\boldsymbol{x}\|_2^2 \tag{4.19a}$$

$$\text{s.t. } \boldsymbol{x} \ge 0, \tag{4.19b}$$

$$\boldsymbol{x} - \boldsymbol{z}_1 = 0, \tag{4.19c}$$

$$\boldsymbol{D}_d\boldsymbol{x} - \boldsymbol{z}_2 = 0, \tag{4.19d}$$

where $\boldsymbol{z}_1 \in \mathbb{R}^{I_d R}$ and $\boldsymbol{z}_2 \in \mathbb{R}^{(I_d - 1)R}$ are auxiliary variables that decouple the non-differentiable regularization terms from the data fidelity term.

The augmented Lagrangian for problem (4.19) is then given as

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{u}_1, \boldsymbol{u}_2) &= g(\boldsymbol{x}) + \lambda_1\|\boldsymbol{z}_1\|_1 + \lambda_2\|\boldsymbol{z}_2\|_1 \\ &+ \boldsymbol{u}_1^T(\boldsymbol{x} - \boldsymbol{z}_1) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}_1\|_2^2 + \boldsymbol{u}_2^T(\boldsymbol{D}_d\boldsymbol{x} - \boldsymbol{z}_2) \\ &+ \frac{\rho}{2}\|\boldsymbol{D}_d\boldsymbol{x} - \boldsymbol{z}_2\|_2^2 + \frac{\lambda_3}{2}\|\boldsymbol{x}\|_2^2 + \iota_{\ge 0}(\boldsymbol{x}), \end{aligned} \tag{4.20}$$

where $\boldsymbol{u}_1 \in \mathbb{R}^{I_d R}$ and $\boldsymbol{u}_2 \in \mathbb{R}^{(I_d - 1)R}$ are dual variables, and $\rho > 0$ is the penalty parameter that controls the balance between primal feasibility and dual

progress. The complete ADMM procedure for solving (4.19) is presented in Algorithm 5.

4.5 Conclusion and discussion

This chapter introduced the Generalized NS-KTR framework, a novel methodology for the effective modeling of high-dimensional, multi-aspect data. The core of the NS-KTR framework lies in its ability to integrate critical, often domain-specific, structural priors into the tensor regression problem. This was achieved through the systematic combination of a CPD with non-negativity constraints and mode-specific hybrid regularizations, including LASSO, total variation, and ridge penalties. Furthermore, the framework was designed to be versatile, accommodating both linear and logistic regression formulations, and was equipped with an efficient ADMM-based algorithm for scalable parameter estimation.

The contributions presented in this chapter provide advancements over conventional tensor regression methods. By moving beyond a simple low-rank assumption, NS-KTR directly addresses the challenge of structural heterogeneity often present in real-world tensor data. The ability to impose regularization to each tensor mode allows the model to preserve distinct structural characteristics, such as spectral smoothness and spatial piecewise constancy in hyperspectral images, which are often lost in unstructured approaches. The enforcement of non-negativity constraints further enhances model interpretability and ensures physically meaningful solutions.

The implications of this work extend to a wider setting beyond the HSI primarily used for validation in Publication VI. The principles of mode-specific regularization and non-negativity are broadly applicable to any domain where tensor data exhibits diverse structural properties across its dimensions. For example, in neuroscience, this could involve modeling fMRI or EEG data where temporal modes might be smooth while spatial modes are sparse. Furthermore, the iterative nature of the ADMM-based algorithm developed for NS-KTR opens up possibilities for algorithmic advancements through deep unrolling [6]. By unrolling the iterative steps of the NS-KTR solver into layers of a deep neural network, it is possible to create a new, model-based deep learning architecture. In such a framework, critical components of the original algorithm, such as the regularization parameters or even the forms of the proximal operators, could be learned directly from data through end-to-end training. This approach has the potential to achieve better estimation results by allowing the model to automatically adapt its structural priors to the specific characteristics of the training dataset, moving beyond hand-tuned or grid-searched hyperparameters. This integration of classical model-based optimization

with data-driven deep learning represents a promising direction for future research in robust and interpretable tensor modeling.

