

**Machine Learning and Stochastic Geometry Techniques for Future Mobile
Communications**

by

Bitan Banerjee

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Communications

Department of Electrical & Computer Engineering
University of Alberta

© Bitan Banerjee, 2023

Abstract

Advancements in wireless communication are continuously evolving, and the progression towards the 6th generation (6G) and beyond of cellular architecture will heavily rely on the implementation of machine learning (ML) algorithms in both cellular devices and base stations (BSs), along with the deployment of highly dense networks. ML algorithms have the potential to grant devices the ability to autonomously adapt and modify themselves, while also facilitating decision-making processes involving non-deterministic polynomial-time (NP)-hard problems. Therefore, this thesis explores the potential of machine learning (ML) algorithms in shaping the future of wireless communication and cellular architecture. Specifically, it focuses on addressing the challenges faced by conventional architectures in meeting the data rate and reliability requirements of the anticipated 6G cellular architecture. The research investigates the application of machine learning and stochastic geometry techniques to propose novel approaches for enhancing performance and overcoming limitations.

The thesis presents a heterogeneous network (HetNet) model that correlates the locations of small cell base stations (SBSs) with macro base stations (MBSs) using a Poisson-Voronoi tessellation. Theoretical analysis of this deployment scheme is studied using the tools of stochastic geometry and the results indicate an improvement up to 21% in the coverage probability and up to 28% in the rate coverage. This thesis also introduces a conditional generative adversarial network (CGAN)-based algorithm for uplink (UL) to downlink (DL) channel covariance matrix (CCM) mapping and direct UL to DL channel state information (CSI) mapping in massive MIMO systems operating in a frequency division duplex (FDD) mode.

Additionally, the research explores multi-agent reinforcement learning (MARL) and

multi-agent federated reinforcement learning (MAFRL) algorithms for access point (AP) selection and clustering in cell-free networks. The MARL and MAFRL algorithms provide a sub-optimal solution to an NP-hard problem and achieve up to 88.3% of the maximum possible sum spectral efficiency achievable if all APs were to serve all users using centralized precoding. Furthermore, the thesis investigates the combination of long short-term memory (LSTM) and CGAN for predicting downlink CSI from earlier uplink CSI estimates and estimating complete uplink CSI from incomplete information. This algorithm demonstrates the ability to provide reliable network service to users moving at vehicular speeds with limited available power.

Through these approaches, the thesis aims to contribute to the development of more efficient and reliable cellular systems for 6G and beyond. The research findings demonstrate the potential of ML algorithms and highlight the benefits of integrating stochastic geometry and machine learning techniques in wireless communication systems.

Preface

The essential content of Chapter 3 is published in the following conference paper:

- B. Banerjee, R. C. Elliott, W. A. Krzymień and J. Melzer, “Correlated Placement of Small Cell Base Stations: A Coverage Enriched HetNet with Massive MIMO,” in *Proc. 91st IEEE Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium (held virtually), May 2020, 6 IEEE-format pages.

The essential content of Chapter 4 is published in the following conference and journal papers:

- B. Banerjee, R. C. Elliott, W. A. Krzymień and H. Farmanbar, “Towards FDD Massive MIMO: Downlink Channel Covariance Matrix Estimation Using Conditional Generative Adversarial Networks,” in *Proc. 32nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2021)*, Helsinki, Finland (held virtually), Sept. 2021, pp. 940-946; *PIMRC 2021 Best Student Paper Award*.
- B. Banerjee, R. C. Elliott, W. A. Krzymień and H. Farmanbar, “Downlink Channel Estimation for FDD Massive MIMO Using Conditional Generative Adversarial Networks,” in *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 122-137, Jan. 2023.

The essential content of Chapter 5 is published in the following conference and journal papers:

- B. Banerjee, R. C. Elliott, W. A. Krzymień and H. Farmanbar, “Access point clustering in cell-free massive MIMO using multi-agent reinforcement learning,” in *33rd*

IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2022), Kyoto, Japan (held virtually), Sept. 2022, pp. 794-800.

- B. Banerjee, R. C. Elliott, W. A. Krzymień and M. Medra, “Access Point Clustering in Cell-Free Massive MIMO Using Conventional and Federated Multi-Agent Reinforcement Learning,” in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 107-123, 2023. DOI:10.1109/TMLCN.2023.3283228.

The essential content of Chapter 6 is published in the following conference paper:

- B. Banerjee, R. C. Elliott, W. A. Krzymień and M. Medra, “Machine Learning Assisted DL CSI Estimation for High-Mobility Multi-Antenna Users with Partial UL CSI Availability in TDD Massive MIMO Systems,” in *Proc. IEEE GLOBECOM 2022 Workshop on Emerging Topics in 6G Communications*, Rio de Janeiro, Brazil (hybrid conference), Dec. 2022, pp. 1579-1585.

For each publication listed above, I carried out the system model development, the mathematical analysis, and computer simulations under the supervision of Prof. Witold Krzymień and in cooperation with Dr. Robert C. Elliott (all chapters), Dr. Jordan Melzer (for Chapter 3), Dr. Hamid Farmanbar (for Chapters 4 and 5), and Dr. Mostafa Medra (for Chapters 5 and 6). I was responsible for the manuscript composition, and Dr. Elliott and Prof. Witold Krzymień contributed to its content and writing.

Acknowledgements

There are several individuals to whom I am greatly indebted, as this thesis would not have been possible without their support. First and foremost, I express my deep gratitude to Prof. Witold Krzymień, who expertly guided me through my graduate education and shared in the excitement of discovery. His unwavering enthusiasm kept me focused on research, and his personal generosity made my time at the University of Alberta truly enjoyable. He supported me during challenging times and always believed in my skills. I am forever grateful to him.

I would also like to extend my thanks to Dr. Robert C. Elliott for his incredible support and guidance. He has assisted me in every aspect of my research, from solving complex integrations to refining my paper writing skills. I am indebted to him for his unwavering support.

Additionally, I would like to express my appreciation to Prof. Jing, Prof. Tellambura, Prof. Ardakani, Prof. Fair, Prof. Jiang, and Prof. Hossain for their valuable comments and suggestions, which greatly enhanced the quality of this thesis. I am also grateful to my lab mates for their support and camaraderie. I extend my thanks to the faculty and administrative staff of the department for their continued support throughout my PhD journey.

I am grateful for the financial support I received from Huawei Technologies Canada Co. Ltd., TELUS Communications, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Rohit Sharma Professorship. I would also like to acknowledge the computing resources provided by Information Services and Technology (IST) at the University of Alberta and Google Colaboratory.

To my beloved parents, Maa Bani Banerjee and Baba Karunamoy Banerjee, I am pro-

foundly indebted. From the very beginning, you blessed me with the precious gift of life and have stood by my side through every triumph and tribulation. Your unwavering love, guidance, and sacrifices have shaped my life in immeasurable ways. I am humbled and privileged to have such loving and supportive parents who have been my guiding stars. Thank you, Maa and Baba, for being the bedrock of my life and for always believing in me.

Foremost, I would like to express my deepest gratitude to Priyanka Ray (Jhil) for her unwavering love and support throughout this journey. Her presence has been a beacon of light, guiding me through the challenging moments and bringing a smile to my face even when the odds seemed against me. Words will always fall short in expressing the depth of my gratitude for you, Jhilmil. You are my rock, my greatest strength, and the person without whom I would never have been able to write this thesis. Thank you, from the bottom of my heart, for being my source of encouragement, unwavering love, and support throughout this remarkable journey.

I would also like to thank my grandparents, Mrs. Bela Bandyopadhyay, Mrs. Alpana Chakraborty, Mr. Shaktipada Bandyopadhyay, and Mr. Anil Chakraborty, for gifting me with amazing parents and filling my childhood with the sweetest and happiest memories. I deeply miss you, and I believe you would have been overjoyed today.

Finally, I would like to thank all my friends and well-wishers in this faraway land from home.

Table of Contents

1	Introduction	1
1.1	Evolution of Mobile Communications	1
1.1.1	Massive MIMO Networks	4
1.1.2	Cell-free Massive MIMO Networks	5
1.1.3	Heterogeneous Cellular Networks	7
1.2	Machine Learning in Mobile Communications	9
1.3	Thesis Objectives and Organization	11
1.3.1	Motivation and Problem Statement	11
1.3.2	Thesis Outline and Contributions	12
2	Background	16
2.1	Stochastic Geometric Modeling of Cellular Networks	16
2.1.1	Poisson Point Process	17
2.1.2	Poisson Cluster Processes	18
2.1.3	Hardcore Point Processes	19
2.1.4	Softcore Point Processes	20
2.2	Massive MIMO Systems	21
2.3	Cell-free Massive MIMO	24
2.4	Machine Learning Algorithms for Wireless Systems	27
2.4.1	Deep Neural Network	27
2.4.2	Generative Adversarial Networks	28
2.4.3	Recurrent Neural Networks	30
2.4.4	Federated Learning	32
3	Coverage Enhancement of Massive MIMO HetNets by Correlated Placement of SBSs	34
3.1	Introduction	34
3.1.1	Background & Motivation	34

3.2	System Model	38
3.2.1	User Association	39
3.2.2	Channel Model	40
3.2.3	Signal-to-Interference-plus-Noise Ratio	40
3.3	Distance Distributions	41
3.4	Tier Association Probabilities and Conditional Distance Distributions	50
3.4.1	Tier Association Probability	51
3.4.2	Conditional Distance Distribution	52
3.5	Coverage Probability & Rate Coverage	53
3.5.1	Coverage Probability	54
3.5.2	Achievable Rate & Rate Coverage	59
3.6	Performance Evaluation	60
3.7	Summary	69
4	Downlink Channel Estimation For FDD Massive MIMO Using Conditional Generative Adversarial Networks	70
4.1	Introduction	70
4.1.1	Prior Research	71
4.1.2	Motivation	73
4.2	CCM Modeling	75
4.3	System Model for Simulated CSI Dataset Generation	78
4.4	CGAN Framework	82
4.4.1	Objective of CGAN	84
4.4.2	Architecture of CGAN	85
4.5	Performance of CCM Estimation	90
4.6	Performance of CSI Estimation	94
4.7	Summary	100
5	Access Point Clustering in Cell-Free Massive MIMO Using Conventional and Federated Multi-Agent Reinforcement Learning	102
5.1	Introduction	102
5.1.1	Prior Research	103
5.1.2	Motivation	106
5.2	Cell-Free Massive MIMO System Model	108
5.3	Reinforcement Learning Framework	110
5.4	Federated Reinforcement Learning Framework	117
5.4.1	Complexity of MARL and MAFRL decisions	120

5.5	Performance Evaluation	121
5.6	Summary	135
6	Machine Learning Assisted DL CSI Estimation for High-Mobility Multi-Antenna Users	136
6.1	Introduction	136
6.1.1	Background & Motivation	137
6.2	Channel Dataset Generation Methodology	139
6.3	Machine Learning Framework	142
6.3.1	LSTM Overview	142
6.3.2	Overall ML Structure	145
6.4	Simulated Performance Evaluation	145
6.5	Summary	150
7	Conclusion	151
7.1	Summary of Contributions	151
7.2	Possible Future Research Directions	154
	Bibliography	156
	Appendix A:	180
A.1	Proof of Theorem 1 from Chapter 3	180
A.2	Proof of Theorem 3 from Chapter 3	181
A.3	Proof of Corollary 1 from Chapter 3	189
A.4	Proof of Lemma 2 from Chapter 3	190
A.5	Proof of Lemma 4 from Chapter 3	191
A.6	Proof of Lemma 5 from Chapter 3	194

List of Tables

4.1	DL CCM NMSE for $N = 500$ Imperfect \mathbf{h} Samples, Antenna Array Size $M = 256$, and Varying Channel Estimation Reliability ζ	95
4.2	Simulation Parameters for CSI Dataset Generation	97
5.1	Sum SE for $\Phi = 10$, Max Sum SE Policies, and Varying Number of Training Episodes	129
6.1	Parameters of ML Neural Networks	144
6.2	NMSE for $U = 300$ LSTM cells, $M = 128$, $T_s = 10 \mu\text{s}$, and varying depth D of LSTM layers	148

List of Figures

1.1	Growth of mobile data traffic [1]	2
3.1	Illustration of the proposed two-tier HetNet, where MBSs (illustrated as red triangles) are located according to a PPP and SBSs (illustrated as green squares) are placed at the vertices of the Poisson-Voronoi tessellation (i.e., corners of the macrocells). The users are illustrated as black dots.	38
3.2	Important distances and angles in a macrocell. R_{oM} is the distance between the typical user ‘ o ’ and its nearest MBS, R_{oS} is the distance between the typical user and its nearest SBS, and R_{MS} is the distance between these two specific BSs. θ is the angle between the lines measuring R_{oM} and R_{MS}	42
3.3	Probability density function of R_{MS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$	43
3.4	Probability density function of R_{oS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$	45
3.5	Cumulative distribution function of \tilde{R}_{oS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$	49
3.6	Coverage probability vs. coverage threshold SINR τ_c , comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment schemes from [105] and [39]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm. (a) $\lambda_S = 2\lambda_M$ and $\lambda'_S = 1.378\lambda_M$. (b) $\lambda_S = 5\lambda_M$ and $\lambda'_S = 3.445\lambda_M$	62
3.7	Rate coverage (complementary CDFs of rate per user), comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment schemes from [105] and [39]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm.	63
3.8	Contours of coverage probability vs. MBS transmit power P_1 and coverage threshold SINR τ_c , comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment scheme from [105]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_2 = 30$ dBm.	64

3.9	Contours of coverage probability vs. association bias factor B and SBS transmit power P_2 , comparing correlated SBS placement strategy with β -GPP and PPP approximations. $N_M = 100$, $U = 10$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $\tau_c = 5$ dB.	66
3.10	Contours of coverage probability vs. number of transmit antennas per MBS (N_M) and number of users associated with each MBS (U), comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment scheme from [105]. $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm, $\tau_c = 5$ dB.	67
3.11	Joint PDF of received SINR at user and distance from nearest MBS, comparing correlated SBS placement strategy with scheme from [105]. $N_M = 100$, $U = 10$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm. (a) 3-dimensional PDF. (b) Contour plots showing where PDF equals $\{1, 3, 5, \dots\} \times 10^{-5}$	68
4.1	Network layout considered for UL and DL CSI dataset generation.	78
4.2	Illustration of pilot reuse factor ϖ (cf. [3]).	80
4.3	Block diagram of our CGAN algorithm.	82
4.4	Example images of exact UL CCMs for $M = 256$	83
4.5	Example images of UL CSI (rotated by 90°) for $M = 256$, $K = 10$, and several values of ϖ	84
4.6	Neural network architecture of generator block.	86
4.7	Neural network architecture of discriminator block.	86
4.8	Operation of 2D convolutional filter, illustrating the first stage of our CGAN generator for $M \times M$ input image. (a) Structure of filters, neural networks, and outputs on each layer. (b) Movement of filter according to stride size.	87
4.9	Example CCM and CSI images for $M = 256$, $K = 10$, and $\varpi = 4$. (a) Exact UL CCM and MMSE-estimated UL CSI. (b) The corresponding exact CCM and MMSE-estimated CSI for the DL. (c) DL CCM and CSI generated by CGAN.	88
4.10	NMSE performance comparison of several DL CCM estimation methods, using exact CCMs with $M = 256$. Error brackets represent 95% confidence interval.	91
4.11	NMSE performance comparison when using imperfect CCMs with varying number of samples N ; $M = 256$. Error brackets represent 95% confidence interval.	92

4.12	NMSE performance comparison between exact and imperfect CCMs for varying antenna array size M . Error brackets represent 95% confidence interval.	92
4.13	Generator loss for varying number of samples N . Light blue line and envelope respectively represent mean value and range of values observed during simulations with dropout regularization; dark grey line and envelope respectively represent mean value and range of values observed during simulations without dropout regularization.	93
4.14	NMSE performance comparison of CGAN, DNN adapted from [169], and PG-DNN adapted from [171] for varying antenna array size M with $\varpi = 1$. Error brackets represent 95% confidence interval.	96
4.15	CGAN NMSE performance comparison for varying antenna array size with several pilot reuse factors. Error brackets represent 95% confidence interval	98
4.16	Sum spectral efficiency per cell performance results for two different antenna array sizes $M = 32$ and 128 , three pilot reuse factor values $\varpi = 1, 2$, and 4 , and three precoding schemes (RZF, ZF, and MR). Three types of DL CSI availability (genie-aided, feedback-based, and CGAN-based) are compared. Error brackets represent 95% confidence interval.	99
4.17	NMSE performance for measured CSI data of 8×8 UPA from [185, 186], for varying number of training epochs. Error brackets represent 95% confidence interval.	100
5.1	Illustration of a cell-free massive MIMO network. (AP: access point, CC: central controller, UE: user equipment.)	106
5.2	Illustration of the decentralized actor, centralized critic MARL algorithm's interactions between the environment and agents.	114
5.3	Illustration of the MAFRL algorithm's interactions between the central controller, environment, and agents.	119
5.4	Average sum SE performance of MARL and MAFRL algorithms with several values of Φ , compared against "All" and "Greedy" strategies, max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221]. "P1": Max sum SE policy having local and global penalties, "P2": Max min SE policy, "P3": Modified max min SE policy, "P4": Hybrid policy, "P5": Max sum SE policy having global penalty only.	123

5.5	CDFs of UEs’ instantaneous SEs for MARL and MAFRL algorithms with $\Phi = 15$ under five reward policies, compared with schemes from [211], [217], and [221]. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only. (a) UE SEs per AP. (b) UE SEs summed over their serving APs.	126
5.6	Average sum SE performance of MARL and MAFRL algorithms under max sum SE policies with several values of Φ , max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221], when the AP locations during testing differ from those during training. Arrows show drop in MARL performance compared to when the training and testing environments match; performance of MAFRL algorithm is mostly unaffected. Performance of “All” strategy also shown for reference. “P1”: Max sum SE policy having local and global penalties, “P5”: Max sum SE policy having global penalty only.	130
5.7	Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and several values of UE speed v , having been trained at 1 m/s. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only.	132
5.8	Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and varying ratios of total number of AP antennas to number of UEs. Performance is compared against “All” and “Greedy” strategies, max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221]. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only. (a) $K = 20$ UEs, varying numbers of antennas per AP (N). (b) $N = 4$, varying number of UEs (K).	134
6.1	Block diagram of LSTM unit (cf. [286]). The addition, multiplication, $\tanh(\cdot)$, and sigmoid ($\sigma(\cdot)$) functions operate on a per-element basis.	143

6.2	Block diagram of CGAN-LSTM-based algorithm for DL CSI estimation. The algorithm works on image pixel values; CSI data matrices are converted to images before the depicted functions, then back to data matrices afterwards. There may alternatively be K identical copies of the LSTM, dense, and reshape layers, with the UE vectors processed in parallel instead of serially.	145
6.3	NMSE performance of CGAN component alone and of CGAN-LSTM algorithm when an $AR(P)$ process is assumed for the LSTM network. Lower orange bars represent $T_s = 10 \mu s$; higher blue bars represent $T_s = 1 ms$. . .	147
6.4	DL sum SE performance for MF precoding, comparing complete CSI, our CGAN-LSTM algorithm, the method from [271], and mean CSI infilling. Higher purple bars represent $T_s = 10 \mu s$; lower blue bars represent $T_s = 1 ms$. (Complete CSI is the same for both values of T_s .)	149
6.5	DL sum SE performance for BD precoding, comparing complete CSI, our CGAN-LSTM algorithm, the method from [271], and mean CSI infilling. Higher purple bars represent $T_s = 10 \mu s$; lower blue bars represent $T_s = 1 ms$. (Complete CSI is the same for both values of T_s .)	149
A.1	PDF of $\cos \theta$	182
A.2	$\frac{\partial}{\partial k} \tilde{\Gamma}(k, ck)$ (or $-\frac{\partial}{\partial k} \tilde{\gamma}(k, ck)$) vs. c and k . Blue plane indicates where z-axis equals zero; thick dashed line indicates where $c = 1$	184

List of Notations

Notation	Meaning
Italic variable (e.g. a, A)	Scalar
Lowercase boldface variable (e.g. \mathbf{a})	Vector
Uppercase boldface variable (e.g. \mathbf{A})	Matrix
Calligraphic variable (e.g. \mathcal{A})	Set
Boldface calligraphic variable (e.g. \mathcal{A})	Family of sets
$ a $ or $ A $	Magnitude of a or A
$ \mathcal{A} $	Cardinality of \mathcal{A}
$ \cdot $	Area if the variable is a region in 2-D space
a^*	Optimal value of a
a^*, \mathbf{A}^*	Complex conjugate of (every element in) a, \mathbf{A}
\mathbf{A}^T	Transpose of \mathbf{A}
\mathbf{A}^H	Hermitian (conjugate) transpose of \mathbf{A}
$ \mathbf{A} $	Determinant of \mathbf{A} (when \mathbf{A} is square)
\mathbf{A}^{-1}	Inverse of \mathbf{A} (when \mathbf{A} is square)
$\mathbf{A}^{1/2}$	Matrix square root of \mathbf{A} (when \mathbf{A} is square)
$\text{tr}(\mathbf{A})$	Trace of \mathbf{A} (when \mathbf{A} is square)
$\text{vec}(\mathbf{A})$	Vectorization operator; transforms \mathbf{A} into a vector by vertically stacking its columns
$\ \mathbf{A}\ _F$	Frobenius norm of \mathbf{A}
$\ \mathbf{A}\ _1$	L1 distance/norm of \mathbf{A}
$\ x\ $	Euclidian norm of x
\mathbf{I}_n	The $n \times n$ identity matrix

$\mathbf{0}_{m \times n}$	An $m \times n$ matrix containing all zeros
\odot	Hadamard product
$f_X(\cdot)$	Probability density function of X
$F_X(\cdot)$	Cumulative distribution function of X
$\mathbb{E}\{\cdot\}$	Expectation operator
\otimes	Kronecker product
$\mathbf{1}(x)$	Indicator function; equals 1 if condition x is true, and 0 otherwise

Abbreviations

1G 1st generation of cellular network/system standards.