In the context of this thesis's overarching theme, this chapter addresses robustness through the lens of structural priors and regularization. By constraining the solution space with well-chosen, domain-aware priors, the NS-KTR model becomes inherently less sensitive to noise and minor variations in the training data. Having now explored robustness from three distinct yet complementary perspectives, output stability under structural perturbations (Chapter 2), resilience to distributional deviations (Chapter 3), and stability through structural priors (this Chapter), the final chapter will summarize these findings, discuss their collective implications, and outline directions for future research.

5. Conclusion

This doctoral research has focused on advancing robust signal processing and machine learning methodologies, addressing critical challenges posed by data irregularities and model uncertainties prevalent in contemporary technological systems. The thesis has investigated and developed novel techniques across three core research pillars: ensuring the stability of graph-based learning models, achieving robust device activity detection in complex IoT environments, and enabling effective and interpretable modeling of high-dimensional structured tensor data.

Specifically, for *GCNN under perturbation*, a comprehensive analytical framework was established (Chapter 2, based on Publications I & II) to quantify their sensitivity to probabilistic structural perturbations. We introduced tight expected GSO error bounds linked to error model parameters and revealed a linear relationship governing error propagation, thereby providing crucial theoretical underpinnings for GCNN stability and offering insights for designing more resilient graph learning architectures.

In the domain of *device AD for massive IoT systems* (Chapter 3, based on Publications III & IV), robust statistical frameworks were developed to counteract the detrimental effects of complex non-Gaussian noise. By formulating novel AD objective functions with robust loss principles and deriving efficient algorithms (RCWO and RCL-MP) with proven convergence, we enhanced AD accuracy and reliability in IoT scenarios, outperforming traditional Gaussian-based approaches.

Finally, for *modeling high-dimensional structured tensor data* (Chapter 4, based on Publications V & VI), a generalized NS-KTR framework was introduced. This framework uniquely integrates non-negativity constraints with mode-specific hybrid regularizations, accommodates both linear and logistic regression, and employs an efficient ADMM-based algorithm for parameter estimation. The NS-KTR approach demonstrated superior performance in capturing structural heterogeneity and ensuring interpretability on both synthetic and real-world hyperspectral imaging datasets.

Collectively, these contributions advance the state-of-the-art across diverse signal processing and learning tasks by providing principled and practically effective solutions for enhancing robustness, whether viewed as output stability under structural changes, resilience to data distributional deviations, or stability achieved through structural priors and regularization.

The methodologies developed in this thesis have implications for both theoretical understanding and practical applications. The GCNN stability analysis provides a foundation for building more trustworthy graph learning systems. The robust AD techniques offer a pathway to more reliable and efficient massive IoT deployments, crucial for the next generation of wireless communications. The NS-KTR framework furnishes a powerful and flexible tool for extracting meaningful insights from complex, high-dimensional tensor data, with applications in areas like remote sensing, biomedical signal processing, and beyond. This research contributes to the overarching goal of creating signal processing and machine learning systems that are more dependable and effective in the face of real-world complexities.

5.1 Future research directions

Several avenues remain for future exploration of the proposed methods.

- For GCNN Stability Analysis:
  - Future work could extend the sensitivity analysis to other types of graph perturbations, such as adversarial attacks specifically designed to degrade GCNN performance, or to dynamic graphs where the structure evolves continuously over time.
  - Generalizing the sensitivity analysis framework to broader GNN architectures. While this thesis focused on GCNNs, the developed analytical framework for sensitivity and stability is not inherently limited to this specific architecture. A future direction is to extend this framework to analyze the robustness of more advanced models. For instance, in Graph Attention Networks (GATs) [68], structural perturbations could affect not only the neighborhood definition but also the learned, data-dependent attention weights. Adapting the principles of GSO error bounding and propagation analysis to such dynamic interaction mechanisms can provide stability guarantees for more GNN models, offering possible insights for their reliable deployment.
- For Robust Device AD:

- Extending robust AD to decentralized and non-terrestrial network architectures. The principles of robust AD developed in this thesis can be extended to future communication paradigms. In cell-free massive MIMO systems, for instance, adapting the framework to a distributed setting presents key challenges in developing decentralized estimation algorithms that can handle spatially inhomogeneous non-Gaussian noise across many access points. Another promising direction is direct-to-satellite IoT access, where massive numbers of low-power devices transmit over long-distance, low-SNR channels. Enhancing the proposed robust AD methods to be resilient against the unique atmospheric interferences and Doppler effects inherent in satellite links would be crucial for enabling reliable, truly global IoT connectivity.

- Enhancing robustness through adaptive and learning-based approaches. While this thesis established robust AD frameworks, adaptive selection of robust loss function parameters presents a promising direction. Rather than using fixed quantile q for Huber's loss, algorithms could dynamically estimate noise characteristics and adaptively tune parameters for optimal performance. Learning-based methods, with sufficient IoT training data, could learn optimal robust loss functions or end-to-end robust AD models. Finally, while algorithm convergence was established, deriving rigorous theoretical guarantees for activity detection performance (e.g., phase transition analysis) under specific non-Gaussian noise models remains open.

- While the proposed algorithms demonstrate strong performance in simulations, their implementation on resource-constrained IoT devices with strict latency and power requirements remains a key challenge. Future work could focus on algorithmic optimizations, such as developing lightweight or hardware-aware versions of the algorithms, to ensure their feasibility in real-world mMTC systems.

• For Structured Tensor Regression:

- Extending the NS-KTR framework to other tensor decomposition formats (e.g., Tucker, Tensor Train) or incorporating more sophisticated, data-driven structural priors could further enhance its modeling capabilities.

- Applying structured tensor decomposition for large-scale model compression. The TR framework in this thesis is fundamentally a parameter compression technique. This principle has potential in large-scale neural network compression. Modern architectures like the Transformer [67] contain fully-connected layers whose large weight matrices

can be reshaped into higher-order tensors and compressed. A promising future direction is to apply low-rank approximation to these weight tensors. This would allow for enforcing hardware-friendly structures like sparsity or block-sparsity, thereby substantially reducing the model's memory footprint and computational cost, and facilitating the deployment of powerful models on resource-constrained devices.

– The current ADMM-based algorithm, while efficient, may face computational challenges when applied to extremely large-scale tensor data. A crucial future direction is to explore more scalable optimization techniques, such as stochastic or distributed algorithms, to enhance the framework's applicability to massive real-world datasets.

– While this work focused on comparisons with classical tensor methods, we acknowledge that deep learning architectures represent the state of the art in predictive performance for many complex tasks. A key advantage and differentiator of the NS-KTR framework, however, lies in its interpretability. By explicitly incorporating domain-aware structural priors, our model yields factors that can provide direct scientific insights into the data, a feature often absent in end-to-end deep learning models. A promising future direction is the synthesis of both paradigms, for instance, through deep unrolling of the ADMM algorithm, to combine the predictive power of deep learning with the interpretability of structured models.

In conclusion, these future research directions highlight a trajectory towards more intelligent, adaptive, and resilient signal processing systems. The avenues spanning complex communication scenarios, advanced machine learning integration, and strengthened theoretical foundations underscore the continued relevance of robust signal processing. This thesis provides a foundation for future exploration by systematically addressing robustness through structural stability, distributional resilience, and model-based priors, contributing methodologies for next-generation communication and learning systems operating reliably under real-world uncertainty.

References

[1] Terje Aven. Upper (lower) bounds on the mean of the maximum (minimum) of a number of random variables. *J. Appl. Probab.*, 22(3):723–728, Sept. 1985.

[2] Jeffrey D Blanchard, Michael Cermak, David Hanle, and Yirong Jing. Greedy algorithms for joint sparse recovery. *IEEE Trans. Signal Process.*, 62(7):1694–1704, Apr. 2014.

[3] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.*, 27(3):265–274, Nov. 2009.

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.*, 34(4):18–42, July 2017.

[5] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika*, 35:283–319, Sept. 1970.

[6] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *J. Mach. Learn. Res.*, 23(189):1–59, 2022.

[7] Zhilin Chen, Foad Sohrabi, Ya-Feng Liu, and Wei Yu. Covariance based joint activity and data detection for massive random access with massive MIMO. In *Proc. IEEE Int. Conf. Commun.*, pages 1–6, Shanghai, China, May 20–24 2019.

[8] Zhilin Chen, Foad Sohrabi, and Wei Yu. Sparse activity detection for massive connectivity. *IEEE Trans. Signal Process.*, 66(7):1890–1904, Apr. 2018.