2G 2nd generation of cellular network/system standards.

3G 3rd generation of cellular network/system standards.

3GPP 3rd Generation Partnership Project.

4G 4th generation of cellular network/system standards.

5G 5th generation of cellular network/system standards.

6G 6th generation of cellular network/system standards.

ACF Autocorrelation function.

AoA Angle of arrival.

AoD Angle of departure.

AP Access point.

AR Autoregressive.

BD Block-diagonalization.

BPP Binomial point process.

BS Base station.

C-RAN Cloud radio access network.

CC Central controller.

CCM Channel covariance matrix.

CDF Cumulative distribution function.

CGAN Conditional generative adversarial network.

CoMP Coordinated multipoint transmission/reception.

CSI Channel state information.

DAS Distributed antenna system.

DCGAN Deep convolutional generative adversarial network.

DL Downlink.

DNN Deep neural network.

EB Exabyte (1 EB = 1000^6 bytes).

EE Energy efficiency.

ESPRIT Estimation of Signal Parameters using Rotational Invariance Techniques.

FDD Frequency division duplex.

FL Federated learning.

FR1 Frequency range 1.

FR2 Frequency range 2.

GAN Generative adversarial network.

GPP Ginibre point process.

GRU Gated recurrent unit.

HetNet Heterogeneous cellular network.

i.i.d. Independent and identically distributed.

IoT Internet of Things.

LP-MMSE Local partial minimum mean square error.

LSTM Long short-term memory.

MAFRL Multi-agent federated reinforcement learning.

MARL Multi-agent reinforcement learning.

MBS Macro base station.

MF Matched filtering.

MIMO Multiple-input multiple-output.

ML Machine learning.

MMSE Minimum mean squared error.

mmWave Millimeter wave radio frequency range.

MR Maximum ratio.

MUSIC MUltiple SIgnal Classification.

NMSE Normalized mean square error.

NN Neural network.

P1 to P5 Policy 1 to Policy 5 (for reinforcement learning rewards).

PAS Power angular spectrum.

PCP Poisson cluster process.

PDF Probability density function.

PPP Poisson point process.

ReLU Rectified Linear Unit activation function.

RL Reinforcement learning.

RMSE Root-mean-square error.

RNN Recurrent neural network.

RRH Remote radio head.

RSS Received signal strength.

RV Random variable.

SBS Small-cell base station.

SC Small cell.

SE Spectral efficiency.

SINR Signal-to-interference-plus-noise ratio.

SISO Single-input single-output.

StoGeo Stochastic Geometry.

TDD Time division duplex.

UE User equipment.

UL Uplink.

URLLC Ultra-reliable low-latency communication.

WGAN Wasserstein generative adversarial network.

ZFBF Zero-forcing beamforming.

Chapter 1

Introduction

1.1 Evolution of Mobile Communications

Mobile communication systems have truly revolutionized the way we connect and communicate with each other, and their evolution shows no signs of slowing down. With each passing decade, a new generation emerges, pushing the boundaries of what is possible in terms of services and coverage. The mobile cellular systems journey began in 1979, and since then, we have witnessed the rise of the first generation (1G), second generation (2G), third generation (3G), fourth generation (4G), and now, fifth generation (5G) cellular networks. The advent of 3G brought about significant improvements over its predecessor, enabling higher mobile data speeds and more advanced services. However, the arrival of 4G truly transformed the landscape. Mobile broadband became dominant over fixed-line broadband services like digital subscriber line (DSL) and fiber or cable broadband. Thus, smartphones, capable of running myriad applications, have become an indispensable part of our daily lives. The demand for mobile data skyrocketed as people embraced streaming, social media, and other bandwidth-intensive activities.

As we venture into the era of 5G and beyond, the need for even higher mobile data rates becomes apparent. High-definition videos, immersive media like augmented and virtual reality, and the Internet of Things (IoT) are examples of emerging bandwidth-demanding applications. The number of mobile subscribers globally has already reached a staggering 8.4 billion, and this figure is projected to surpass 9 billion by 2028 [1]. With around 86%

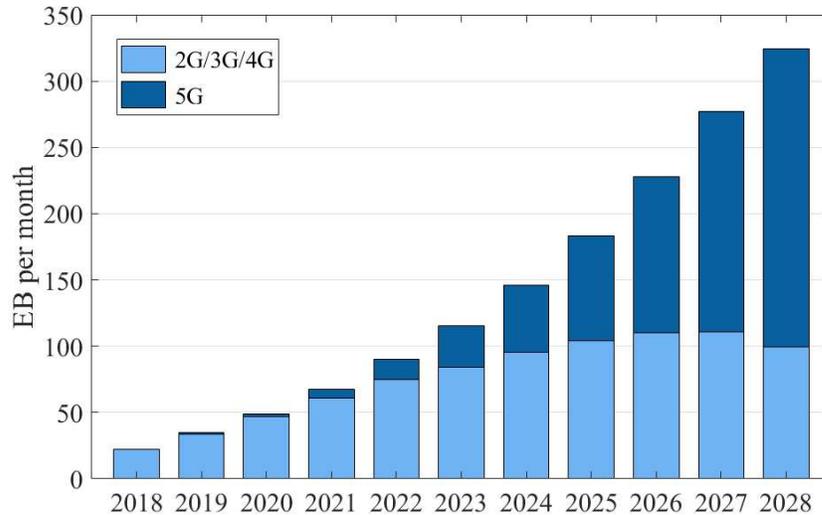


Figure 1.1: Growth of mobile data traffic [1]

of these users having access to mobile broadband, the impact of mobile communication on our lives will only continue to grow. While the growth of smartphone subscriptions may be reaching saturation point, the demand for mobile data, especially for 5G users, shows no signs of slowing down. 5G subscribers are forecasted to grow from 1 billion to 5 billion during the 2022-28 period. According to the mobility report from Ericsson [1], there will be a 40–fold increase in 5G data traffic by 2020-28, and by 2030 monthly 5G data traffic per user will surge to a staggering 257 GB [2]. The growth of mobile data traffic is illustrated in Fig. 1.1. This exponential growth poses both challenges and opportunities for mobile network service providers, who must continually innovate and expand their infrastructure to meet the ever-increasing demands of users.

As we progress towards the era of 5G and beyond, several crucial deployment techniques emerge as key drivers of innovation in mobile communication systems. These include massive multiple-input multiple-output (MIMO) antenna systems, heterogeneous networks (HetNets), network densification techniques like cell-free massive MIMO, utilization of higher radio frequency spectra like millimeter-wave (mmWave) frequencies, and the integration of machine learning (ML) algorithms into macro and small-cell base stations (MBSs and SBSs) – all of which are expected to play significant roles in shaping the future of sixth

generation (6G) networks [2, 3].

However, despite these technologies' immense potential, certain challenges exist within the current landscape of MIMO cellular networks. Two primary obstacles that demand attention are inter-cell interference and poor cell-edge coverage. Interference between neighboring cells can hinder network performance, resulting in degraded signal quality and reduced system capacity. Furthermore, cell-edge coverage refers to the ability of BSs to provide reliable connectivity at the outskirts of their coverage areas, where signal strength tends to diminish, and interference is higher. Overcoming these challenges is crucial to ensure seamless connectivity and optimal user experiences in future wireless networks.

Incorporating ML algorithms into base stations introduces new challenges that must be addressed too. One such challenge is power consumption. ML algorithms typically require significant computational resources, leading to increased power consumption. In the context of base stations, which are essential for network operation, minimizing power consumption is of utmost importance to ensure sustainable and efficient network deployment. Additionally, enhancing the computational capability of base stations becomes crucial to accommodate the increased processing demands of ML algorithms. Furthermore, ML algorithms must operate seamlessly in real-time environments to effectively handle dynamic network conditions and evolving user demands. Adapting and learning from real-time data is vital for ML algorithms to make accurate predictions and decisions, optimizing network performance. Designing ML algorithms that can operate efficiently in real-time scenarios without compromising the reliability and stability of the network poses a significant research challenge.

To tackle these challenges, there is a need to develop novel cellular architectures that leverage the potential of technologies like massive MIMO and HetNets while addressing issues such as inter-cell interference and cell-edge coverage. These architectures should incorporate intelligent ML algorithms at the base stations to enhance network performance and user experiences while considering power consumption, computational capability, and

real-time operation requirements. This holistic approach ensures that the network infrastructure and algorithms work synergistically to meet the evolving needs of future mobile communication systems.

1.1.1 Massive MIMO Networks

A BS equipped with a massive MIMO antenna array, where the number of antenna elements is in the order of 10^2 to 10^3 , significantly enhanced the capacity and reliability of a wireless channel between the BS and user equipment (UE) without the use of additional spectrum [3]. With MIMO (whether massive or not), two fundamental improvements are achieved compared to single-antenna structures: array gain and spatial multiplexing gain. By utilizing multiple transmit antennas, the base station (BS) can employ directional beamforming to steer signals toward users, resulting in an improved signal-to-interference-plus-noise ratio (SINR) for each user, which is known as array gain. Additionally, MIMO enables the transmitter to concurrently transmit multiple data streams within the same frequency band with different directional beamforming vectors, thereby increasing the data rate, known as spatial multiplexing gain. Furthermore, MIMO networks provide spatial diversity gain, which ensures reliable connectivity in ultra-reliable low-latency communication (URLLC) scenarios. By transmitting redundant information streams along multiple spatial paths, the system becomes resilient to outages and ensures uninterrupted communication. It is important to note that apart from higher capacity, spectral efficiency (SE), and improved coverage, massive MIMO in particular, has two more important advantages over a traditional MIMO, named favorable propagation and channel hardening, which are explained in detail in Section 2.2.

Additionally, massive MIMO systems exhibit near-optimal performance with simple linear precoders and detectors [4] and enhanced robustness against fading and interference [5]. These advantages have led to the adoption of massive MIMO transceivers in the 3rd Generation Partnership Project (3GPP) standards starting in Release 15 [6]. Moreover, massive MIMO has been successfully commercialized and implemented in practical cellular sys-

tems [7]. Massive MIMO systems typically operate in time division duplex (TDD) mode due to the radio channel reciprocity feature. In frequency division duplex (FDD) mode, the uplink (UL) and downlink (DL) transmissions occur simultaneously but on significantly separated carrier frequencies. On the other hand, TDD mode employs the same frequency band for UL and DL transmissions but at different times during the transmission frame. Applying beamforming techniques is an essential part of massive MIMO, and knowledge of DL channel state information (CSI) is necessary to obtain optimal beamforming weights for DL transmission. UL CSI is obtained from the pilot signals transmitted by the UE. In TDD mode DL CSI is easily obtained from UL CSI using the radio channel reciprocity property of TDD. However, in the FDD mode of communication, radio channel reciprocity is not applicable as the carrier frequency for UL and DL are different and small-scale fading is dependent on the carrier frequency. Therefore, for massive MIMO communication, the TDD mode of communication is conventional. However, current cellular infrastructure, especially voice communication largely operates in FDD mode. Thus there is a motivation behind the implementation of massive MIMO operating in FDD mode. In Chapter 4, we discussed the technologies for implementing FDD massive MIMO in greater detail.

1.1.2 Cell-free Massive MIMO Networks

The concept of massive MIMO can be implemented in a distributed manner, eliminating the need for traditional cells. This approach is known as cell-free massive MIMO [8]. In a cell-free massive MIMO system, there exists a central controller (CC) alongside a network of distributed access points (APs) [9]. These APs can be equipped with single or multiple antennas and are interconnected to the CC through a fronthaul link. The presence of a large number of distributed APs enables efficient data transfer for a smaller number of UEs within the system. Cell-free massive MIMO typically encompasses two variants: canonical cell-free massive MIMO and scalable cell-free massive MIMO. In canonical cell-free massive MIMO, all APs simultaneously communicate with all UEs, while in scalable cell-free

massive MIMO, each AP serves only a subset of UEs within its coverage area. Scalable cell-free massive MIMO addresses the scalability challenges associated with the canonical approach. In canonical cell-free Massive MIMO, as the number of APs increases, the overhead for channel estimation and centralized processing becomes significant. Scalable cell-free Massive MIMO aims to overcome these limitations.

For canonical cell-free massive MIMO, during the UL transmission, the APs receive signals from UEs and forward them to the CC through the fronthaul link. The CC employs joint decoding techniques to process the received signals. Conversely, the CC forwards transmission coefficients and DL data to all APs for DL transmission [8]. In the scalable variant of cell-free massive MIMO, APs typically do not forward the received pilots from UE to the CC and take care of the signal processing locally. Although it can still use centralized precoding, and the system performance is better in that case as well, local precoding schemes are typically employed in the scalable variant [10]. This approach eliminates the need for centralized coordination and significantly reduces the fronthaul load.

The advantages of cell-free massive MIMO over small cell systems are numerous. Firstly, it provides improved uniformity of coverage for UEs. Secondly, it exhibits higher energy efficiency, thus enabling economical resource utilization. Thirdly, it achieves higher SE, allowing for increased data rates and system capacity [8]. Additionally, cell-free massive MIMO offers flexibility and ease of deployment, as it reduces the need for cell planning and configuration. Moreover, it provides a more uniform quality of service for UEs, irrespective of their location within the coverage area. Consequently, cell-free massive MIMO has been recognized as a promising technology for 5G wireless systems [9].

Despite its potential, cell-free massive MIMO presents several research challenges that require attention. One such challenge is handling mobile users because APs in a cell-free scenario have smaller coverage, resulting in frequent changes in associated AP. Another research area is making the network scalable, which requires determining the optimal AP selection methods in a scalable cell-free massive MIMO is crucial for maximizing the sys-

tem's capacity and enhancing the overall user experience [11].

1.1.3 Heterogeneous Cellular Networks

Integrating low-power BSs within macro cellular networks in a denser manner gives rise to heterogeneous cellular networks (HetNets). In a HetNet, a combination of macrocells, remote radio heads (RRHs), and low-power nodes such as picocells and femtocells (comes under an umbrella term small cell (SC)), and relays coexist within the same frequency band. This arrangement brings transmit nodes closer to end users, improving coverage and capacity. Deployment of HetNet began for 4G cellular architecture, and it is still relevant for 5G cellular architecture and beyond. Each cell type within a HetNet may have different specifications, including transmit power, coverage area, backhaul connectivity, and responsibilities. Macro cells, designed to cover distances of a few kilometers while ensuring a minimum data rate within an acceptable delay for thousands of users, typically operate with transmission power ranging from 5-40W and have dedicated backhaul connections. On the other hand, picocells serve a smaller number of users within a range of 100m to 250m, primarily enhancing in-building cellular coverage in areas with low macro penetration. Picocells typically operate with transmit power ranging from 0.2-1W [12]. Overall, the deployment of SCs helps alleviate traffic congestion on macrocells.

HetNets are combined with spatial multiplexing to increase the throughput. Adding more BSs to a designated coverage area and increasing the number of antennas at each BS can satisfy the area throughput demands. While macro BSs can be equipped with massive MIMO antenna arrays, a smaller number of antenna elements is usually sufficient for APs within each SC, as they cover a smaller area. The introduction of HetNets enhances the performance of SCs, but it also presents challenges in terms of backhaul connectivity, power consumption, and interference management.

Efficiently addressing the backhauling challenge is crucial for HetNet deployment. Backhaul connections between the BSs and the core network must be established to enable seam-

less data transfer. The choice of backhaul transmission media, such as fiber optic cables or wireless links, depends on factors such as cost, availability, and scalability. Another important consideration is power consumption. Energy efficiency becomes critical to ensure sustainable operation with the densification of BSs in a HetNet. Optimizing power allocation, utilizing sleep mode techniques, and implementing energy-saving protocols are essential for minimizing power consumption and reducing the environmental impact.

Interference management is another significant challenge in HetNets. As multiple cells coexist within the same frequency band, interference between neighboring cells can degrade network performance. Advanced interference coordination techniques, such as coordinated multi-point transmission and reception, inter-cell interference coordination, and advanced resource allocation algorithms, are employed to mitigate interference and maximize network capacity.

Although all the above-mentioned points are applicable from the onset of 4G mobile systems, there are a couple of research problems still open for investigation for 5G and beyond. To fully exploit the benefits of a HetNet system, it is critical to investigate the user association problem, i.e., how to assign active users to the BSs such that the system-wide capacity is maximized and users' experience is enhanced. Two main methods to solve this problem are association bias and load balancing [13]. In HetNet, the typical power of SBSs is much smaller than that of MBSs; therefore, UEs tend to associate with MBS more often. However, it reduces the benefits of including SCs in the network, and therefore, an association bias is applied to increase the preferentiality of UEs in associating with the SBS. Similarly, load balancing is a technique used in HetNets to distribute traffic and user load among different APs/BSs in a balanced manner. The goal is to optimize resource utilization, enhance network capacity, and ensure a better user experience. Load balancing algorithms dynamically analyze the network conditions and decide how to distribute user traffic across available BSs. The decision-making process considers factors such as current load, channel conditions, and capacity constraints.

1.2 Machine Learning in Mobile Communications

Machine learning (ML) and wireless communications are two rapidly advancing technologies that are shaping our modern world. ML's fundamental concept revolves around empowering computers to learn and execute tasks without explicit programming guidance. This remarkable ability is achieved by training algorithms on relevant data, allowing them to acquire knowledge and improve performance over time. The convergence of ML and wireless communications holds immense potential, manifesting in two significant aspects.

The first aspect involves the application of ML techniques to optimize wireless networks. This synergy is a natural fit since wireless networks encompass numerous inferential and control tasks, often operating under dynamic or uncertain conditions. ML algorithms can efficiently analyze vast amounts of data generated by wireless networks, allowing them to adapt and optimize their operations in real-time. These networks constantly transmit data at high rates, creating a wealth of examples for ML models to learn from. By leveraging ML, wireless networks can enhance their efficiency, reliability, and adaptability, leading to improved user experiences and network performance.

The second application lies in utilizing wireless networks as ML platforms themselves. This utilization is particularly relevant in the context of emerging wireless networks that support Internet of Things (IoT) applications. These networks not only facilitate data transmission but also encompass sensing, inference, and control capabilities. With the increasing processing power of edge devices connected to wireless networks, they can serve as powerful ML platforms. By harnessing the computational capabilities of these devices, ML algorithms can be deployed at the network's edge, enabling real-time data analysis, decision-making, and intelligent control. This distributed ML paradigm reduces latency, conserves bandwidth resources, and enhances privacy by processing data locally rather than relying on centralized servers.

The combination of ML and wireless communications opens up a myriad of possibil-

ities. From optimizing network performance to enabling intelligent edge computing, this convergence creates a fertile ground for innovation and transformative applications. As ML techniques continue to advance and wireless networks evolve, we can expect even deeper integration, leading to unprecedented advancements in fields such as autonomous systems, smart cities, healthcare, and beyond. The future holds immense promise as these two technologies propel each other forward, shaping a world driven by intelligent wireless connectivity.