[9] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.*, 32(2):145–163, Mar. 2015.

[10] Laurent Clavier, Troels Pedersen, Ignacio Larrad, Mads Lauridsen, and Malcolm Egan. Experimental evidence for heavy tailed interference in the IoT. *IEEE Wireless Commun. Lett.*, 25(3):692–695, July 2020.

[11] Elisabeth de Carvalho, Emil Björnson, Jesper H. Sørensen, Erik G. Larsson, and Petar Popovski. Random Pilot and Data Access in Massive MIMO for Machine-Type Communications. *IEEE Trans. Wireless Commun.*, 16(12):7703–7717, Sept. 2017.

[12] Xiaowen Dong, Dorina Thanou, Laura Toni, Michael Bronstein, and Pascal Frossard. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Process. Mag.*, 37(6):117–127, Oct. 2020.

[13] David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci. U.S.A.*, 106(45):18914–18919, Nov. 2009.

[14] Alexander Fengler, Saeid Haghighatshoar, Peter Jung, and Giuseppe Caire. Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver. *IEEE Trans. Inf. Theory*, 67(5):2925–2951, May 2021.

[15] Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability of graph scattering transforms. In *Proc. 33th Conf. Neural Inform. Process. Syst.*, pages 8036–8046, Vancouver, BC, Canada, 2019.

[16] Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. *IEEE Trans. Signal Process.*, 68:5680–5695, Sept. 2020.

[17] Ping Gao and Cihan Tepedelenlioglu. Space-time coding over fading channels with impulsive noise. *IEEE Trans. Wireless Commun.*, 6(1):220–229, Jan. 2007.

[18] Zhan Gao, Elvin Isufi, and Alejandro Ribeiro. Stability of graph convolutional neural networks to stochastic perturbations. 188, 108216:1–15, Nov. 2021.

[19] Zhan Gao, Amanda Prorok, and Elvin Isufi. On the Trade-Off between Stability and Representational Capacity in Graph Neural Networks. *arXiv*, Dec. 2023.

[20] GH Golub and CF Van Loan. *Matrix Computations vol. 3*. The Johns Hopkins Univ. Press, Baltimore, MD, USA, 2012.

[21] Rajarshi Guhaniyogi, Shaan Qamar, and David B Dunson. Bayesian tensor regression. *J. Mach. Learn. Res.*, 18(1):2733–2763, Aug. 2017.

[22] Ziya Gülgün and Erik G Larsson. Massive MIMO with Cauchy noise: Channel estimation, achievable rate and data decoding. *IEEE Trans. Wireless Commun.*, 23(3):1929–1942, Mar. 2023.

[23] Stephan Günnemann. *Graph Neural Networks: Adversarial Robustness*. Springer, 2022.

[24] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab., Los Alamos, NM, USA, 2008.

[25] Saeid Haghighatshoar, Peter Jung, and Giuseppe Caire. Improved scaling law for activity detection in massive MIMO systems. In *Proc. IEEE Int. Symp. Inf. Theory*, pages 381–385, Vail, CO, USA, June 17-22, 2018.

[26] R. A. Harshman. Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis. *UCLA Working Pap. Phonet.*, 16:1–84, Dec. 1970.

[27] Feng Huang, Xiang Yue, Zhankun Xiong, Zhouxin Yu, Shichao Liu, and Wen Zhang. Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations. *Briefings Bioinf.*, 22(3), July 2020.

[28] Peter J Huber. Robust estimation of a location parameter. *Ann. Math. Stat.*, 35(1):73–101, 1964.

[29] Elvin Isufi, Fernando Gama, David I. Shuman, and Santiago Segarra. Graph filters for signal processing and machine learning on graphs. *IEEE Trans. Signal Process.*, 72:4745–4781, 2024.

[30] Henry Kenlay, Dorina Thanou, and Xiaowen Dong. Interpretable stability bounds for spectral graph filters. In *Proc. 38th Int. Conf. Mach. Learning*, volume 139, pages 5388–5397, Virtual, July 18-24, 2021.

[31] Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of graph convolutional neural networks under edge rewiring. In *Proc. 46th IEEE Int. Conf. Acoustic, Speech and Signal Process.*, pages 8513–8517, Toronto, Canada, June 6-11, 2021.

[32] Soobin Kim, Sang-Soo Baek, Hyoun-Tae Hwang, Jin Hwi Kim, and Kyung Hwa Cho. Unstructured mesh-based graph neural networks for estimating the spatiotemporal distribution of a human-induced chemical in freshwater. *Water Research X*, 28:100367, Sept. 2025.

[33] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. 5th Int. Conf. Learn. Representations*, pages 1–14, Toulon, France, Apr. 24-26, 2017.

[34] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8:1519–1555, Dec. 2007.

[35] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[36] Jean Kossaifi, Zachary C. Lipton, Arinbjörn Kolbeinsson, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. Tensor regression networks. *J. Mach. Learn. Res.*, 21(1), Jan. 2020.

[37] Joseph B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18(2):95–138, 1977.

[38] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label efficient semi-supervised learning via graph filtering. In *Proc. 32nd Conf. Comput. Vision and Pattern Recognition*, pages 9574–9583, Long Beach, CA, USA, June 16-20, 2019.

[39] Xiao Li, Sameer Pawar, and Kannan Ramchandran. Sub-linear time compressed sensing using sparse-graph codes. In *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, pages 14–19. IEEE, 2015.

[40] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Stat. Biosci.*, 10(3):520–545, Mar. 2018.

[41] Lek-Heng Lim and Pierre Comon. Nonnegative approximations of nonnegative tensors. *J. Chemom.*, 23(7-8):432–441, July 2009.

[42] Liang Liu, Erik G Larsson, Wei Yu, Petar Popovski, Cedomir Stefanovic, and Elisabeth De Carvalho. Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things. *IEEE Signal Process. Mag.*, 35(5):88–99, Sept. 2018.

[43] Liang Liu and Wei Yu. Massive connectivity with massive MIMO—Part I: Device activity detection and channel estimation. *IEEE Trans. Signal Process.*, 66(11):2933–2946, June 2018.

[44] Sicong Liu, Fang Yang, Wenbo Ding, and Jian Song. Double kill: Compressive-sensing-based narrow-band interference and impulsive noise mitigation for vehicular communications. *IEEE Trans. Veh. Technol.*, 65(7):5099–5109, July 2015.

[45] Zhen Long, Ce Zhu, Jiani Liu, and Yipeng Liu. Bayesian low rank tensor ring for image recovery. *IEEE Trans. Image Process.*, 30:3568–3580, Mar. 2021.

[46] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11(1), 2010.

[47] Leatile Marata. *Advanced signal processing techniques for machine type communications*. PhD thesis, Oulu University, 2024.

[48] Leatile Marata, Onel Luis Alcaraz López, Hamza Djelouat, Markus Leinonen, Hirley Alves, and Markku Juntti. Joint coherent and non-coherent detection and decoding techniques for heterogeneous networks. *IEEE Trans. Wireless Commun.*, 22(3):1730–1744, Sept. 2022.

[49] Leatile Marata, Esa Ollila, and Hirley Alves. Activity detection for massive random access using covariance-based matching pursuit. *IEEE Trans. Veh. Technol.*, pages 1–11, May 2025.

[50] R. A. Maronna, R. D. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. Wiley, New York, 2006.

[51] David Middleton. Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models. *IEEE Trans. Inf. Theory*, 45(4):1129–1149, May 1999.

[52] Jari Miettinen, Sergiy A Vorobyov, and Esa Ollila. Modelling and studying the effect of graph errors in graph signal processing. 189, 108256:1–8, Dec. 2021.

[53] Guy Ohayon, Tomer Michaeli, and Michael Elad. The perception-robustness tradeoff in deterministic image restoration. In *Proc. 41st Int. Conf. Mach. Learning*, volume 235, pages 38599–38638, Vienna, Austria, July 21-27, 2024.

[54] Esa Ollila. Matching pursuit covariance learning. In *Proc. 32nd Eur. Signal Process. Conf.*, pages 2447–2451, Lyon, France, Aug. 26-30, 2024.

[55] Esa Ollila, Ilya Soloveychik, David E. Tyler, and Ami Wiesel. Simultaneous penalized M-estimation of covariance matrices using geodesically convex optimization. *Arxiv:1608.08126v1*, 2016.

[56] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, Sept. 2011.