ML algorithms can be classified into three categories: supervised, unsupervised, and reinforcement learning (RL). Each category addresses distinct problem types and has found valuable applications within the realm of wireless communications.

Supervised learning involves teaching an algorithm the input-output relationship of a function. In the context of communications, a common application of supervised learning is the design of a receiver for a specific transmission scheme, which can be viewed as a classification task. For instance, consider a modulation scheme that maps input bits to constellation points. The receiver's objective is to classify each received noisy symbol into one of the constellation points, making it a classification problem [16, 38]. Channel estimation is another relevant example of a regression problem encountered in wireless communication systems [14–16]. Here, the goal is to estimate channel coefficients from noisy versions of known pilot signals. Traditional methods assume a known channel model and attempt to estimate its parameters.

On the other hand, unsupervised learning deals with training data lacking output values, aiming to learn functions that describe the input data. These functions can prove useful for enhancing supervised learning efficiency. Common unsupervised learning problems employed in communication systems include clustering, dimension reduction, and density estimation. Recently, autoencoders, a specific type of neural network architecture commonly used in unsupervised learning tasks, became popular. An autoencoder consists of an encoder and a decoder. The encoder part takes the input data and maps it to a lower-

dimensional representation and the decoder then reconstructs the original input data from the encoded representation. The autoencoder learns to capture the underlying structure and patterns in the data by reconstructing it accurately. As it excels in data compression and dimensionality reduction, it is primarily applied to CSI feedback-related research problems [17–19].

RL constitutes another class of machine learning problems, where the objective is to learn how to interact in an unknown environment based on feedback in the form of costs or rewards following each action [20]. RL has been applied in wireless networks since the 1990s [21, 22], notably in power optimization within the physical layer to achieve energy-efficient operation [23]. RL algorithms find utility in two types of problems that require interactions with the wireless network environment. First, a device may possess an accurate model of the environment, but determining an optimal operational policy for the device remains challenging. RL can provide a solution in such cases. Second, RL methods can be employed in wireless networking problems when the environment is known to be at least wide-sense stationary (or changing slowly enough to be treated as such), but a precise model to characterize its statistical behavior is lacking. This situation often arises when operating over unlicensed frequency bands.

1.3 Thesis Objectives and Organization

1.3.1 Motivation and Problem Statement

The evolution of mobile communications towards the 6G drives more data and bandwidth-hungry applications. Therefore, current research works on 6G attempt to develop technologies that can enhance coverage and provide a higher data rate to all the UEs. Moreover, these technologies must support UEs at the edge of the cell, as well as UEs moving at a high velocity. Considering these goals and more uniformity in service, our first work developed an SBS placement strategy utilizing stochastic geometry (StoGeo) tools to improve the coverage at cell edges [24]. We also developed the required mathematical tools to derive the

performance metrics for this SBS placement scheme.

In recent times, FDD mode of communication in massive MIMO networks has gained a lot of attention. However, one of the main challenges in FDD is the unavailability of UL-DL radio channel reciprocity. In our second work, we developed an ML algorithm for the estimation of DL channel information using UL channel information in FDD massive MIMO networks [25, 26]. In this work we developed a conditional generative adversarial network (CGAN)-based algorithm to estimate the DL channel covariance matrix (CCM) and CSI.

Cell-free massive MIMO is also considered to be a key technology for the upcoming 6G systems because it can provide a more uniform coverage than the conventional massive MIMO networks. Moreover, it can be a more energy-efficient alternative to conventional cellular architecture. However, one of the key challenges in cell-free massive MIMO is the optimal association between a UE and a subset or cluster of access points (APs) in the network. To solve this problem, we have developed a reinforcement learning (RL)-based framework for AP clustering that helps APs to learn how to select the UEs in a mobile environment [27, 28].

It was mentioned earlier that massive MIMO typically operates in TDD mode, and radio channel reciprocity is the primary reason behind it. However, for mobile UEs the radio channel reciprocity may not stand as the location of UE during UL channel estimation and DL transmission is not the same. Moreover, availability of limited UL CSI can make the DL CSI estimation and correspondingly determine efficient beamforming a challenging task. Therefore, in [29], we have developed a CGAN and long short-term memory (LSTM)-based algorithm to determine an efficient beamforming vector.

1.3.2 Thesis Outline and Contributions

This section discusses the organization of the thesis and outlines the contributions of each chapter.

- Chapter 2 reviews the necessary technical background for the thesis on StoGeo, massive MIMO, cell-free massive MIMO, and ML algorithms in cellular communications.
- In Chapter 3, an intuitive solution to improve the coverage of a HetNet is discussed. We propose a new HetNet deployment model where the locations of SBSs are correlated with those of the MBSs. We place the SBSs at the vertices of each macrocell, where the macrocells are modeled by a Poisson-Voronoi tessellation with the MBSs as seeds. Theoretical analysis of this deployment scheme is performed using the tools of StoGeo. A novel distribution is also derived for the distance between the typical user and its closest SBS. Two tractable expressions for the distance distribution between a user and its closest SBS are presented, obtained by modeling the locations of SBSs as a Poisson point process and a β -Ginibre point process (β -GPP). The β -GPP models the SBS placement more accurately as it captures the correlation between the MBSs and SBSs. The performance of the proposed model is evaluated for several values of the network parameters and our results demonstrate the improvement in coverage probability, average achievable rate, and rate coverage compared to other schemes in the literature.
- For the implementation of massive multiple-input multiple-output (MIMO) cellular systems in frequency division duplex (FDD) mode, accurate estimation of DL CSI is necessary, but full radio channel reciprocity between the UL and DL does not exist in that mode. Existing work on estimating DL CSI in FDD massive MIMO systems has considered such approaches as angle-of-arrival reciprocity, compressive sensing, using second-order channel statistics (particularly the CCM), and machine learning using deep neural networks (DNNs). Typical DNN-based approaches are unsuitable for this problem because DNNs require large datasets, and thousands of training epochs, and are susceptible to environmental variations. To overcome these shortcomings, in

Chapter 4 we develop a conditional generative adversarial network (CGAN) approach to UL-to-DL mapping of both CCMs and CSI. To apply this method, we convert the UL and DL CCMs/CSI to images and employ CGAN techniques previously applied to image translation. The normalized mean square error performance of the proposed CGAN is evaluated for several array sizes for both CCM and CSI mapping. For UL-to-DL CSI mapping, we also examine the SE performance of our CGAN-based method and the impact of pilot reuse; both simulated and measured CSI data are considered. Our results demonstrate performance improvement over existing algorithms.

- For scalable cell-free massive MIMO network the best possible personalized user-centric cluster of nearby APs should serve each user. Unfortunately, determining that cluster is a combinatorially-complex problem made even harder when the users are in motion. Therefore, in Chapter 5, we develop a multi-agent reinforcement learning (MARL) algorithm for AP selection and clustering. Each AP is an agent in the MARL algorithm, and it is trained to near-optimally select for itself which users to serve. Conventional MARL algorithms require a centralized reward system to train the agents, and the agents' neural network weights tend to depend strongly on their locations during training. To counteract these problems, we also consider a federated MARL framework. Simulation results demonstrate both our conventional and federated MARL algorithms outperform existing published AP selection algorithms and also provide performance comparable to the case of all APs serving all users. The results also show the conventional algorithm has somewhat superior performance in the environment it was trained in, but the federated algorithm transfers its learning to changed environments much better, with very little performance loss.
- As already mentioned, estimation of DL CSI is necessary for massive MIMO systems to enable precoding and in general achieve high SE. In massive MIMO TDD mode of communication is used because of the UL and DL radio channel reciprocity, and

DL CSI estimation becomes significantly easier. However, CSI estimation (both UL and DL) is challenging in an environment with highly mobile users due to rapidly varying fading. The estimation becomes even more challenging when the UL CSI knowledge is incomplete due to system constraints. In Chapter 6, we combine two machine learning techniques to tackle this twofold problem: 1) predicting DL CSI from earlier UL CSI estimates, and 2) estimating full UL CSI from its incomplete form. We apply LSTM for the first sub-problem to capture the spatio-temporal correlation between CSI at different time instances and UE positions. We use a conditional generative adversarial network (CGAN) for the second sub-problem to estimate the full UL CSI from its incomplete version. We study the normalized mean squared error performance of the proposed CGAN-LSTM method and compare the achieved SE of the system with what is maximally achievable with full CSI knowledge.

- Finally, Chapter 7 summarizes the contributions of the thesis and gives directions for future research.

Chapter 2

Background

This chapter provides the mathematical background and descriptions for different concepts and ML algorithms used in the rest of the thesis.

2.1 Stochastic Geometric Modeling of Cellular Networks

The locations of base stations and user terminals in wireless networks are often not according to some predetermined pattern, nor fixed. While base station placement is not entirely random, it has become increasingly irregular due to the deployment of small cells and pico cells. On the other hand, user terminals are typically randomly distributed and frequently change their locations. Consequently, traditional fixed models like the hexagonal grid model fail to represent the network topology adequately. This has led to the growing adoption of StoGeo-based modeling approaches within the research community [30]. These models provide a more realistic representation of network architecture and offer mathematical tractability in some cases [31]. StoGeo plays a crucial role in calculating the statistical properties of point collections and enables the computation of averages across all possible realizations.

StoGeo is a mathematical discipline that specifically focuses on analyzing random spatial patterns, particularly point patterns. A point pattern, or point process, refers to a set of points or locations generated randomly according to a specific mechanism. A countable random set $\Phi = \{x_1, x_2, \dots\} \in \mathbb{R}^2$ represents a point process. Various mathematical models exist

to describe point processes, including but not limited to the binomial point process (BPP), Poisson point process (PPP), Ginibre point process (GPP), various Poisson cluster processes (PCPs), Matérn hardcore processes, softcore processes, etc. These models allow researchers to capture and study the characteristics and behavior of point patterns in wireless networks, aiding in developing and optimizing network designs and protocols.

2.1.1 Poisson Point Process

The two key characteristics of a PPP are 1) the points' locations are considered randomly and independently of each other distributed within the space, and 2) the number of points in any region follows a Poisson distribution. This means that the probability of finding a certain number of points in a given region depends only on the intensity of points in that region. Typically, two types of PPPs are considered for network modeling: homogeneous PPP and inhomogeneous PPP. In the homogeneous PPP, the distribution of points is uniformly random across the entire space under consideration. In this model, the probability of finding a point in any given region is the same across the entire space, and the intensity of points remains constant. Homogeneous PPP is characterized by invariant statistical properties across the entire spatial domain. In contrast, an inhomogeneous PPP, also known as a non-homogeneous PPP or spatially-varying PPP, allows for non-uniform point distributions across space. In this model, the density of points can vary across different regions of the space. The intensity function describes the spatial variation, representing the expected number of points per unit area in different regions. The intensity function can be defined as a function of the spatial coordinates or other relevant factors that influence the point distribution.

The formal definition of a PPP is given in [30], which in simpler terms is the following: Φ is a PPP in \mathbb{R}^2 if and only if,

- for every bounded closed set A , the number of points $N(A)$ is Poisson distributed

with mean $\lambda(A) = \int_A \lambda(x)dx$, where $\lambda(x)$ is the intensity at location x and

$$P[N(A) = n] = \frac{(\lambda(A))^n}{n!} e^{-\lambda(A)}, \quad n = 0, 1, \dots \quad (2.1)$$

- if A_1, A_2, \dots, A_m are disjoint sets, $N(A_1), N(A_2), \dots, N(A_m)$ are independent random variables.

For a homogeneous PPP $\lambda(x)$ reduces to a constant λ that is the same throughout the entire space, whereas, in inhomogeneous PPP $\lambda(x)$ varies based on the region or coordinate.

The PPP has been extensively used to characterize the locations of wireless nodes in prior research [32–35]. The main advantage of PPP is its tractability; it provides a simple and analytically tractable framework for studying the spatial distribution of wireless nodes. The mathematical properties of the PPP allow for the derivation of closed-form expressions and efficient analysis of various system performance metrics, such as coverage probability, interference, or network capacity.

2.1.2 Poisson Cluster Processes

PCPs are an extension of the PPP that allows for clustering points in space. In the PCP, the points are organized into clusters, where each cluster consists of a random number of points distributed according to a specified cluster distribution. The centers of the clusters themselves are distributed according to a PPP. This means that the overall process exhibits spatial randomness due to the PPP and local clustering due to the clusters.

Mathematically, the PCP is defined as follows: the point process defining the centroids or parent locations of clusters follows a PPP, i.e., for any bounded region A , the number of clusters $N(A)$ in A follows a Poisson distribution. Each cluster has a random number of points distributed according to a specified cluster distribution. The points within each cluster are distributed according to a separate distribution, which may vary depending on the application. There are two commonly-used PCP types, Matérn cluster process and Thomas cluster process. In Thomas cluster process, the offspring points around the parent location

or centroid follow a Gaussian distribution (symmetric in all dimensions) [36], whereas in Matérn cluster process, offspring points are uniformly distributed within a fixed radius from the parent location [37].

The PCP provides a flexible framework for modeling spatial phenomena with both randomness and clustering characteristics. The analysis of the PCP in wireless communications involves examining the key parameters that influence network behavior. This includes deriving expressions for the intensity function, which characterizes the average density of clusters and points within the network. Understanding the intensity function is crucial for evaluating coverage, capacity, and interference in wireless networks [38, 39]. Another important aspect of PCP analysis (and of point processes in general) is the study of nearest-neighbor and contact distance distributions [37, 40]. The nearest-neighbor distance distribution describes the distribution of distances between each point and its nearest neighboring point in the same point process. It quantifies the typical spacing between adjacent points and provides insights into the spatial regularity or clustering of the pattern. In contrast, contact distance distribution is the distribution of the distance between a reference point and its nearest point in the point process, where the reference point is not from the same point process. PCP allows for investigating spatial correlations and dependencies between clusters and their associated points. This information is valuable for designing resource allocation strategies, interference management techniques, and optimizing network performance.

2.1.3 Hardcore Point Processes

A hardcore point process is a type of spatial point process that incorporates a minimum distance constraint between points, also known as the hardcore distance or exclusion radius. This minimum distance constraint introduces repulsion or inhibition between points [41]. Hardcore processes are particularly useful in modeling the locations of base stations in wireless communication systems.

Multiple variations of hardcore processes exist, but the most widely known ones are the

Matérn hardcore process type I, II, and III [42]. Matérn hardcore processes, especially type I and II, exhibit properties that make them suitable for modeling realistic cellular networks, whereas type III is typically avoided because it is much harder to analyze [43]. In Matérn hardcore process type I, points are initially placed following a PPP, and afterward, if any of the points have a neighbor within a constant distance r , then it is removed. In Matérn hardcore process type II as well, points are first distributed following a PPP; thereafter, all the points are marked by a random value following a uniform distribution on $(0, 1]$. Afterward, all the points that have a neighbor within a constant distance r and have a smaller mark are removed from the point process. Both types of hard-core processes remove all points simultaneously, meaning that even the thinned-out points can eliminate other points. If points were removed one by one and the conditions were rechecked after each removal, a denser arrangement could be achieved for both types of processes. This particular approach is sometimes known as a Matérn process of type III. However, the Matérn process of type III is significantly less tractable than the processes of type I and II [30], and thus is considered much less often.

By employing hardcore point processes like the Matérn hardcore processes, researchers can accurately model the spatial arrangement of base stations and explore the effects of their locations on network performance. Incorporating the hardcore constraint of minimum distances between points allows a more realistic representation of the network's spatial layout and interference characteristics than that of a PPP. This modeling approach assists in optimizing resource allocation, interference management, and overall system design in wireless communication systems.

2.1.4 Softcore Point Processes

A softcore point process is a type of point process that incorporates a soft constraint on point spacing, in contrast to the hard constraint enforced in a hardcore point process. Unlike the hardcore process, which strictly enforces a minimum distance between points, the

softcore process allows for some degree of proximity between points, albeit with a diminishing probability as points get closer. In a softcore point process, the probability of observing points within a certain distance from each other gradually decreases as the distance decreases. One softcore point process used in cellular network modeling is the GPP. This point process belongs to a class known as determinantal point processes because the product densities of the determinantal point process are given by the determinant of a matrix that is defined using a kernel function [30]. For GPP in \mathbb{R}^2 , the kernel function is given by $K(x, y) = \pi^{-1}e^{-(|x|^2+|y|^2)/2}e^{xy}$.

In wireless communications, a softcore point process is a useful modeling framework for capturing the spatial distribution of wireless devices, such as base stations or mobile users, considering a soft constraint on point spacing. Thus, softcore processes offer more flexibility in capturing real-world scenarios where certain proximity between points is permissible or expected. This flexibility enables more accurate modeling of interference patterns, coverage areas, and resource allocation strategies in wireless networks [44].

2.2 Massive MIMO Systems

Massive MIMO systems with on the order of hundreds of antenna elements have emerged as a crucial component in the next generation of cellular networks, addressing the escalating quality of service requirements and offering the potential for significant improvements in SE and energy efficiency (EE) through relatively simple processing techniques. The concept of massive MIMO was introduced in [45], where each BS is equipped with hundreds or more antenna elements. The utilization of massive MIMO brings forth several advantages:

- **High SE:** Massive MIMO leverages both spatial multiplexing gain and antenna array gain, resulting in substantial improvements in SE. The ability to serve multiple users simultaneously in the same time-frequency resource with minimal inter-user interference allows for efficient utilization of the available spectrum.

- **High Reliability:** The significant diversity gain achieved through massive MIMO enhances the system's reliability. By exploiting the multitude of antennas, the system can mitigate fading effects and improve overall link quality, enhancing reliability and robustness in wireless communication.
- **High EE:** Massive MIMO systems concentrate the radiated energy on specific users in the downlink (DL), resulting in higher energy efficiency. The system optimizes power allocation by directing energy towards intended users, improving energy efficiency.
- **Weak Inter-User Interference:** Massive MIMO systems benefit from the orthogonality of user channels and narrower beams. This reduces inter-user interference, allowing for better spatial separation of users and enabling higher system capacity.

The properties of channel hardening and favorable propagation phenomena play a pivotal role in the performance of massive MIMO systems. These phenomena ensure that the system's communication performance becomes less dependent on small-scale fading realizations and primarily relies on the first and second-order moments of the channels, which represent the large-scale fading characteristics [46]. Exploiting these properties offers significant advantages. First, effective channels, achieved through combining and precoding techniques, become resilient to small-scale fading, reducing the burden on resource allocation and scheduling [3]. This enables simpler signal processing algorithms and enhances the system's overall efficiency. Second, the nearly orthogonal user channel vectors in massive MIMO systems eliminate the need for complex receive combining and transmit precoding schemes to manage interference. Simple linear techniques can achieve near-optimal performance, ensuring efficient interference management and reducing system complexity [47].

An interesting phenomenon known as "favorable propagation" occurs as the number of BS antennas increases significantly. Let M be the number of antennas in the BS, K be the number of users in the area, where $M \gg K$, and the uplink channel gain vector between

BS and users i and j be \mathbf{h}_i and \mathbf{h}_j , respectively, with both $\in \mathbb{C}^{M \times 1}$. Then we have

$$\frac{\mathbf{h}_i^H \mathbf{h}_j}{\sqrt{\mathbb{E}\{\|\mathbf{h}_i\|^2\} \mathbb{E}\{\|\mathbf{h}_j\|^2\}}} \rightarrow 0 \quad \text{almost surely as } M \rightarrow \infty \quad (2.2)$$

It is important to note that the numerator in (2.2) does not go to zero as M increases; rather, the ratio of numerator to denominator goes to zero. In other words, the channel gain vectors themselves do not become orthogonal, but their directions do. The phenomenon of favorable propagation plays a crucial role in achieving near-optimal performance with simple linear processing techniques. Specifically, this property allows for efficient signal detection and interference suppression in wireless communications. A basic linear detector, such as a matched filter, can effectively mitigate noise and interference on the UL. On the DL, linear beamforming techniques like maximum ratio or zero-forcing (ZF) enable base stations (BS) to simultaneously transmit multiple data streams to multiple users without causing mutual interference. While real-world propagation channels may not always exhibit favorable propagation characteristics, research has shown that approximate forms of favorable propagation can still be achieved in certain scenarios. For instance, favorable propagation can be approximated in non-line-of-sight scenarios with rich scattering or line-of-sight scenarios with distinct user angles [48].