[57] Mathew Penrose. Random geometric graphs. *Oxford Stud. in Probab.*, 5, 2003.

[58] JC Poythress, Jeongyoun Ahn, and Cheolwoo Park. Low-rank, orthogonally decomposable tensor regression with application to visual stimulus decoding of fMRI data. *J. Comput. Graphical Stat.*, 31(1):1–14, Aug. 2021.

[59] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1):259–268, Nov. 1992.

[60] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Trans. Signal Process.*, 61(7):1644–1656, Apr. 2013.

[61] Kamil Senel and Erik G Larsson. Grant-free massive mtc-enabled massive mimo: A compressive sensing approach. *IEEE Trans. Commun.*, 66(12):6164–6175, Dec. 2018.

[62] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.*, 65(13):3551–3582, July 2017.

[63] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc., Ser. B*, 58(1):267–288, 1996.

[64] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. B*, 67:91–108, Dec. 2005.

[65] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Process.*, 86(3):572–588, Mar. 2006.

[66] Trinh Van Chien, Emil Björnson, and Erik G. Larsson. Joint Pilot Design and Uplink Power Allocation in Multi-Cell Massive MIMO Systems. *IEEE Trans. Wireless Commun.*, 17(3):2000–2015, Jan. 2018.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. 31st Conf. Neural Inform. Process. Syst.*, page 6000–6010, 2017.

[68] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. 6th Int. Conf. Learn. Representations*, pages 1–12, Vancouver, BC, Canada, Apr. 30 - May 3, 2018.

[69] Xinjue Wang, Esa Ollila, and Sergiy A Vorobyov. Graph neural network sensitivity under probabilistic error model. In *Proc. 30th Eur. Signal Process. Conf.*, pages 2146–2150, Belgrade, Serbia, Aug. 29 - Sept. 2, 2022.

[70] Boris Weisfeiler and AA Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[71] David P Wipf and Bhaskar D Rao. Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.*, 52(8):2153–2164, Aug. 2004.

[72] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza, Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. In *Proc. 36th Int. Conf. Mach. Learning*, pages 6861–6871, Long Beach, California, USA, June 9-15, 2019.

[73] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24, Mar. 2021.

[74] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. In *Proc. 28th Int. Joint Conf. Artif. Intell.*, pages 3961–3967, Macao, China, Aug. 10-16, 2019.

[75] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proc. 7th Int. Conf. Learn. Representations*, pages 1–17, New Orleans, LA, USA, May 6-9, 2019.

[76] Yao Lei Xu, Kriton Konstantinidis, and Danilo P Mandic. Graph-regularized tensor regression: A domain-aware framework for interpretable modeling of multiway data on graphs. *Neural Computation*, 35(8):1404–1429, 2023.

[77] Hao Yan, Kamran Paynabar, and Massimo Pacella. Structured point cloud data analysis via regularized tensor regression for process modeling and optimization. *Technometrics*, 61(3):385–395, June 2019.

[78] Jun Zhang and Ziping Zhao. Improved Stability Bounds for Graph Convolutional Neural Networks Under Graph Perturbations. In *2024 IEEE Inf. Theory Workshop*, pages 24–28. IEEE.

[79] Ning Zhang, Henry Kenlay, Li Zhang, Mihai Cucuringu, and Xiaowen Dong. On the Stability of Graph Convolutional Neural Networks: A Probabilistic Perspective. *arXiv*, June 2025.

[80] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv*, June 2016.

[81] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *J. Am. Stat. Assoc.*, 108(502):540–552, July 2013.

[82] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B*, 67(2):301–320, Mar. 2005.

[83] Abdelhak M Zoubir, Visa Koivunen, Yacine Chakhchoukh, and Michael Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Process. Mag.*, 29(4):61–80, June 2012.

[84] Abdelhak M Zoubir, Visa Koivunen, Esa Ollila, and Michael Muma. *Robust Statistics for Signal Processing*. Cambridge University Press, 2018.

Business, Economy

Art, Design, Architecture

Science, Technology

Crossover

Doctoral Theses

Aalto DT 249/2025

ISBN 978-952-64-2879-6
ISBN 978-952-64-2878-9 (pdf)

Aalto University
School of Electrical Engineering
Department of Information
and Communications Engineering
aalto.fi