The phenomenon of channel hardening in wireless communications results from the significant spatial diversity achieved by employing multiple antennas [49]. In systems with channel hardening, the behavior of the channel becomes nearly deterministic, and the average gain can well approximate the instantaneous channel gain. Mathematically it is expressed as [3]

$$\frac{\|\mathbf{h}_k\|^2}{\mathbb{E}\{\|\mathbf{h}_k\|^2\}} \rightarrow 1 \quad \text{almost surely as } M \rightarrow \infty \quad (2.3)$$

This means that variations in the channel due to fading or other impairments are mitigated, and the channel's behavior becomes more predictable and stable. The presence of favorable propagation and channel hardening is advantageous as it simplifies the design and optimization of wireless communication systems, allowing for more efficient resource allocation and

transmission strategies.

Even though massive MIMO offers several advantages, it is not free from challenges, and one of them is pilot contamination. Pilot contamination occurs when the pilot signals from different UEs interfere with each other due to the limited number of orthogonal pilot sequences available. This interference can degrade the channel estimation accuracy and subsequently impact the system's performance. Typically such reduced quality of channel estimation can occur in two scenarios [50, 51]:

- *Limited pilot resources:* In a massive MIMO system, the number of UEs is typically much larger than the number of orthogonal pilot sequences available. As a result, multiple UEs may end up using the same pilot sequence simultaneously, leading to pilot contamination and resulting errors in radio channel estimation.
- *Hardware impairment:* Even if orthogonal pilot sequences are allocated to UEs, there can be unintended non-orthogonal interference caused by imperfect synchronization, residual interference, or imperfect channel estimation due to hardware impairments. This interference can degrade the channel estimation quality and hence cause effects similar to those caused by pilot contamination.

There are several methods of mitigating the problem of pilot contamination, but a detailed discussion of them is outside the scope of this thesis.

2.3 Cell-free Massive MIMO

Cell-free massive MIMO systems represent a different approach to deploying massive MIMO systems, where numerous APs are strategically placed throughout the coverage area to serve users collectively. This network architecture, which combines the principles of MIMO, coordinated transmission, and ultra-dense networks with a foundation in cloud radio access network (C-RAN), has been shown to possess superior energy efficiency compared to conventional cellular networks, provided that an appropriate power control strategy is im-

plemented [52]. In this system, the typical problems caused by inter-cell interference are significantly reduced by removing cell boundaries, resulting in a more consistent and uniform service quality across the entire coverage area. Typically, cell-free massive MIMO has two variants: canonical cell-free massive MIMO and scalable cell-free massive MIMO. In simple words, they can be differentiated as follows: all the APs in a canonical cell-free massive MIMO serve all the UEs, and in its scalable counterpart, each UE is served by a subset of APs in the coverage area. This section primarily focuses on the scalable cell-free massive MIMO as the canonical form has several practical challenges. The assumption of all APs serving all the UEs results in a huge fronthaul load, requires high computation complexity, and is poor in energy efficiency [53].

In a scalable cell-free massive MIMO, each AP is equipped with a baseband processor to carry out major signal processing tasks at the AP itself. This also enables incorporating additional APs in the network without modifying the CC [10]. A large body of existing work has been devoted to the AP selection (also known as AP clustering) problem [54–58], and recently ML algorithms are also used to solve this problem [59]. Similar to the conventional massive MIMO network, the cell-free variant also requires efficient precoding to fully realize its potential in proving high SE and EE. In the scalable variant, centralized precoding is still possible (and indeed performs the best). However, to help reduce the load on the fronthaul links, distributed local precoding is often used. A brief overview of it is provided below.

Let us consider a geographical area with L APs with K single-antenna UEs, where each AP is equipped with N antennas, where $L \times N \gg K$. The DL channel gain vector $\mathbf{g}_{k\ell}$ between user k and AP ℓ can be expanded as

$$\mathbf{g}_{k\ell} = \sqrt{\beta_{k\ell}} \mathbf{h}_{k\ell}, \quad (2.4)$$

where $\beta_{k\ell}$ is the large-scale fading channel power gain that accounts for path loss and shadow fading, and $\mathbf{h}_{k\ell} \in \mathbb{C}^{N \times 1}$ is the small-scale fading channel gain vector. The APs make an estimate $\hat{\mathbf{h}}_{k\ell}$ of the UL channels based on pilot sequences sent by the UEs. Let us assume

that UE k is served by the APs in set \mathcal{L}_k . We define an $N \times N$ binary diagonal matrix $\mathbf{D}_{k\ell}$ to indicate if UE k is associated with AP ℓ :

$$\mathbf{D}_{k\ell} = \begin{cases} \mathbf{I}_N, & \ell \in \mathcal{L}_k; \\ \mathbf{0}_{N \times N}, & \ell \notin \mathcal{L}_k. \end{cases} \quad (2.5)$$

The effective DL channel vector between AP ℓ and UE k can then be considered to be $\mathbf{h}_{k\ell}^H \mathbf{D}_{k\ell}$. Some of the commonly used local precoding schemes for the scalable cell-free massive MIMO are [10]:

- **Maximum ratio (MR) precoding:** This precoding maximizes the fraction of the transmitted power from AP ℓ that is received at the desired UE. The precoding vector is given by

$$\bar{\mathbf{w}}_{k\ell}^{\text{MR}} = \mathbf{D}_{k\ell} \hat{\mathbf{h}}_{k\ell}, \quad (2.6)$$

where $\hat{\mathbf{h}}_{k\ell}$ is the estimated small-scale fading channel gain vector based on pilot sequences sent by the UEs.

- **Local partial minimum mean square error (LP-MMSE) precoding:** The locally optimal precoding vector is given by

$$\bar{\mathbf{w}}_{k\ell}^{\text{LP-MMSE}} = p_k \left(\sum_{i \in \mathcal{L}_k} p_i \left(\hat{\mathbf{h}}_{i\ell} \hat{\mathbf{h}}_{i\ell}^H + \mathbf{C}_{i\ell} \right) + \sigma^2 \mathbf{I}_N \right)^{-1} \mathbf{D}_{k\ell} \hat{\mathbf{h}}_{k\ell}, \quad (2.7)$$

where $\mathbf{C}_{i\ell} \in \mathbb{C}^{N \times N}$ is the covariance matrix of the error between $\mathbf{h}_{k\ell}$ and $\hat{\mathbf{h}}_{k\ell}$, and σ^2 is the variance of the noise (assumed to be distributed $\sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_N)$).

A localized version of regularized zero-forcing (see Section 4.3) is also possible by removing $\mathbf{C}_{i\ell}$ from (2.7).

The degree to which channel hardening and favorable propagation are experienced in cell-free massive MIMO is smaller than with co-located massive MIMO [10]. Consequently, more involved forms of precoding (such as LP-MMSE) that more actively remove interference between UEs are generally required for good performance in the cell-free case.

2.4 Machine Learning Algorithms for Wireless Systems

One of the major features of 6G wireless systems expected to be is the inherent inclusion of machine learning (ML) algorithms. This inherent inclusion is expected to involve two crucial aspects: first, employing ML algorithms to tackle computationally intensive problems and make intelligent decisions, and second, enhancing the hardware capabilities of wireless systems to enable on-chip ML functionalities [2]. In this thesis, we primarily focus on the first aspect, and in this section, a brief overview of ML algorithms that can be useful in intelligent decision-making is presented.

2.4.1 Deep Neural Network

A deep neural network (DNN) is an artificial neural network consisting of multiple layers of interconnected nodes or neurons. DNNs are designed to learn and represent complex patterns and relationships in data by progressively extracting higher-level features from lower-level ones. Each layer of neurons in a DNN processes the input data and passes it to the next layer until a final output is produced. This is one of the most widely used ML algorithms, with applications ranging from computer vision, natural language processing, speech and audio processing to robotics and autonomous systems, healthcare systems [60].

In the context of wireless communication, DNNs can be utilized for various tasks, including signal detection, channel estimation, modulation classification, and resource allocation. They can enhance the performance of wireless systems by improving the accuracy and efficiency of these tasks, leading to enhanced spectrum utilization and higher data rates [61, 62]. They can also be useful for solving optimization problems, especially for problems that require many iterations to converge, resulting in high complexity and latency [63]. Although DNNs are widely used in wireless communication, they have some challenges too. Training a DNN requires a large dataset, which may not always be available. Moreover, the training is susceptible to high dimensionality and scalability issues [64].

2.4.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a powerful class of deep learning models that have revolutionized the field of generative modeling [65]. GANs consist of two neural networks, namely the generator and the discriminator, which are trained simultaneously in a competitive manner. The generator network learns to generate synthetic data samples that resemble the training data distribution. It takes random noise as input and transforms it into realistic-looking data samples. On the other hand, the discriminator network acts as a classifier that learns to distinguish between real and fake data samples. It is trained using a combination of real training data and generated samples from the generator. This adversarial training process drives both networks to improve their performance iteratively, generating increasingly realistic samples over time.

GANs have demonstrated remarkable success in various applications, including image synthesis, text generation, video generation, and even music composition [66]. They have enabled the creation of highly realistic images that are indistinguishable from real photographs and have pushed the boundaries of what is possible in generative modeling. Some notable GAN architectures include the original Vanilla GAN proposed by Goodfellow *et al.* in 2014 [67], as well as subsequent improvements such as Deep Convolutional GANs (DCGANs) [68], conditional GANs (CGANs) [69], and Wasserstein GANs (WGANs) [70]. Among these variants, CGAN is unique because it uses an additional input, called a conditioning variable or label. It is provided to both the generator and discriminator to serve as a guide or constraint for the generation process, enabling the GAN to generate samples that align with the provided condition. For example, in an image generation scenario, the conditioning variable could represent the class label of the image to be generated [69]. These advancements have addressed training stability and mode collapse¹ issues and improved the overall quality of generated samples. GANs have also sparked significant research and

¹Mode collapse refers to a common problem in generative models, particularly in the context of generative adversarial networks (GANs). Mode collapse occurs when the generator network in a GAN fails to capture the full diversity of the training data and instead generates only a limited set of similar samples or modes [71].

development in the field of unsupervised learning, as they can learn representations and capture underlying structures of complex data without the need for explicit labels. Their ability to generate novel and diverse samples has made them a valuable tool for data augmentation, simulation, and creative applications.

Generative Adversarial Networks (GANs) have garnered significant attention in the field of wireless communications due to their ability to generate realistic data and learn complex distributions. Some of the key applications of GANs in wireless communications are the following:

- **Channel Modeling and Simulation:** GANs can be used to generate realistic channel models for wireless communication systems. Training a GAN on real-world channel data can make it learn the underlying statistical properties and generate synthetic channel samples that closely resemble the wireless propagation environment, aiding in system design and performance evaluation [17].
- **Wireless Signal Generation:** GANs can generate realistic wireless signals, such as modulated waveforms or received signal samples, for testing and training wireless communication systems. This can be particularly useful when collecting real-world data is challenging or expensive [72].
- **Wireless Signal Detection and Classification:** GANs can be employed for signal detection and classification tasks in wireless communications. Training a GAN on labeled datasets of different wireless signals allows it to discriminate between different signal types, enabling accurate and robust signal detection and classification in challenging environments [73].
- **Channel Equalization and Denoising:** GANs can be utilized to enhance the quality of received signals in wireless communication systems. Training a GAN on pairs of noisy and clean signals allows it to denoise and equalize the received signals, improving the overall system performance [74].

- **Wireless Localization:** GANs can assist in wireless localization tasks by generating synthetic radio signal strength measurements based on known positions. By training a GAN on a dataset of signal strength measurements and corresponding locations, it can learn to generate realistic signal strength maps, aiding in accurate wireless localization [75].

2.4.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of deep learning models specifically designed for sequential data processing. Unlike feedforward neural networks, which process data inputs independently, RNNs can capture temporal dependencies and model sequential information using recurrent connections. The key characteristic of RNNs is their ability to maintain an internal memory state, also known as the hidden state, which allows them to retain information from previous time steps and incorporate it into the current computation. This recurrent structure enables RNNs to process sequences of varying lengths and capture long-term dependencies in the data [76]. RNNs are widely used in various tasks involving sequential data, such as natural language processing, speech recognition, machine translation, and time series analysis. They excel in tasks that require context understanding and sequential reasoning.

One popular variant of RNNs is the Long Short-Term Memory (LSTM) network, which addresses the vanishing gradient problem² commonly encountered in traditional RNNs [78]. LSTM networks utilize a more complex gating mechanism that allows them to selectively remember or forget information over time, enabling them to capture long-range dependencies and handle sequences with extended temporal gaps. Another notable variant is the Gated Recurrent Unit (GRU) [79], which simplifies the LSTM architecture. GRUs

²The vanishing gradient problem is a challenge that can occur during the training of RNNs, particularly those with long-term dependencies. When an RNN is trained using gradient-based optimization algorithms, such as backpropagation through time, gradients are computed by backpropagating the errors from the output layer to the initial hidden state. However, in RNNs, the gradients are successively multiplied by the weight matrices of each time step during backpropagation. If these weight matrices have eigenvalues that are less than one, the gradients computed during the backpropagation process diminish or “vanish” as they propagate back through time, making it difficult for the network to learn and update the parameters of early layers [77].

provide a trade-off between complexity and performance and have been successful in various applications. RNNs have been instrumental in advancing state of the art in natural language processing tasks, including language modeling, sentiment analysis, and machine translation [80]. They can effectively model sentence contextual dependencies and generate coherent and meaningful text. Moreover, RNNs are valuable in time series analysis and forecasting. By leveraging the sequential nature of the data, RNNs can capture patterns and trends over time.

There have been further advancements in RNNs, the use of bidirectional RNNs, which process the sequence in both forward and backward directions to capture dependencies from past and future contexts simultaneously [81]. In recent years, the introduction of attention mechanisms [82] allows the network to focus on specific parts of the input sequence. Recurrent Neural Networks (RNNs) have found various applications in wireless communications, leveraging their ability to model sequential and temporal dependencies in data. Here are some key applications of RNNs in the field of wireless communications:

- **Channel Prediction:** RNNs can be used to predict the future state of wireless channels, enabling proactive adaptation of communication parameters. By leveraging the temporal dependencies in channel measurements, RNN-based models can forecast channel variations, leading to improved link reliability and resource allocation [83].
- **Modulation Classification:** RNNs have been applied to the task of automatic modulation classification, where the goal is to identify the modulation scheme used in a received signal. By learning different modulation schemes' temporal patterns and statistical characteristics, RNN-based classifiers can achieve accurate modulation classification in wireless communication systems [84].
- **Spectrum Sensing:** RNNs can be employed for spectrum sensing in cognitive radio networks. By analyzing sequential radio spectrum measurements, RNN-based models can learn to detect the presence or absence of primary users, enabling efficient

spectrum utilization and dynamic spectrum access [85].

- **Interference Detection and Mitigation:** RNNs can assist in detecting and mitigating interference in wireless networks. RNN-based models can identify and suppress interference sources by analyzing the temporal patterns of received signals, improving signal quality and network performance [86].
- **Traffic Prediction:** RNNs have been used to predict network traffic patterns in wireless communication systems. By analyzing historical traffic data, RNN-based models can learn to forecast future traffic loads, facilitating efficient resource allocation and network planning [87].

2.4.4 Federated Learning

Federated Learning (FL) is a distributed machine learning approach that enables training models on decentralized data sources while preserving data privacy. Unlike traditional centralized machine learning, where data is collected and stored in a central server, FL allows training models directly on the local or edge devices where the data resides. The central idea behind FL is to leverage the collective intelligence of a network of devices while respecting data privacy constraints. Instead of uploading raw data to a central server, the model is sent to the edge devices, and the training process occurs locally. The devices compute model updates based on their local data and send only the updates, rather than the raw data, to the central server. The server aggregates the updates from multiple devices to improve the global model, which is then redistributed to the devices for further training iterations [88].

FL offers several advantages. First, it enables training models on a vast amount of distributed data, including data that may be inaccessible due to privacy concerns or network constraints. This allows for improved generalization and model performance. Second, FL reduces the need for data transfer, addressing bandwidth limitations and privacy concerns associated with centralized data storage. Finally, FL promotes user privacy, as the raw data remains on the local devices and is never directly exposed to the central server. This ap-

proach has gained significant attention in applications where data privacy is crucial, such as healthcare, finance, and IoT systems. It allows for collaborative learning on sensitive data without compromising individual privacy or breaching data security regulations. Several FL algorithms and frameworks have been developed to facilitate the implementation of FL systems. Notable examples include Google's Federated Averaging [89], which focuses on distributed model averaging, and TensorFlow Federated [90], an open-source framework that simplifies the development of FL models.

FL has gained significant attention in wireless communications due to its potential to address privacy concerns and enable collaborative learning in distributed networks. Here are some applications of FL in wireless communications:

- **Resource Allocation and Optimization:** FL can be used to optimize resource allocation in wireless networks, such as power control, channel allocation, and user association. By leveraging the distributed intelligence of user devices, FL algorithms can learn from local data and make joint decisions to improve network performance and efficiency [91].
- **Interference Management:** FL enables collaborative interference management in wireless networks. By sharing local interference information and learning jointly, user devices can coordinate their transmissions to mitigate interference and improve overall system capacity [92].
- **Beamforming and Antenna Selection:** User devices can enhance signal quality and increase spectral efficiency by exchanging local channel state information and jointly learning beamforming vectors and antenna selection strategies [93].
- **Spectrum Sensing and Access:** User devices can collectively sense the spectrum and learn to make intelligent decisions on channel availability, ensuring efficient spectrum utilization while minimizing interference [94].

Chapter 3

Coverage Enhancement of Massive MIMO HetNets by Correlated Placement of SBSs

3.1 Introduction

Among the goals of 5G and future 6G cellular networks are to ensure significant coverage and capacity improvement to satisfy the exponential growth of mobile data traffic. One of the key enablers for increasing network capacity is the deployment of additional low-power BSs overlaying the coverage area of higher-power BSs [2, 95]. Due to the heterogeneity of BSs, such a network is known as a HetNet. Being key components of the 5G cellular standards, HetNets and large-scale antenna arrays, also known as massive MIMO [3], have attracted considerable attention in the research community.

3.1.1 Background & Motivation

In one of the seminal articles on modeling the layout and performance of HetNets using Stochastic Geometry [32], Dhillon et al. have provided a comprehensive analysis based on the assumption that the placement of BSs within each HetNet tier follows a PPP. That paper has considered the coverage probability and average rate of a K -tier HetNet downlink with single-antenna BSs. In this HetNet modeling, the location of each BS is uniformly distributed in the plane and placed independently of any other BS (whether in the same tier or a different tier). Due

to its analytical tractability, this model is considered the baseline model for analyzing HetNets using StoGeo. However, this independent deployment strategy is not very beneficial for improving the coverage; later results have indicated a maximum gain of 5% in the coverage probability of such HetNets compared to using only a single tier [33].

To further enhance the performance of HetNets, the use of massive MIMO has been standardized for use at BSs beginning with Release 15 of the 3GPP standards [6]. The introduction of massive MIMO BSs has resulted in complicated mathematical modeling. Some earlier results on general MIMO HetNets have been obtained in [34, 35]. Dhillon *et al.* [34] have derived closed-form expressions for coverage probability and achievable rate. However, these expressions are not analytically tractable, and therefore, the authors of [96] have derived expressions for coverage probability, i.e., the probability of SINR being greater than a threshold, using Toeplitz matrix representation. Afterwards, this method has been used in several papers [97–99]. Incorporating massive MIMO BSs in a HetNet with some corresponding interference management policies has been examined in [100]; the interference co-ordination strategies bolstered the inherent interference suppression of massive MIMO. Several further follow-up articles and books have been published on the modeling and analysis of massive MIMO enabled HetNets, e.g., [13, 30, 101]. Apart from interference management, several works have also been published on user association and resource allocation strategies [102, 103].

While there has been a significant body of follow-up work, little of it has focused on the correlated placement of the SBSs and the macrocell base stations MBSs. As noted, initial work in StoGeo HetNet modeling assumed random deployment of SBSs in its analysis; however, it is more effective for the SBSs to be deployed in certain regions of a network to improve coverage and capacity [104]. In existing literature, the following approaches have been taken for placement of SBSs: a) non-uniform deployment of SBSs [105], b) correlated BS placement based on the location of the users [39, 106], and c) deployment of SBSs based on a repulsion model [99]. In [105], the authors have used a non-uniform SBS de-

ployment, where the SBSs are prohibited from being placed within a given distance from an MBS. Their results indicate that this placement scheme can result in a similar probability of coverage as compared to a uniformly-distributed SBS layout, even with 50% fewer SBSs. In [39], the authors have proposed a correlated placement of users and SBSs by modeling the locations of users as a PCP around the SBSs. However, their results suggest that this model lacks efficient utilization of the MBSs, with less than 30% of users associating with the macro tier. Similarly, in [106], the authors have examined correlated BS placement, but conditioned on covariates based on existing real-life data¹, e.g., the location of the population and the roads in an area. In [99], the authors have proposed a repulsion-based model for the placement of SBSs. We are therefore motivated to develop a new StoGeo HetNet model approach where the placement of SBSs is correlated with the placement of MBSs, rather than with users or other factors.

Hardcore or softcore processes that account for the repulsion between nodes are required to model such networks more accurately. Some of these popular softcore and hardcore models are the Matérn hard-core processes, Strauss process, and the GPP. In this chapter, we employ a special form of GPP, called a β -GPP, to develop an analytical framework for comprehensive modeling and analysis of the proposed approach. In [107], a β -GPP has been used to model a network where nodes exhibit repulsion. It has been further used to model energy-harvesting networks [108], user placements [109], wireless mutual broadcast networks [110], millimeter-wave HetNets [111], and wireless-powered ad hoc networks [112]. An analysis of interference in a single-tier wireless network modeled by determinantal point processes (of which β -GPP is a sub-class) under spatially-correlated shadowing has been performed in [113]. The authors of [99] have investigated the performance of a HetNet with inter-tier interference cancellation where the placement of MBSs has been modeled using a β -GPP.

An intuitive solution to improve the coverage in a HetNet is to place SBSs at the vertices

¹However, the methodology in [106] could also, in principle, use BS locations of one tier as a covariate for another tier.

of each macrocell, where the MBSs (whose locations are modeled by a PPP) act as seeds for a Poisson-Voronoi tessellation. Each Voronoi cell of the tessellation therefore models the macrocell for the MBS acting as its seed. The vertices represent the locations of the worst interference, where the same average signal power is received from the three closest MBSs. In our scheme, users within multiple macrocells can be served by SBSs instead. Thus, this approach can potentially improve the network coverage significantly in a cost-effective manner. We note that deployment of SBSs at cell edges was suggested earlier by the 3GPP standardization body as far back as Release 9 [114]. However, to the best of our knowledge, the first step towards a mathematical model for analysis of this HetNet architecture has first been presented in our conference paper [24]. In [24], we provided different distance distributions for the proposed model, i.e., the distribution of the distance between the typical user and its nearest SBS and the distance between the MBS and SBS that are closest to the typical user. Herein, we now assume that the two tiers cause interference with each other, whereas for simplicity in [24], it was assumed that the tiers were non-interfering. Additionally, this chapter details the derivation of analytical expressions for performance measures of the proposed HetNet model, which were absent in [24]. Specifically, using both our earlier distance distributions and both PPP-based and β -GPP-based approximations of the distribution of the distance between the typical user and its nearest SBS, we derive analytical expressions for the user association probability, coverage probability, and rate coverage. Although the authors of [111] have considered a HetNet where the placement of SBSs is modeled using a β -GPP, the main problem with these point processes and their analyses is the limited analytical tractability, which makes it difficult to analyze the properties of these repulsive point processes, as well as gaining insights from the derived mathematical expressions. Therefore, in this work we develop methods to derive closed or semi-closed form expressions for β -GPP-based network models. We also provide simulation and numerical results for these performance measures and compare them with other existing HetNet models.

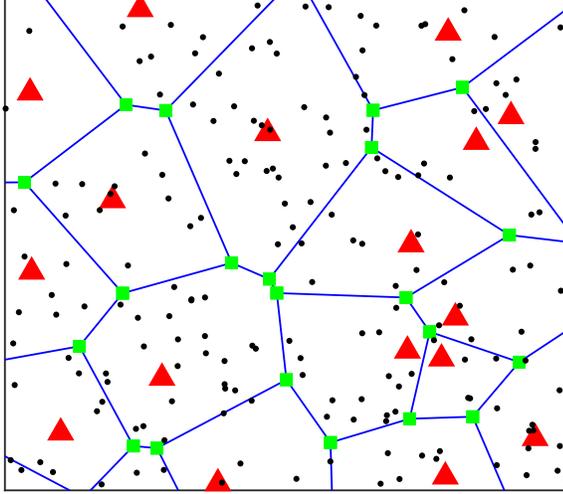


Figure 3.1: Illustration of the proposed two-tier HetNet, where MBSs (illustrated as red triangles) are located according to a PPP and SBSs (illustrated as green squares) are placed at the vertices of the Poisson-Voronoi tessellation (i.e., corners of the macrocells). The users are illustrated as black dots.

Organization: This chapter is organized as follows. We describe the StoGeo-based system model in Section 3.2, where we also define the user association policy and the channel model. Based on the system model, we derive the relevant empirical distance distributions in Section 3.3. Afterward, for the PPP and β -GPP approximations, we derive expressions for the tier association probability and conditional distance distributions in Section 3.4, and for the coverage probability and rate coverage in Section 3.5. A performance evaluation of the proposed HetNet and discussions based on the derived expressions and simulation results are given in Section 3.6. Finally, we conclude this paper in Section 3.7.

3.2 System Model

We consider TDD downlink transmission in a two-tier HetNet, which consists of MBSs and SBSs, as shown in Fig. 3.1. The set of BS tiers can be denoted as $\mathcal{J} = \{1, 2\}$. We assume without loss of generality that tier 1 represents MBSs. The MBSs are placed following Φ_1 , a homogeneous PPP with intensity λ_M . The points of Φ_1 act as seeds for a Poisson-Voronoi tessellation that represents the macrocells. The SBSs are located at the vertices of the macrocells, with the set of SBSs denoted by Φ_2 . Single-antenna users are as-

sumed to be located independently and uniformly randomly over the plane. Massive MIMO [3] is deployed in the macrocells, where each MBS is equipped with N_M antennas and simultaneously transmits to U users ($U \ll N_M$) within one time-frequency resource. Each MBS is assumed to use zero-forcing beamforming (ZFBF) to transmit U data streams, with equal power allocated to each of the streams/users. Such a transmission scheme has been widely used in the existing multiuser MIMO work such as [115, 116]. Each SBS has only one antenna and thus serves one user per time-frequency resource. We also assume perfect downlink channel state information is known at each BS for all its users. The MBS tier can interfere with transmissions of the SBS tier, and vice versa. This contrasts with our conference paper [24], where we assumed the tiers were mutually non-interfering.

3.2.1 User Association

We consider a biased maximum power user association strategy with open access, in which a user will be associated with the BS that provides the largest biased average received power for data transmissions, and users are allowed to be associated with any BS. The average power a user receives from BS b , where $b \in \{\Phi_1 \cup \Phi_2\}$, is

$$P_{r,b} = \mathbb{E}_{h_b} [h_b(P_b/U_b) ||d_b||^{-\alpha_b}], \quad (3.1)$$

where the channel power gain and transmit power for BS b are denoted by h_b and P_b , respectively. The distance between the user and BS b is denoted by d_b , and the path loss exponent is given by α_b (> 2). The number of users simultaneously communicating with BS b is denoted by U_b ; $U_b = U$ for MBSs and $U_b = 1$ for SBSs. It is trivial to see that, if the distribution for h_b and the values of P_b , U_b , and α_b are the same for all BSs in tier j , then if a user associates with tier j , it will be with the closest BS of that tier.

To compensate for the differences in antennas and transmit powers between the macro and small-cell tiers, as well as allowing for offloading traffic between tiers, a tier-dependent bias factor B_j is incorporated [117]. Thus, the association strategy of the typical user can

be expressed as

$$j^* = \arg \max_j B_j \mathbb{E}\{h_j\} (P_j/U_j) \|d_j\|^{-\alpha_j}, \quad j \in \{1, 2\}, \quad (3.2)$$

where d_j is the distance to the nearest BS in tier j , and h_j is the channel power gain for that BS. We specify without loss of generality that bias factor $B_1 = 1$ and $B_2 = B$.

3.2.2 Channel Model

We assume that the channels between any individual antenna at a BS and the user are independent and identically distributed (i.i.d.), are quasi-static, and undergo Rayleigh fading. As mentioned earlier, each MBS employs ZFBF precoding to simultaneously communicate with U users using N_M antennas over the same time interval and frequency band. Therefore, the small-scale fading power gain between any MBS and the users it serves can be modeled as a gamma distribution with shape parameter $N_M - U + 1$ and scale parameter 1, i.e., $h_1 \sim \Gamma(N_M - U + 1, 1)$ [118–120]. If the MBS is instead an interfering BS for a user in another macrocell, the small-scale fading power gain is instead distributed as $h_1 \sim \Gamma(U, 1)$ [120]. Similarly, the small-scale fading power gain between any SBS and any user follows an exponential distribution, i.e., $h_2 \sim \exp(1)$ [32, 120]. We also note that the exponential distribution can be expressed as a special case of the gamma distribution, such that $h_2 \sim \Gamma(1, 1)$.

3.2.3 Signal-to-Interference-plus-Noise Ratio

The received SINR for a typical user ‘ o ’ when served by MBS k is given by

$$\gamma_1 = \frac{\frac{P_1}{U} h_{o,k} \|d_{o,k}\|^{-\alpha_1}}{I_1 + \sigma_n^2}. \quad (3.3)$$

I_1 is the interference from the remaining MBSs $\{\Phi_1 \setminus k\}$ and all SBSs Φ_2 , and $h_{o,i}$ ($h_{o,k}$) is the small-scale fading power gain between the typical user and BS i (k); $h_{o,i} \sim \Gamma(U_i, 1)$ and $h_{o,k} \sim \Gamma(N_M - U + 1, 1)$ [120]. $d_{o,k}$ and $d_{o,i}$ are the distances from the typical user to its associated MBS k and to an interfering BS i , respectively. P_1 and σ_n^2 are the transmit power of an MBS and the noise power, respectively.

Similarly, the received SINR when the typical user ‘ o ’ is served by SBS ℓ is given by

$$\gamma_2 = \frac{P_2 h_{o,\ell} \|d_{o,\ell}\|^{-\alpha_2}}{I_2 + \sigma_n^2}, \quad (3.4)$$

where I_2 is the interference from the remaining SBSs $\{\Phi_2 \setminus \ell\}$ and all MBSs Φ_1 , P_2 is the transmit power of an SBS, and the small-scale fading power gain between the typical user and SBS y is $h_{o,y} \sim \exp(1)$ [32, 120].

The expressions for interference when the typical user is associated with MBS k or SBS ℓ are respectively given by

$$I_1 = \sum_{j=1}^2 \sum_{x \in \Phi_j \setminus k} \frac{P_j}{U_j} h_{o,x} \|d_{o,x}\|^{-\alpha_j}; \quad (3.5)$$

$$I_2 = \sum_{j=1}^2 \sum_{x \in \Phi_j \setminus \ell} \frac{P_j}{U_j} h_{o,x} \|d_{o,x}\|^{-\alpha_j}, \quad (3.6)$$

where $U_1 = U$, and $U_2 = 1$.

3.3 Distance Distributions

From the above equations, it is evident that one of the most important variables is the distance between a user and the serving BS. The distribution of the random variable (RV) representing the distance between a user and MBS is available in the literature. However, the RV for the distance between a user and its nearest SBS, where SBSs are placed at the macrocell vertices, has not been analyzed prior to our work in [24].

To derive the distribution of the distance between a user and the nearest SBS, denoted by R_{oS} , we use two other RVs, R_{oM} and R_{MS} . R_{oM} denotes the distance between the MBS and the typical user within its cell, and R_{MS} denotes the distance from the MBS to the SBS (located at a macrocell vertex) that is nearest to the typical user. These distances are shown in Fig. 3.2. It is known that R_{oM} is Rayleigh distributed and its probability density function (PDF) is [121]

$$f_{R_{oM}}(r_1) = 2\pi\lambda_M r_1 \exp(-\pi\lambda_M r_1^2), \quad r_1 \geq 0. \quad (3.7)$$

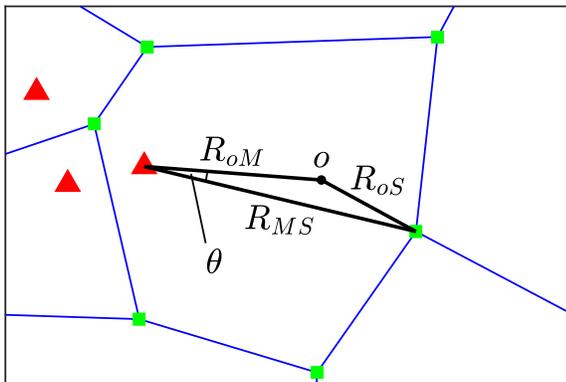


Figure 3.2: Important distances and angles in a macrocell. R_{oM} is the distance between the typical user ‘ o ’ and its nearest MBS, R_{oS} is the distance between the typical user and its nearest SBS, and R_{MS} is the distance between these two specific BSs. θ is the angle between the lines measuring R_{oM} and R_{MS} .

The PDF of the distance R_{Mv} from the MBS to a *random* vertex of its Voronoi cell is available in the literature [122]:

$$f_{R_{Mv}}(r) = 2\pi^2 \lambda_M^2 r^3 \exp(-\pi \lambda_M r^2), \quad r \geq 0. \quad (3.8)$$

It can be observed that both follow a Nakagami- m distribution with PDF

$$f(r) = \frac{2m^m r^{2m-1}}{\Omega^m \Gamma(m)} \exp\left(-\frac{mr^2}{\Omega}\right), \quad r \geq 0 \quad (3.9)$$

R_{oM} has parameters $m = 1$ and $\Omega = 1/(\pi \lambda_M)$ and R_{Mv} has parameters $m = 2$ and $\Omega = 2/(\pi \lambda_M)$. We also note that in both cases, Ω can be expressed as $m/(\pi \lambda_M)$. However, we need the distance from the MBS to the specific vertex nearest to the typical user, not to a random vertex. This condition on the variable has the effect of skewing the distances to larger values compared to considering a random vertex. The effect is similar to the “zero-cell” biasing effect on StoGeo measures [123, Ch. 9.3.3], which occurs when the specific cell containing the origin of the plane is considered rather than the “typical” cell. The reason for the skew is similar as well — the origin of the plane is statistically more likely to be located inside a larger cell than a smaller one, leading to the larger distances involved.

Proposition 1 *The distribution of R_{MS} , the distance between the MBS and the specific SBS nearest to the typical user, is closely approximated by a Nakagami- m distribution with*

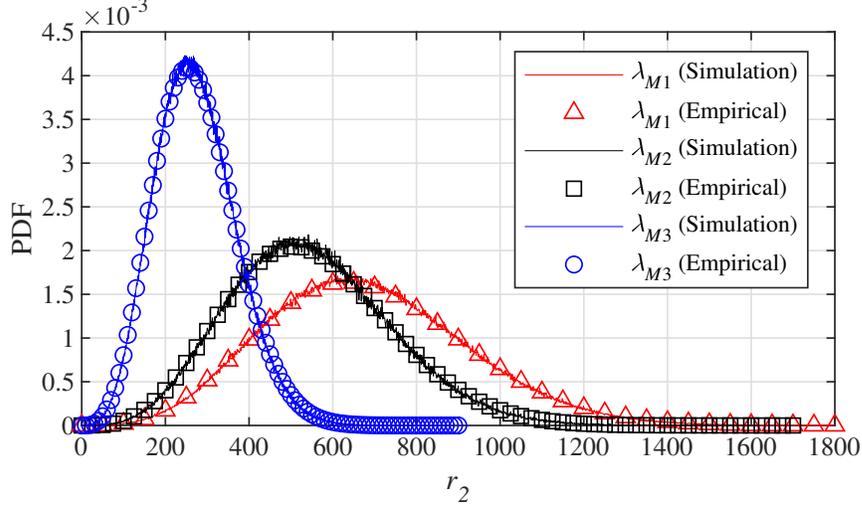


Figure 3.3: Probability density function of R_{MS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$.

empirical values of m and Ω :

$$\begin{aligned}
 f_{R_{MS}}(r_2) &\approx 2 \left(\frac{m_2}{\Omega_2} \right)^{m_2} \frac{r_2^{2m_2-1}}{\Gamma(m_2)} \exp\left(-\frac{m_2 r_2^2}{\Omega_2}\right), \quad r_2 \geq 0, \\
 &\approx 2(\pi \lambda_M)^{2.104} \frac{r_2^{3.208}}{\Gamma(2.104)} \exp(-\pi \lambda_M r_2^2), \quad r_2 \geq 0,
 \end{aligned} \tag{3.10}$$

where $m_2 = 2.104$ and $\Omega_2 = m_2/(\pi \lambda_M)$.

The empirical values of m and Ω have been verified for several values of λ_M as shown in Fig. 3.3. We have generated the theoretical results for three different values of λ_M using (3.10) and verified the PDF by simulation results, generated using those three λ_M values. As we can observe from Fig. 3.3, our proposed distribution closely follows the simulation results, with root mean squared error (RMSE) values of 1.8×10^{-5} , 3.54×10^{-5} , and 0.0013 for λ_{M1} , λ_{M2} , and λ_{M3} , respectively. Hence, the proposition is validated.

The distribution of R_{oS} is derived by utilizing the RVs R_{oM} and R_{MS} . It must be noted that R_{oM} and R_{MS} are not independent, but actually highly correlated. This is intuitive; larger distances of one should mean larger distances for the other.

Theorem 1 *The distribution of R_{oS} is given by*

$$f_{R_{oS}}(r) \approx \int_0^\infty f_{R_{oM}, R_{MS}}(r_1, r_1 + r) dr_1 + \int_r^\infty f_{R_{oM}, R_{MS}}(r_1, r_1 - r) dr_1, \quad r_1 \geq 0, r \geq 0; \quad (3.11)$$

$f_{R_{oM}, R_{MS}}(r_1, r_2)$ is the joint distribution of R_{oM} and R_{MS} :

$$\begin{aligned} f_{R_{oM}, R_{MS}}(r_1, r_2) &= 4(1 - \rho)^{m_2} \sum_{k=0}^{\infty} \left[\frac{(m_1)_k}{k!} \rho^k \right. \\ &\times \left(\frac{m_1}{\Omega_1(1 - \rho)} \right)^{m_1+k} r_1^{2(m_1+k)-1} \frac{\exp(-m_1 r_1^2 / [\Omega_1(1 - \rho)])}{\Gamma(m_1 + k)} \\ &\times \left(\frac{m_2}{\Omega_2(1 - \rho)} \right)^{m_2+k} r_2^{2(m_2+k)-1} \frac{\exp(-m_2 r_2^2 / [\Omega_2(1 - \rho)])}{\Gamma(m_2 + k)} \\ &\left. \times {}_1F_1\left(m_2 - m_1; m_2 + k; \frac{m_2 \rho}{\Omega_2(1 - \rho)} r_2^2\right) \right], \quad r_1 \geq 0, r_2 \geq 0, \end{aligned} \quad (3.12)$$

where $m_1 = 1$, $\Omega_1 = 1/\pi\lambda_M$, m_2 and Ω_2 are same as the values given in (3.10), and ${}_1F_1(a; b; z)$ is Kummer's confluent hypergeometric function, $(\cdot)_k$ denotes the Pochhammer symbol, and $\Gamma(\cdot)$ is the Gamma function. The correlation coefficient between R_{oM} and R_{MS} is denoted by ρ . Note that ρ is independent of λ_M because of the isotropic nature of a PPP. We have empirically determined the value of ρ to be about 0.77 by simulations; we use this value to generate the analytical results.

Proof. Details of the derivation are given in Appendix A.1. ■

The integration in Theorem 1 can be numerically computed. The accuracy of the derived PDF is examined for different values of λ_M and shown in Fig. 3.4 with RMSE values of 4.62×10^{-5} , 9.53×10^{-5} , and 2.02×10^{-4} for λ_{M1} , λ_{M2} , and λ_{M3} , respectively.

Derivation of network performance metrics (e.g., coverage probability, average achievable rate) requires the distance distribution R_{oS} , and using the distribution in Theorem 1 leads to extremely complicated expressions. Therefore, to simplify the expressions of performance metrics, we provide two alternatives to approximate the distribution of R_{oS} . In this article, we model the locations of vertices of a Poisson-Voronoi tessellation using two different point processes: a) an independent homogeneous PPP, and b) a β -GPP.

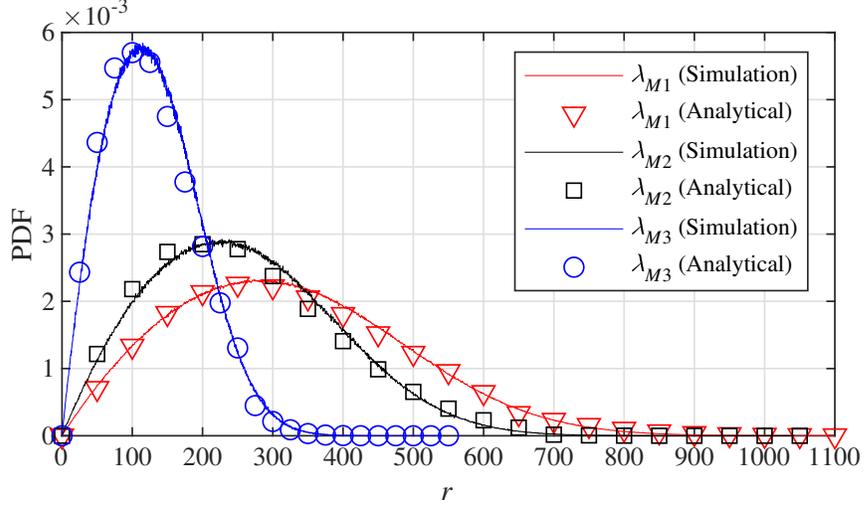


Figure 3.4: Probability density function of R_{oS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$.

a) Independent PPP modeling: In the first method we follow [124], which modeled the locations of the vertices of a Poisson-Voronoi tessellation as an independent Poisson point process with intensity

$$\lambda_S = \frac{2 \pi^{1/2} \Gamma(5/2)}{3 \Gamma^2(3/2)} \lambda_M = 2\lambda_M, \quad (3.13)$$

which is obtained by substituting $d = 2$ in [124, Eq. (7.7)].

The PDF of R_{oS} using the PPP approximation (which we denote by \tilde{R}_{oS}) can be derived by using the void probability of a PPP [34],

$$F_{\tilde{R}_{oS}}(r) = 1 - \mathbb{P}[\tilde{R}_{oS} > r] = 1 - e^{-\pi \lambda_S r^2}, \quad r \geq 0. \quad (3.14)$$

Correspondingly, the PDF is similar to the one in (3.7):

$$f_{\tilde{R}_{oS}}(r) = 2\pi \lambda_S r e^{-\pi \lambda_S r^2}, \quad r \geq 0. \quad (3.15)$$

The modeling of vertices of the Poisson-Voronoi tessellation using an independent PPP does not capture the spatial correlation between placement of MBSs and SBSs, but yields tractable expressions that suffice for a rough estimate of performance metrics. However, for more accurate modeling, we employ the second method.

b) β -GPP modeling: In recent years, it has been observed that a homogeneous PPP is not ideal to model complex BS deployments. In our proposed SBS placement strategy, there is some form of repulsion between the locations of SBSs, since they are placed on the vertices of Poisson-Voronoi tessellation; this effect cannot be modeled using a PPP. In [107] and [125], a β -GPP has been shown to be a more accurate point process than a PPP to model practical BS deployment because it models the spatial repulsion between BSs. A GPP belongs to the class of determinantal point processes and therefore it is a soft-core model. This is favorable for our scenario because, unlike hard-core point processes, there is no strict constraint on the distance between two SBSs. In some recent papers [105, 111, 126], expressions for coverage probability and rate coverage have been developed for a β -GPP model, but they are too complex to provide any meaningful insight. Therefore, in this paper we have developed simplified expressions for coverage probability and rate coverage using a β -GPP.

In this model, the placement of SBSs is defined as $\Phi_2 = \{X_i\}_{i \in \mathbb{Z}^+} \sim \text{GPP}(\lambda'_S, \beta)$. β (with $0 < \beta \leq 1$) is a repulsion parameter and λ'_S can be made set proportional to λ_M , similar to (3.13); together these parameters help capture the correlation between Φ_2 and Φ_1 . Although the locations of the Voronoi tessellation vertices are correlated with those of the seeds (and in fact the realization of the seeds completely determines the realization of the vertices), for simplicity we again model the β -GPP SBS process Φ_2 as being independent of Φ_1 . The leaves the correlation between the two to be expressed solely in the value of the parameters β and λ'_S .

Methods of simulating a GPP are discussed in [127, 128]. The squared distances from the points in a β -GPP Φ_2 to an arbitrary location have the same distribution as a set of Gamma-distributed RVs $\{Q_k\}, \forall k \in \mathbb{Z}^+$. The PDF of Q_k is

$$f_{Q_k}(x) = \frac{x^{k-1} \exp\left(-\frac{\pi\lambda'_S}{\beta}x\right)}{\left(\frac{\beta}{\pi\lambda'_S}\right)^k \Gamma(k)}, x \geq 0. \quad (3.16)$$

That is, the shape parameter of the distribution for Q_k is k and the rate parameter $\frac{\pi\lambda'_S}{\beta}$ (or

equivalently, the scale parameter is $\frac{\beta}{\pi\lambda'_S}$). The squared distance of point $x_k \in \Phi_2$ from an arbitrary location (e.g., the origin) then follows $\|x_k\|^2 = Q_k$. Lastly, each point is retained independently of the others with probability β (points not retained are removed from the realization) [107].

Note: In equations pertaining to a β -GPP, the fraction $\frac{\pi\lambda'_S}{\beta}$ appears frequently. For compactness of notation, we hereafter denote this fraction as

$$c_\beta = \frac{\pi\lambda'_S}{\beta}. \quad (3.17)$$

For a β -GPP, the distance distribution R_{oS} can be derived from the contact distribution. For a motion-invariant point process Φ like β -GPP, the contact distribution is defined as the CDF of $\|u - \Phi\|$, where u is any arbitrary location. Specifying u to be the origin o gives the CDF of R_{oS} . This distribution is [107, Eq. (19)]

$$\begin{aligned} F_{R_{oS}}(r) &= \mathbb{P}[\|o - \Phi_2\| \leq r] \\ &= 1 - \prod_{k=1}^{\infty} [1 - \beta\tilde{\gamma}(k, c_\beta r^2)], \quad r \geq 0, \end{aligned} \quad (3.18)$$

where $\tilde{\gamma}(a, x) \triangleq (\int_0^x e^{-t} t^{a-1} dt) / \Gamma(a)$ is the regularized lower incomplete Gamma function.

Next, we derive a closed-form approximation for the CDF in (3.18) using the following theorem and lemma.

Theorem 2 (Monotone Convergence Theorem [129]): *Suppose $(\mathcal{X}, \mathcal{S}, \mu)$ is a measure space, and $0 \leq f_1 \leq f_2 \leq \dots$ is an increasing sequence of \mathcal{S} -measurable functions. A function $f : \mathcal{X} \rightarrow [0, \infty]$ is defined as*

$$f(x) = \lim_{k \rightarrow \infty} f_k(x)$$

Then

$$\lim_{k \rightarrow \infty} \int f_k d\mu = \int f d\mu = \int \lim_{k \rightarrow \infty} f_k d\mu.$$

Proof. We refer the reader to [129] for more in-depth details and definitions related to the theorem, and specifically to [129, pp. 78–79] for its proof. ■

Lemma 1 *An infinite sum over $k \in \mathbb{Z}^+$ of $\tilde{\gamma}(k, x)$ simplifies as follows:*

$$\sum_{k=1}^{\infty} \tilde{\gamma}(k, x) = x. \quad (3.19)$$

Proof. Using the integral form of the regularized lower incomplete Gamma function, (3.19) can be expressed as

$$\begin{aligned} \sum_{k=1}^{\infty} \tilde{\gamma}(k, x) &= \sum_{k=1}^{\infty} \frac{1}{\Gamma(k)} \int_0^x \exp(-t) t^{k-1} dt \\ &= \lim_{k \rightarrow \infty} \sum_{n=1}^k \frac{1}{\Gamma(n)} \int_0^x \exp(-t) t^{n-1} dt \\ &\stackrel{(a)}{=} \lim_{k \rightarrow \infty} \int_0^x \exp(-t) \left(\sum_{n=1}^k \frac{t^{n-1}}{\Gamma(n)} \right) dt. \end{aligned} \quad (3.20)$$

Line (a) follows from the property that the order of finite summation and integration can be interchanged.

We next apply the monotone convergence theorem. It can be observed that each term in the sequence $f_1(t), f_2(t), \dots$, where $f_k(t) = \sum_{n=1}^k \exp(-t) \frac{t^{n-1}}{\Gamma(n)}$ is measurable in $\mathbb{R}_{\geq 0}$ (and in fact over all of \mathbb{R} , although we are only interested in non-negative numbers here). Also, the sequence converges pointwise to 1 as $k \rightarrow \infty$ for all t . We also note the property $\sum_{n=1}^{\infty} [t^{n-1}/\Gamma(n)] = \sum_{n=0}^{\infty} [t^n/\Gamma(n+1)] = \exp(t)$. Furthermore, for $t \geq 0$,

$$\begin{aligned} 0 &\leq \sum_{n=0}^{k=1} \exp(-t) \frac{t^{n-1}}{\Gamma(n)} \leq \sum_{n=0}^{k=2} \exp(-t) \frac{t^{n-1}}{\Gamma(n)} \leq \dots \\ &\Rightarrow 0 \leq f_1(t) \leq f_2(t) \leq \dots \end{aligned} \quad (3.21)$$

Since $f_k(t)$ converges to $f(t) = 1$ as $k \rightarrow \infty$, and $0 \leq f_1(t) \leq f_2(t) \leq \dots$, the conditions of the monotone convergence theorem are satisfied. Thus,

$$\begin{aligned} \sum_{k=1}^{\infty} \tilde{\gamma}(k, x) &= \lim_{k \rightarrow \infty} \int_0^x \exp(-t) \left(\sum_{n=1}^k \frac{t^{n-1}}{\Gamma(n)} \right) dt \\ &= \int_0^x f(t) dt = \int_0^x 1 dt = x, \end{aligned} \quad (3.22)$$

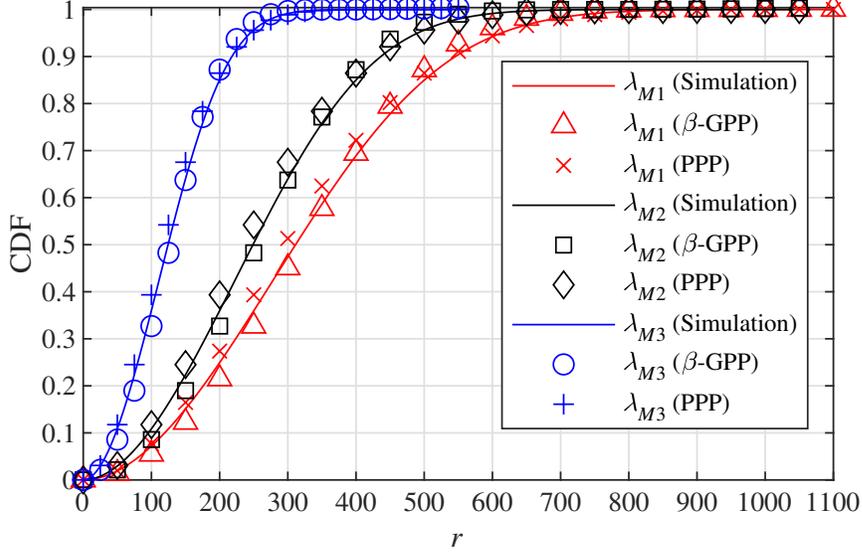


Figure 3.5: Cumulative distribution function of \tilde{R}_{oS} for different values of λ_M , with $\lambda_{M1} = 1/(\pi 500^2)$, $\lambda_{M2} = 1/(\pi 400^2)$, and $\lambda_{M3} = 1/(\pi 200^2)$.

which completes the proof. ■

Using the property proven by Lemma 1, we derive the closed-form approximation of $F_{\tilde{R}_{oS}}(r)$.

Theorem 3 A closed-form² approximation of the CDF of \tilde{R}_{oS} is given by

$$\begin{aligned}
 F_{\tilde{R}_{oS}}(r) \simeq & 1 - \left[(1 - \beta)^m \left(1 + \frac{\beta}{1-\beta} \sum_{k=1}^m \tilde{\Gamma}(k, c_\beta r^2) \right) \right] \left[\prod_{k=m+1}^{n-1} [1 - \beta \tilde{\gamma}(k, c_\beta r^2)] \right] \\
 & \times \left[1 - \pi \lambda'_S r^2 + \beta \sum_{k=1}^{n-1} \tilde{\gamma}(k, c_\beta r^2) \right], \quad r \geq 0,
 \end{aligned} \tag{3.23}$$

where $m = \left\lfloor \frac{c_\beta}{\ln(10\beta/[1-\beta])} r^2 \right\rfloor$, $n = \lceil 10c_\beta r^2 \rceil$, and $\tilde{\Gamma}(k, x)$ is the regularized upper incomplete Gamma function.

Proof. See Appendix A.2. In the event that $m = 0$, the part in square brackets in the first line of (3.23) reduces to 1. ■

The accuracy of the CDFs in (3.14) and (3.23) is shown in Fig. 3.5. For the latter, the best-fit values of $\beta = 0.95$ and $\lambda'_S = 1.378\lambda_M$ are obtained from curve fitting. The value

²In their integral forms, $\tilde{\gamma}(k, x)$ and $\tilde{\Gamma}(k, x)$ are not closed-form. However, for $k \in \mathbb{Z}^+$, $\tilde{\Gamma}(k, x)$ can alternatively be expressed as $e^{-x} \sum_{i=0}^{k-1} \frac{x^i}{i!}$ [130, Eq. 8.352.4] and $\tilde{\gamma}(k, x)$ as $1 - e^{-x} \sum_{i=0}^{k-1} \frac{x^i}{i!}$ [130, Eq. 8.352.6]. Both of these summation-based expressions are closed-form.

of β is independent of λ_M since β -GPP is isotropic. Visibly, both approximate CDFs are similar to the simulated CDF. The RMSE values of the β -GPP approximation for λ_{M1} , λ_{M2} , and λ_{M3} are 0.0149, 0.0149, and 0.0146, whereas for the PPP-based approximation, they are 0.015, 0.0157, and 0.0159, respectively.

Corollary 1 *Differentiating the approximate CDF of \tilde{R}_{oS} in (3.23), the approximate closed-form PDF of \tilde{R}_{oS} is obtained as follows, with integers m and n defined the same as for (3.23):*

$$\begin{aligned}
f_{\tilde{R}_{oS}}(r) &\simeq \frac{\partial}{\partial r} [1 - \Xi_1(r)\Xi_2(r)\Xi_3(r)] \\
&= 2\pi\lambda'_S r \Xi_2(r) \left\{ (1-\beta)^{m-1} \tilde{\Gamma}(m, c_\beta r^2) \Xi_3(r) + \tilde{\gamma}(n-1, c_\beta r^2) \Xi_1(r) \right. \\
&\quad \left. + \Xi_1(r)\Xi_3(r) e^{-c_\beta r^2} \sum_{k=m+1}^{n-1} \frac{(c_\beta r^2)^{k-1}}{\Gamma(k) [1 - \beta\tilde{\gamma}(k, c_\beta r^2)]} \right\}, \quad r \geq 0, \\
\text{where } \Xi_1(r) &= (1-\beta)^m \left[1 + \frac{\beta}{1-\beta} \sum_{k=1}^m \tilde{\Gamma}(k, c_\beta r^2) \right], \\
\Xi_2(r) &= \prod_{k=m+1}^{n-1} [1 - \beta\tilde{\gamma}(k, c_\beta r^2)], \text{ and} \\
\Xi_3(r) &= 1 - \pi\lambda'_S r^2 + \beta \sum_{k=1}^{n-1} \tilde{\gamma}(k, c_\beta r^2). \tag{3.24}
\end{aligned}$$

Proof. See Appendix A.3. We note that in the event that $m = 0$, $\tilde{\Gamma}(0, x) = 0$, so the first term within the curly brackets will disappear. ■

3.4 Tier Association Probabilities and Conditional Distance Distributions

In this section, we derive some preliminary expressions: 1) the probability of the typical user associating with either the MBS or SBS tier, and 2) the conditional distance distribution of a user from its serving BS given its serving tier is known. These expressions are required in the next section for the derivation of coverage probability and rate coverage.

3.4.1 Tier Association Probability

Expressions for the tier association probability can be derived using the PDFs/CDFs of R_{oM} and R_{oS} and the user association strategy from (3.2). First note that the mean of a Gamma-distributed random variable $h \sim \Gamma(a, 1)$ is a . The probability of the typical user associating with an MBS, denoted by A_1 , is given by

$$\begin{aligned}
A_1 &= \mathbb{E}_{R_{oM}} \left\{ \mathbb{P} \left[\mathbb{E}\{h_1\} \frac{P_1}{U} R_{oM}^{-\alpha_1} > \mathbb{E}\{h_2\} B P_2 R_{oS}^{-\alpha_2} \right] \right\} \\
&= \mathbb{E}_{R_{oM}} \left\{ \mathbb{P} \left[\frac{(N_M - U + 1)}{U} P_1 R_{oM}^{-\alpha_1} > B P_2 R_{oS}^{-\alpha_2} \right] \right\} \\
&= \mathbb{E}_{R_{oM}} \left\{ \mathbb{P} \left[R_{oS} > \left(\frac{U B P_2}{(N_M - U + 1) P_1} \right)^{\frac{1}{\alpha_2}} R_{oM}^{\frac{\alpha_1}{\alpha_2}} \right] \right\} \\
&= \mathbb{E}_{R_{oM}} \left\{ 1 - F_{R_{oS}} \left(\left[\frac{U B P_2}{(N_M - U + 1) P_1} \right]^{\frac{1}{\alpha_2}} R_{oM}^{\frac{\alpha_1}{\alpha_2}} \right) \right\} \\
&= 1 - \int_0^\infty F_{R_{oS}} \left(\left[\frac{U B P_2}{(N_M - U + 1) P_1} \right]^{\frac{1}{\alpha_2}} r^{\frac{\alpha_1}{\alpha_2}} \right) f_{R_{oM}}(r) dr. \tag{3.25}
\end{aligned}$$

The fraction in the last line of (3.25) appears as a common element in many of the equations, so for compactness, we denote

$$\zeta = \frac{U B P_2}{(N_M - U + 1) P_1}. \tag{3.26}$$

Expressions for the tier association probability for a PPP HetNet model are already available in the literature (e.g., [117]), using the corresponding PPP distance distributions:

$$A_{1,\text{PPP}} = 2\pi\lambda_M \int_0^\infty r \exp \left[-2\pi\lambda_M \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} - \pi\lambda_M r^2 \right] dr. \tag{3.27}$$

The above equation uses the substitution $\lambda_S = 2\lambda_M$ from (3.13). In the special case of $\alpha_1 = \alpha_2 = \alpha$, (3.27) reduces to

$$A_{1,\text{PPP}} = \left[1 + 2\zeta^{\frac{2}{\alpha}} \right]^{-1}. \tag{3.28}$$

For the β -GPP model, using the CDF of \tilde{R}_{oS} from (3.23), we obtain

$$\begin{aligned}
A_{1,\beta\text{-GPP}} &= 2\pi\lambda_M \int_0^\infty (1 - \beta)^m \left\{ r \exp(-\pi\lambda_M r^2) \left[1 + \frac{\beta}{1-\beta} \sum_{k=1}^m \tilde{\Gamma} \left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} \right) \right] \times \right. \\
&\quad \left. \left[\prod_{k=m+1}^{n-1} \left(1 - \beta \tilde{\gamma} \left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} \right) \right) \right] \left[1 - \pi\lambda'_S \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} + \beta \sum_{k=1}^{n-1} \tilde{\gamma} \left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} \right) \right] \right\} dr, \tag{3.29}
\end{aligned}$$

with $m = \left\lfloor \frac{c_\beta \zeta^{2/\alpha_2} r^{2\alpha_1/\alpha_2}}{\ln(10\beta/[1-\beta])} \right\rfloor$ and $n = \left\lceil 10 c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} \right\rceil$. Unfortunately, setting $\alpha_1 = \alpha_2 = \alpha$ does not yield a significant simplification in this case.

The probability of the typical user associating with an SBS, denoted by A_2 , can be found through similar steps as in (3.25), i.e., calculating $\mathbb{E}_{R_{oS}} \left\{ 1 - F_{R_{oM}} \left(\zeta^{\frac{-1}{\alpha_1}} R_{oS}^{\frac{\alpha_2}{\alpha_1}} \right) \right\}$. If A_1 is already known, A_2 may be obtained more simply by $A_2 = 1 - A_1$.

3.4.2 Conditional Distance Distribution

In this subsection, we derive the conditional distance distributions $f_{R_{oM}|j^*=1}(r)$ and $f_{R_{oS}|j^*=2}(r)$, i.e., the PDF of the distance between a user and its serving BS given its serving tier $j^* \in \{1, 2\}$ is known. Deriving these expressions is similar to the steps in the previous section. To begin, the conditional CDF is first calculated. For the MBS,

$$\begin{aligned}
F_{R_{oM}|j^*=1}(r) &= \mathbb{P}[R_{oM} < r | j^* = 1] \\
&= \frac{\mathbb{P}[R_{oM} < r, j^* = 1]}{\mathbb{P}[j^* = 1]} = \frac{\mathbb{P}[R_{oM} < r, j^* = 1]}{A_1} \\
&= \frac{1}{A_1} \int_0^r \mathbb{P}[\mathbb{E}\{h_1\} \frac{P_1}{U} r_1^{-\alpha_1} > \mathbb{E}\{h_2\} B P_2 r_2^{-\alpha_2}] \\
&\quad \times f_{R_{oM}}(r_1) dr_1 \\
&= \frac{1}{A_1} \int_0^r \left[1 - F_{R_{oS}} \left(\zeta^{\frac{1}{\alpha_2}} r_1^{\frac{\alpha_1}{\alpha_2}} \right) \right] f_{R_{oM}}(r_1) dr_1. \tag{3.30}
\end{aligned}$$

The PDF is then found by differentiating $F_{R_{oM}|j^*=1}(r)$ with respect to r :

$$\begin{aligned}
f_{R_{oM}|j^*=1}(r) &= \frac{\partial}{\partial r} F_{R_{oM}|j^*=1}(r) \\
&= \frac{1}{A_1} \left[1 - F_{R_{oS}} \left(\zeta^{\frac{1}{\alpha_2}} r^{\frac{\alpha_1}{\alpha_2}} \right) \right] f_{R_{oM}}(r). \tag{3.31}
\end{aligned}$$

Through similar steps, the conditional PDF for the SBS can be found to be:

$$f_{R_{oS}|j^*=2}(r) = \frac{1}{A_2} \left[1 - F_{R_{oM}} \left(\zeta^{\frac{-1}{\alpha_1}} r^{\frac{\alpha_2}{\alpha_1}} \right) \right] f_{R_{oS}}(r). \tag{3.32}$$

It then remains to just insert the appropriate distance distributions from earlier into (3.31) and (3.32). For the PPP model, we end up with expressions similar to related earlier works

(e.g., [117]). After substituting $\lambda_S = 2\lambda_M$, we have:

$$f_{R_{oM}|j^*=1}(r) = \frac{2\pi\lambda_M}{A_{1,PPP}} r \exp\left[-\pi\lambda_M\left(2\zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} + r^2\right)\right], r \geq 0; \quad (3.33)$$

$$f_{R_{oS}|j^*=2}(r) = \frac{4\pi\lambda_M}{A_{2,PPP}} r \exp\left[-\pi\lambda_M\left(\zeta^{-\frac{2}{\alpha_1}} r^{\frac{2\alpha_2}{\alpha_1}} + 2r^2\right)\right], r \geq 0. \quad (3.34)$$

For the β -GPP model, the conditional distance distributions are:

$$\begin{aligned} f_{R_{oM}|j^*=1}(r) &= \frac{2\pi\lambda_M(1-\beta)^m}{A_{1,\beta\text{-GPP}}} r \exp(-\pi\lambda_M r^2) \left[1 - \pi\lambda'_S \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} + \beta \sum_{k=1}^{n-1} \tilde{\gamma}\left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}}\right) \right] \\ &\times \left[1 + \frac{\beta}{1-\beta} \sum_{k=1}^m \tilde{\Gamma}\left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}}\right) \right] \left[\prod_{k=m+1}^{n-1} \left(1 - \beta\tilde{\gamma}\left(k, c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}}\right) \right) \right], r \geq 0, \end{aligned} \quad (3.35)$$

where $m = \left\lfloor \frac{c_\beta \zeta^{2/\alpha_2} r^{2\alpha_1/\alpha_2}}{\ln(10\beta/[1-\beta])} \right\rfloor$ and $n = \lceil 10 c_\beta \zeta^{\frac{2}{\alpha_2}} r^{\frac{2\alpha_1}{\alpha_2}} \rceil$;

$$\begin{aligned} f_{R_{oS}|j^*=2}(r) &= \frac{2\pi\lambda'_S}{A_{2,\beta\text{-GPP}}} r \Xi_2(r) \exp\left(-\pi\lambda_M \zeta^{\frac{-2}{\alpha_1}} r^{\frac{2\alpha_2}{\alpha_1}}\right) (1-\beta)^{m-1} \tilde{\Gamma}(m, c_\beta r^2) \Xi_3(r) \\ &+ \tilde{\gamma}(n-1, c_\beta r^2) \Xi_1(r) + \Xi_1(r) \Xi_3(r) e^{-c_\beta r^2} \sum_{k=m+1}^{n-1} \frac{(c_\beta r^2)^{k-1}}{\Gamma(k) [1 - \beta\tilde{\gamma}(k, c_\beta r^2)]} \Big\}, r \geq 0, \end{aligned} \quad (3.36)$$

where $m = \left\lfloor \frac{c_\beta}{\ln(10\beta/[1-\beta])} r^2 \right\rfloor$, $n = \lceil 10 c_\beta r^2 \rceil$, and $\Xi_1(r)$, $\Xi_2(r)$, and $\Xi_3(r)$ are as defined in (3.24).

3.5 Coverage Probability & Rate Coverage

In this section, we derive expressions for two common performance metrics for HetNets modeled by StoGeo, namely the coverage probability and rate coverage. To derive mathematical expressions for both these metrics, first the interference must be characterized. There are two typical techniques to do so: 1) using the Laplace transform of the interference distribution, and 2) using moment matching to approximate the interference as a Gamma-distributed random variable. The method using the Laplace transform unfortunately does not yield a closed-form solution for the β -GPP model, and correspondingly provides little meaningful insight about the network performance [111]. Therefore, in this article we

use the latter Gamma approximation method for both the β -GPP and PPP models of SBS placement. To the best of our knowledge, this approximation for the β -GPP model is not yet available in the literature.

3.5.1 Coverage Probability

The coverage probability, denoted by P_C , is defined as the probability that the instantaneous SINR γ_{j^*} of the typical user is greater than a threshold SINR τ_c required for coverage. Since the typical user is associated with one tier j^* at any given time, the coverage probability is

$$P_C(\tau_c) = A_1 P_{C|M}(\tau_c) + A_2 P_{C|S}(\tau_c), \quad (3.37)$$

where $P_{C|M}(\tau_c)$ and $P_{C|S}(\tau_c)$ are the conditional coverage probabilities given the user is associated with the MBS and SBS tier, respectively. The expressions for A_1 and A_2 were derived earlier in Section 3.4-3.4.1; therefore, it remains to derive expressions for $P_{C|M}(\tau_c) = \mathbb{P}[\gamma_M \geq \tau_c | j^* = 1]$ and $P_{C|S}(\tau_c) = \mathbb{P}[\gamma_S \geq \tau_c | j^* = 2]$. In both cases, we assume the network is interference-limited, and as such the noise power σ_n^2 is negligible relative to the interference power.

Deriving $P_{C|M}$ and $P_{C|S}$ takes the following general form:

$$\begin{aligned} P_{C|M(S)}(\tau_c) &\simeq \mathbb{P} \left[\frac{\frac{P_{j^*}}{U_{j^*}} h_{o,b^*} \|d_{o,b^*}\|^{-\alpha_{j^*}}}{\sum_{j=1}^2 \sum_{x \in \Phi_j \setminus b^*} \frac{P_j}{U_j} h_{o,x} \|d_{o,x}\|^{-\alpha_j}} \geq \tau_c \right] \\ &= \mathbb{E}_{r_{j^*}|j^*} \left\{ \mathbb{P} \left[\frac{P_{j^*}}{U_{j^*}} h_{o,b^*} r_{j^*}^{-\alpha_{j^*}} \geq \tau_c I_{j^*} \right] \right\}, \end{aligned} \quad (3.38)$$

where b^* represents the MBS or SBS from tier j^* that the user associates with, and $\|d_{o,b^*}\| = r_{j^*}$, which is r_1 (or R_{oM}) for $j^* = 1$ and r_2 (or R_{oS}) for $j^* = 2$. I_{j^*} is I_1 from (3.5) for $j^* = 1$ and I_2 from (3.6) for $j^* = 2$.

As shown in [131, Lemma 3], if a distribution has mean η and variance σ^2 , a Gamma distribution $\Gamma(k, \theta)$ with shape parameter $k = \eta^2/\sigma^2$ and scale parameter $\theta = \sigma^2/\eta$ will have the same first and second moments. Thus, under the method of moment matching, the

original distribution can be approximated using the Gamma distribution with these matching moments. Using this method, we characterize the interference terms I_1 and I_2 . First, we separate the contributions to the interference from the MBS and SBS tiers as follows:

$$\begin{aligned} I_1 &= \sum_{x \in \Phi_1 \setminus k} \frac{P_1}{U} h_{o,x} \|d_{o,x}\|^{-\alpha_1} + \sum_{x \in \Phi_2} P_2 h_{o,x} \|d_{o,x}\|^{-\alpha_2} \\ &= I_{1,1} + I_{1,2}. \end{aligned} \quad (3.39)$$

$$\begin{aligned} I_2 &= \sum_{x \in \Phi_1} \frac{P_1}{U} h_{o,x} \|d_{o,x}\|^{-\alpha_1} + \sum_{x \in \Phi_2 \setminus \ell} P_2 h_{o,x} \|d_{o,x}\|^{-\alpha_2} \\ &= I_{2,1} + I_{2,2}. \end{aligned} \quad (3.40)$$

Before characterizing I_1 and I_2 using a Gamma distribution, we first derive the mean and variance for each of $I_{1,1}$, $I_{1,2}$, $I_{2,1}$, and $I_{2,2}$.

Lemma 2 *The mean and variance³ of $I_{1,1}$ respectively are:*

$$\mathbb{E}\{I_{1,1}\} = \frac{2\pi\lambda_M P_1}{\alpha_1 - 2} r_1^{2-\alpha_1} \quad (3.41)$$

$$\mathbb{V}(I_{1,1}) = \frac{\pi\lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_1^{2-2\alpha_1} \quad (3.42)$$

Similarly, the mean and variance of $I_{2,1}$ respectively are:

$$\mathbb{E}\{I_{2,1}\} = \frac{2\pi\lambda_M P_1}{\alpha_1 - 2} r_{z1}^{2-\alpha_1} \quad (3.43)$$

$$\mathbb{V}(I_{2,1}) = \frac{\pi\lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_{z1}^{2-2\alpha_1} \quad (3.44)$$

where $r_{z1} = \zeta^{-1/\alpha_1} r_2^{\alpha_2/\alpha_1}$.

Proof. See Appendix A.4. ■

If the SBS tier is modeled using a PPP, then similar results are obtained for $I_{1,2}$ and $I_{2,2}$:

³We must note that (3.42) corrects an erroneous derivation in [105], which instead states the formula in (3.42) represents (their equivalent of) $\mathbb{E}\{I_{1,1}^2\}$. Calculating $\mathbb{E}\{I_{1,1}^2\}$ gives $\mathbb{E}\{\sum_{x \in \Phi_1 \setminus k} \frac{P_1^2}{U^2} h_{o,x}^2 \|d_{o,x}\|^{-2\alpha_1}\} + \mathbb{E}\{\sum_{x \in \Phi_1 \setminus k} \sum_{y \in \Phi_1 \setminus \{k,x\}} \frac{P_1^2}{U^2} h_{o,x} h_{o,y} \|d_{o,x}\|^{-\alpha_1} \|d_{o,y}\|^{-\alpha_1}\}$; [105] has missed the second part, which does not equal zero. Working out the two parts of the sum, one finds its first part equals $\mathbb{V}(I_{1,1})$ as given by (3.42), and the second part equals $(\mathbb{E}\{I_{1,1}\})^2$.

Lemma 3 For SBS placement modeled using a PPP having $\lambda_S = 2\lambda_M$, the mean and variance of $I_{1,2}$ are:

$$\mathbb{E}\{I_{1,2}\} = \frac{4\pi\lambda_M P_2}{\alpha_2 - 2} r_{\neq 2}^{2-\alpha_2} \quad (3.45)$$

$$\mathbb{V}(I_{1,2}) = \frac{4\pi\lambda_M P_2^2}{\alpha_2 - 1} r_{\neq 2}^{2-2\alpha_2} \quad (3.46)$$

where $r_{\neq 2} = \zeta^{1/\alpha_2} r_1^{\alpha_1/\alpha_2}$. Similarly, for SBS placement modeled using a PPP, the mean and variance of $I_{2,2}$ are:

$$\mathbb{E}\{I_{2,2}\} = \frac{4\pi\lambda_M P_2}{\alpha_2 - 2} r_2^{2-\alpha_2} \quad (3.47)$$

$$\mathbb{V}(I_{2,2}) = \frac{4\pi\lambda_M P_2^2}{\alpha_2 - 1} r_2^{2-2\alpha_2} \quad (3.48)$$

Proof. The proof follows much the same as for Lemma 2, so we omit it here. In this case, $h_{o,x} \sim \Gamma(1, 1)$, so $\mathbb{E}_{h_{o,x}}\{h_{o,x}\} = 1$ and $\mathbb{E}_{h_{o,x}}\{h_{o,x}^2\} = 2$. ■

On the other hand, if the SBS tier is modeled with a β -GPP, the derivation and results for the means and variances of $I_{1,2}$ and $I_{2,2}$ are more complicated.

Lemma 4 For SBS placement modeled using a β -GPP, the mean and variance of $I_{1,2}$ are:

$$\mathbb{E}\{I_{1,2}\} = \frac{2\pi\lambda'_S P_2}{\alpha_2 - 2} r_{\neq 2}^{2-\alpha_2} \quad (3.49)$$

$$\mathbb{V}(I_{1,2}) = \frac{2\pi\lambda'_S P_2^2}{\alpha_2 - 1} r_{\neq 2}^{2-2\alpha_2} - \beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_{\neq 2}^2)}{\Gamma^2(k+1)}, \quad (3.50)$$

where $r_{\neq 2} = \zeta^{1/\alpha_2} r_1^{\alpha_1/\alpha_2}$.

Proof. See Appendix A.5. We note that the variance is less than that for a PPP, as expected, because a β -GPP is repulsive. ■

Lemma 5 For SBS placement modeled using a β -GPP, the mean and variance of $I_{2,2}$ are

$$\mathbb{E}\{I_{2,2}\} = \frac{2\pi\lambda'_S P_2}{\alpha_2 - 2} r_2^{2-\alpha_2} - \beta P_2 c_\beta^{\frac{\alpha_2}{2}} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \quad (3.51)$$

$$\begin{aligned}
\mathbb{V}(I_{2,2}) &= 2\pi P_2^2 \lambda_S' \frac{r_2^{2-2\alpha_2}}{\alpha_2 - 1} - 2\beta P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\alpha_2, c_\beta r_2^2)}{\Gamma^2(k+1)} \\
&\quad - \beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} + 2\beta^2 P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \\
&\quad \times \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^3(k+1)} - \beta^2 P_2^2 c_\beta^{\alpha_2} e^{-2c_\beta r_2^2} \left(\sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \right)^2.
\end{aligned} \tag{3.52}$$

Proof. See Appendix A.6. ■

Now that we have the required means and variances, we note again that the point processes for each tier are modeled as being independent of each other. As such, measures on those point processes will also be independent. Thus, $I_{1,1}$ and $I_{1,2}$ are independent of each other, as are $I_{2,1}$ and $I_{2,2}$. Consequently, the mean and variance for I_{j^*} will just be, respectively, the sum of the means and sum of the variances for its constituent components. Thus, I_1 and I_2 can be characterized as follows.

Lemma 6 *For SBS placement modeled using a PPP having $\lambda_S = 2\lambda_M$, interference I_1 can be characterized as $I_1 \sim \Gamma(k_1, \theta_1)$, where*

$$k_1 = 4\pi\lambda_M \frac{\left[\frac{P_1}{\alpha_1-2} r_1^{2-\alpha_1} + \frac{2P_2}{\alpha_2-2} r_{\neq 2}^{2-\alpha_2} \right]^2}{\left[\frac{P_1^2(U+1)}{U(\alpha_1-1)} r_1^{2-2\alpha_1} + \frac{4P_2^2}{\alpha_2-1} r_{\neq 2}^{2-2\alpha_2} \right]}, \tag{3.53}$$

$$\theta_1 = \frac{\left[\frac{P_1^2(U+1)}{U(\alpha_1-1)} r_1^{2-2\alpha_1} + \frac{4P_2^2}{\alpha_2-1} r_{\neq 2}^{2-2\alpha_2} \right]}{2 \left[\frac{P_1}{\alpha_1-2} r_1^{2-\alpha_1} + \frac{2P_2}{\alpha_2-2} r_{\neq 2}^{2-\alpha_2} \right]}, \tag{3.54}$$

and $r_{\neq 2} = \zeta^{1/\alpha_2} r_1^{\alpha_1/\alpha_2}$. Similarly, I_2 can be characterized as $I_2 \sim \Gamma(k_2, \theta_2)$, where

$$k_2 = 4\pi\lambda_M \frac{\left[\frac{P_1}{\alpha_1-2} r_{\neq 1}^{2-\alpha_1} + \frac{2P_2}{\alpha_2-2} r_2^{2-\alpha_2} \right]^2}{\left[\frac{P_1^2(U+1)}{U(\alpha_1-1)} r_{\neq 1}^{2-2\alpha_1} + \frac{4P_2^2}{\alpha_2-1} r_2^{2-2\alpha_2} \right]}, \tag{3.55}$$

$$\theta_2 = \frac{\left[\frac{P_1^2(U+1)}{U(\alpha_1-1)} r_{\neq 1}^{2-2\alpha_1} + \frac{4P_2^2}{\alpha_2-1} r_2^{2-2\alpha_2} \right]}{2 \left[\frac{P_1}{\alpha_1-2} r_{\neq 1}^{2-\alpha_1} + \frac{2P_2}{\alpha_2-2} r_2^{2-\alpha_2} \right]}, \tag{3.56}$$

and $r_{\neq 1} = \zeta^{-1/\alpha_1} r_2^{\alpha_2/\alpha_1}$.

Proof. $\mathbb{E}\{I_1\}$ is found by adding (3.41) and (3.45), and $\mathbb{V}(I_1)$ by adding (3.42) and (3.46). Then, as per [131, Lemma 3], the Gamma distribution parameters are given by $k_1 = (\mathbb{E}\{I_1\})^2/\mathbb{V}(I_1)$ and $\theta_1 = \mathbb{V}(I_1)/\mathbb{E}\{I_1\}$. Similarly, $\mathbb{E}\{I_2\}$ is found by adding (3.43) and (3.47), and $\mathbb{V}(I_2)$ by adding (3.44) and (3.48). Then, $k_2 = (\mathbb{E}\{I_2\})^2/\mathbb{V}(I_2)$ and $\theta_2 = \mathbb{V}(I_2)/\mathbb{E}\{I_2\}$. ■

Lemma 7 For SBS placement modeled using a β -GPP, interference I_1 can be characterized as $I_1 \sim \Gamma(k_1, \theta_1)$, where

$$k_1 = \frac{4\pi^2 \left[\frac{\lambda_M P_1}{\alpha_1 - 2} r_1^{2-\alpha_1} + \frac{\lambda'_S P_2}{\alpha_2 - 2} r_{\neq 2}^{2-\alpha_2} \right]^2}{\left[\frac{\pi \lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_1^{2-2\alpha_1} + \frac{2\pi \lambda'_S P_2^2}{\alpha_2 - 1} r_{\neq 2}^{2-2\alpha_2} - \varsigma_{1,2} \right]}, \quad (3.57)$$

$$\theta_1 = \frac{\left[\frac{\pi \lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_1^{2-2\alpha_1} + \frac{2\pi \lambda'_S P_2^2}{\alpha_2 - 1} r_{\neq 2}^{2-2\alpha_2} - \varsigma_{1,2} \right]}{2\pi \left[\frac{\lambda_M P_1}{\alpha_1 - 2} r_1^{2-\alpha_1} + \frac{\lambda'_S P_2}{\alpha_2 - 2} r_{\neq 2}^{2-\alpha_2} \right]}, \quad (3.58)$$

$$\varsigma_{1,2} = \beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2\left(1 + k - \frac{\alpha_2}{2}, c_\beta r_{\neq 2}^2\right)}{\Gamma^2(k+1)}, \quad (3.59)$$

and $r_{\neq 2} = \zeta^{1/\alpha_2} r_1^{\alpha_1/\alpha_2}$. Meanwhile, I_2 can be characterized as $I_2 \sim \Gamma(k_2, \theta_2)$, where

$$k_2 = \frac{\left[\frac{2\pi \lambda_M P_1}{\alpha_1 - 2} r_{\neq 1}^{2-\alpha_1} + \frac{2\pi \lambda'_S P_2}{\alpha_2 - 2} r_2^{2-\alpha_2} - \varsigma_{2,2} \right]^2}{\left[\frac{\pi \lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_{\neq 1}^{2-2\alpha_1} + \mathbb{V}(I_{2,2}) \right]}, \quad (3.60)$$

$$\theta_2 = \frac{\left[\frac{\pi \lambda_M P_1^2 (U+1)}{U(\alpha_1 - 1)} r_{\neq 1}^{2-2\alpha_1} + \mathbb{V}(I_{2,2}) \right]}{\left[\frac{2\pi \lambda_M P_1}{\alpha_1 - 2} r_{\neq 1}^{2-\alpha_1} + \frac{2\pi \lambda'_S P_2}{\alpha_2 - 2} r_2^{2-\alpha_2} - \varsigma_{2,2} \right]}, \quad (3.61)$$

$$\varsigma_{2,2} = \beta P_2 c_\beta^{\frac{\alpha_2}{2}} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma\left(1 + k - \frac{\alpha_2}{2}, c_\beta r_2^2\right)}{\Gamma^2(k+1)}, \quad (3.62)$$

$r_{\neq 1} = \zeta^{-1/\alpha_1} r_2^{\alpha_2/\alpha_1}$, and $\mathbb{V}(I_{2,2})$ is given in (3.52).

Proof. The proof is largely the same as for Lemma 6, with (3.49)–(3.52) respectively replacing (3.45)–(3.48). ■

Now we can finally derive expressions of the conditional coverage probabilities $P_{C|M}(\tau_c)$ and $P_{C|S}(\tau_c)$.

Theorem 4 Using $I_1 \sim \Gamma(k_1, \theta_1)$, we can have the coverage probability when the user associates with the MBS tier as follows:

$$P_{C|M}(\tau_c) = \int_0^\infty \frac{\Gamma(N_M - U + 1 + k_1)}{\Gamma(k_1 + 1)\Gamma(N_M - U + 1)} \left(\frac{P_1}{\tau_c U r_1^{\alpha_1} \theta_1} \right)^{k_1} \times {}_2F_1 \left(k_1, N_M - U + 1 + k_1; k_1 + 1; \frac{-P_1}{\tau_c U r_1^{\alpha_1} \theta_1} \right) f_{R_{oM}|j^*=1}(r_1) dr_1, \quad (3.63)$$

where ${}_2F_1(a, b; c; z)$ is the Gauss hypergeometric function.

Proof. If the user associates with an MBS, then h_{o,b^*} is distributed $\sim \Gamma(N_M - U + 1, 1)$, and thus the useful signal power $\frac{P_1}{U} h_{o,b^*} r_1^{-\alpha_1}$ is distributed $\sim \Gamma(N_M - U + 1, \frac{P_1}{U r_1^{\alpha_1}})$. The probability $\mathbb{P}[\gamma_M(r_1) \geq \tau_c | j^* = 1]$ is primarily derived from [131, Proposition 11]. Next, we integrate this expression over the PDF $f_{R_{oM}|j^*=1}(r_1)$ to find the average over all possible values of r_1 . ■

Theorem 5 Using $I_2 \sim \Gamma(k_2, \theta_2)$, we can have the coverage probability when the user associates with the SBS tier as follows:

$$P_{C|S}(\tau_c) = \int_0^\infty \left(\frac{P_2}{P_2 + \tau_c r_2^{\alpha_2} \theta_2} \right)^{k_2} f_{R_{oS}|j^*=2}(r_2) dr_2. \quad (3.64)$$

Proof. The proof is similar to that of Theorem 4, where now the useful signal power is distributed $\sim \Gamma\left(1, \frac{P_2}{r_2^{\alpha_2}}\right)$. Furthermore, ${}_2F_1(k_2, k_2 + 1; k_2 + 1; -z)$ reduces to $(1 + z)^{-k_2}$, leading to the less complex expression in (3.64). ■

Substituting $P_{C|M}(\tau_c)$ and $P_{C|S}(\tau_c)$ from Theorems 4 and 5 into (3.37), we obtain the expression for coverage probability.

3.5.2 Achievable Rate & Rate Coverage

In the previous subsection, we considered the coverage probability, which is a performance metric that depends on the channel quality between a user and its serving BS and the quality of the interfering channels. In this subsection, we develop the mathematical expressions for rate coverage. Rate coverage is defined as the probability of the instantaneous achievable rate being greater than a threshold rate χ . We assume each BS allocates one resource

block with a normalized unit bandwidth to each user associated with it. The instantaneous achievable rate of the typical user is defined as

$$\mathcal{R} = \sum_{j=1}^2 A_j \mathcal{R}_j = \sum_{j=1}^2 A_j \log_2(1 + \gamma_j), \quad (3.65)$$

where \mathcal{R}_j is the user's instantaneous rate if associated with tier j , and A_j is the association probability for tier j th, and γ_j is the instantaneous SINR when the user is associated with a j th-tier BS.

Using (3.65), we also derive the expression for rate coverage, defined as

$$\mathcal{R}_C(\chi) = \sum_{j=1}^2 A_j \mathbb{P}[\mathcal{R}_j \geq \chi]. \quad (3.66)$$

Based on this definition, we can further derive

$$\begin{aligned} \mathcal{R}_C(\chi) &= \sum_{j=1}^2 A_j \mathbb{P} \left[\log_2 \left(1 + \frac{\frac{P_j}{U_j} h_{o,b^*} \|d_{o,b^*}\|^{-\alpha_j}}{I_j} \right) \geq \chi \right] \\ &= \sum_{j=1}^2 A_j \mathbb{P} \left[\frac{\frac{P_j}{U_j} h_{o,b^*} \|d_{o,b^*}\|^{-\alpha_j}}{I_j} \geq 2^\chi - 1 \right] \\ &= P_C(2^\chi - 1). \end{aligned} \quad (3.67)$$

From the above expression, the similarity between coverage probability and rate coverage can be easily observed. For coverage probability, the definition uses τ_c instead of $2^\chi - 1$. By replacing τ_c with $2^\chi - 1$ in the expressions for $P_{C|M}$ and $P_{C|S}$, we obtain the expression for \mathcal{R}_C . Therefore, we omit the remaining details of the derivation here.

3.6 Performance Evaluation

In this section, we provide insights into the network performance for our proposed SBS deployment scheme. Before presenting the results, we briefly describe the network parameters. The two-tier network is generated with MBSs placed following a homogeneous PPP with default intensity $\lambda_M = 1/(\pi \times 500^2)$ (i.e., on average one point within a circular area having a radius of 500 normalized units of distance), and SBSs are placed at the vertices of

the macrocells. For each realization of the network, the BSs are placed over a square region with side lengths of 20000 (normalized units of distance). There is additionally a smaller central square area within this larger region, with the smaller area having side lengths of $20000/\sqrt{1.5}$. (This smaller region is used to avoid edge effects for the simulation.) A user is dropped independently and uniformly randomly over the area of the interior area 10000 times. This process is repeated for a total of 5000 realizations of the BS positions, and the results are averaged over all drops and realizations. Unless stated otherwise, we assume $N_M = 100$ and $U = 10$. Other default parameter values include bandwidth $W = 10$ MHz, path loss exponent $\alpha = 3.76$ for both tiers, MBS transmit power $P_1 = 43$ dBm, SBS transmit power $P_2 = 30$ dBm, and bias factor $B = 5$ (or 7 dB). We consider thermal noise of -174 dBm/Hz and a noise figure for the UE receivers of $N_f = 14$ dB. Hence, the noise power is $\sigma_n^2 = -174 + 10 \log_{10}(W) + N_f = -90$ dBm [33, 105].

Fig. 3.6 shows the coverage probability for varying coverage threshold SINR values τ_c . The results are generated for a fixed value of the SBS tier bias factor $B = 5$. We compare the results of the proposed network layout and its PPP and β -GPP approximations with the non-uniform layout from [105] and the the PCP layout from [39]. In the case of the scheme from [105], we have considered the same exclusion radius of $R = 200$ m used as the default value in [105]. For a fair comparison, we consider $\lambda_S = 2\lambda_M$ to generate the results for [105] and [39], the same as for our PPP approximation. For the β -GPP, we use the best-fit values of $\beta = 0.95$ and $\lambda'_S = 1.378\lambda_M$ as obtained earlier at the end of Section 3.3.

From the results in Fig. 3.6(a), we observe that the coverage probability performance degrades for greater values of τ_c as expected, and that our proposed SBS deployment strategy outperforms existing SBS deployment strategies and models. The results demonstrate that for $\tau_c \geq -1$ dB, locating the SBSs at the corners of the macrocells provides around 1.13–1.21 times higher coverage probabilities than the non-uniform layout of [105], and around 1.4–1.95 times higher probabilities than the PCP scheme of [39]. We also note that although the PPP-based approximation of our SBS placement strategy results in the most tractable

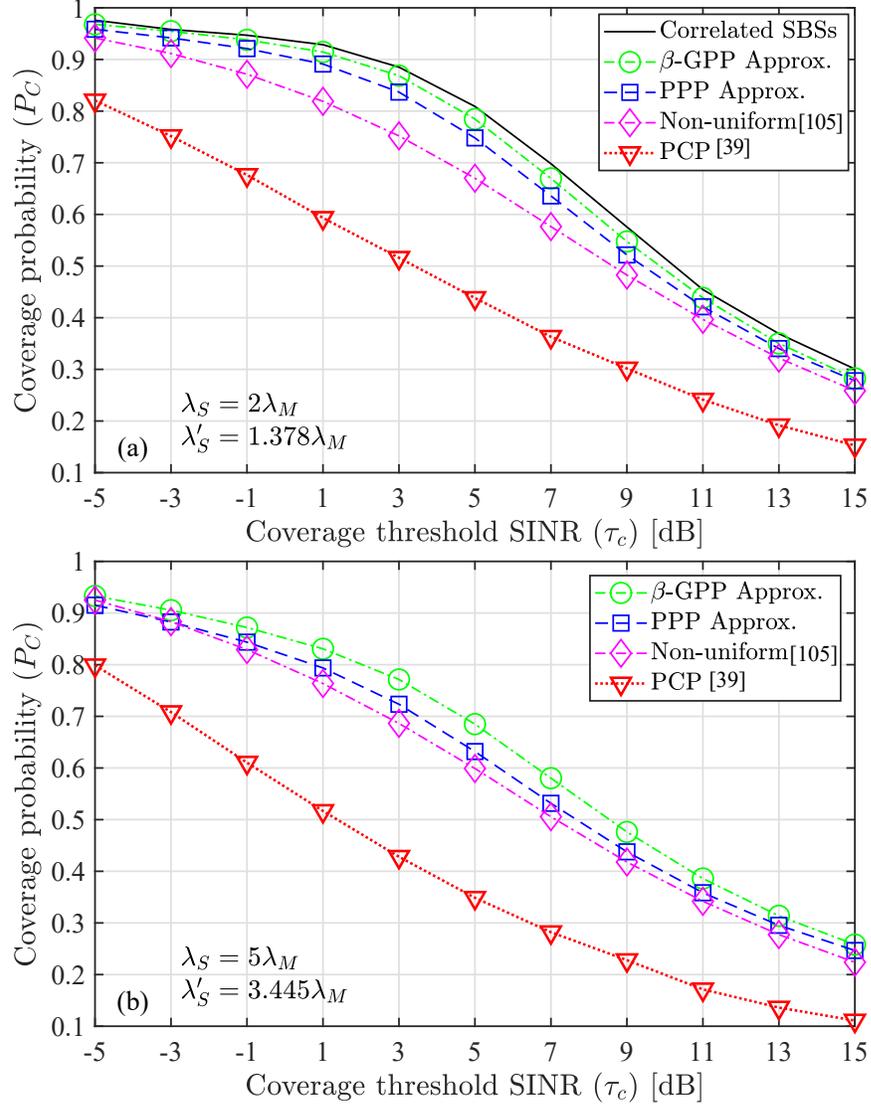


Figure 3.6: Coverage probability vs. coverage threshold SINR τ_c , comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment schemes from [105] and [39]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm. (a) $\lambda_S = 2\lambda_M$ and $\lambda'_S = 1.378\lambda_M$. (b) $\lambda_S = 5\lambda_M$ and $\lambda'_S = 3.445\lambda_M$.

expressions, the performance of the PPP-based approximation varies somewhat from the simulated results of the actual network layout. More notably, though, the β -GPP approximation is considerably better than the PPP approximation; the β -GPP coverage probability lies approximately midway between that of the PPP-based approximation and the simulated coverage probability of the “true” network layout. The correlated SBS layout has median and 90th percentile SINR threshold values of about 10.2 dB and 2.4 dB, respectively. In

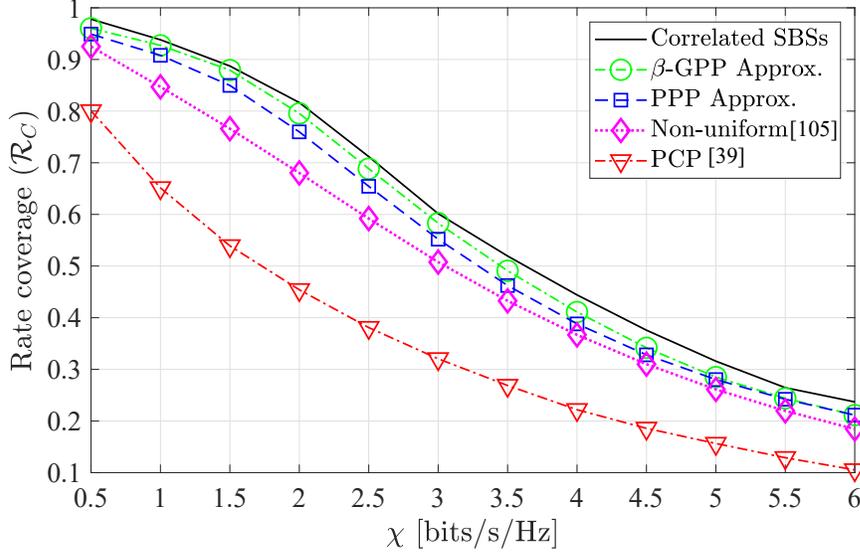


Figure 3.7: Rate coverage (complementary CDFs of rate per user), comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment schemes from [105] and [39]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm.

comparison, the β -GPP approximation provides median and 90th percentile values of about 9.8 dB and 1.7 dB, respectively (about 0.4–0.7 dB lower), whereas the PPP approximation yields median and 90th percentile values of about 9.4 dB and 0.5 dB, respectively (about 0.8–1.9 dB lower).

For the sake of interest, in Fig. 3.6(b), we have also generated coverage probability results when scaling the intensity for the SBSs of the various StoGeo models by a factor of 2.5, i.e., $\lambda_S = 5\lambda_M$ and $\lambda'_S = 3.445\lambda_M$. (We do not depict the correlated SBS scenario this time, since with its SBSs being located at the macrocell vertices, its SBS intensity cannot change independently of λ_M .) As can be seen, in all cases worse coverage probability is the result, due to the higher number of interfering SBSs. However, we must note these results would also be dependent on the association bias B . As an example, prior work [117] has shown that if the path loss exponent for both tiers is the same, there is no association bias (i.e., $B = 1$), and all BSs have a single antenna, the PPP coverage probability no longer is a function of the intensity λ of the tiers nor of the transmit powers of the BSs.

On the other hand, we have also tested the coverage probability when changing the values

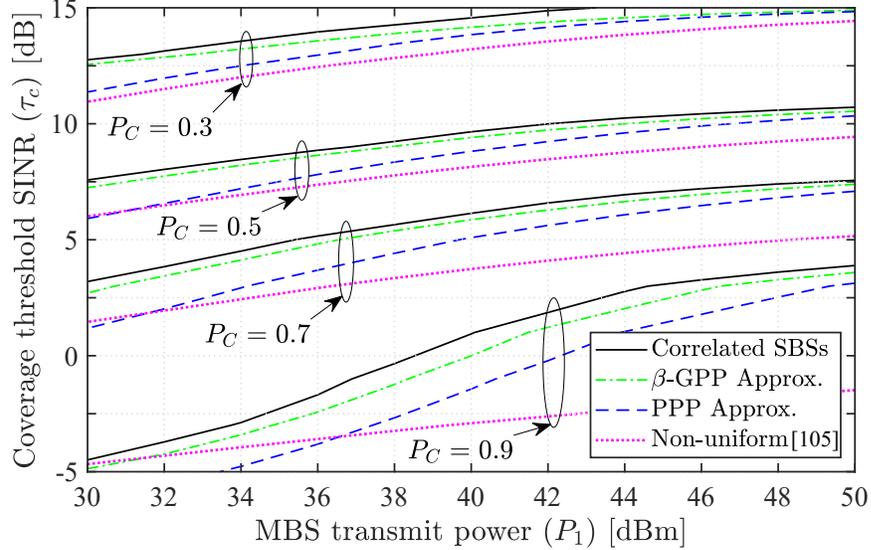


Figure 3.8: Contours of coverage probability vs. MBS transmit power P_1 and coverage threshold SINR τ_c , comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment scheme from [105]. $N_M = 100$, $U = 10$, $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_2 = 30$ dBm.

of λ_M , λ_S , and λ'_S , but keeping their ratios fixed to the same value as in Fig. 3.6(a) (i.e., $\lambda_S = 2\lambda_M$ and $\lambda'_S = 1.378\lambda_M$.) In this case, we found that the coverage probability stays the same when that ratio is constant. This is logical, as in an interference-limited system, both the useful power and interference power should change at the same rate when all distances are scaled proportionally. Thus, the SINR and coverage probabilities remain the same as well.

We provide performance results in terms of rate coverage in Fig. 3.7. The overall performance trends are similar to those for the coverage probability. The probability of a user achieving a given rate with the correlated SBS placement strategy is about 5–28% higher than with the scheme from [105] and about 1.22–2.24 times larger than with the scheme from [39]. Once again, the β -GPP approximation provides results closest to those of correlated SBS placement, followed by the PPP approximation.

We are also interested in investigating how the coverage probability changes when varying the values of other network parameters, to gain insights into the optimal values of those parameters. From this point forward, we no longer show results for the PCP scheme from

[39], as it performs significantly worse than the rest of the schemes. In Fig. 3.8, we vary the value of P_1 and τ_c while keeping the other parameters constant. The same trend of P_C decreasing with increasing τ_c can be observed as was seen before. More notably, it can be seen that P_C increases, though rather slowly, as P_1 increases. For example, the correlated SBS scheme has a coverage probability of about 0.7 (or outage probability of 0.3) with $P_1 = 30$ dBm and $\tau_c = 3.2$ dB, but to increase the coverage probability to about 0.9 (outage 0.1) with the same τ_c requires P_1 to be about 45.6 dBm. This represents a roughly 36–37 times increase in the MBS transmit power to drop the outage probability by a factor of 3. This result only pertains to keeping P_2 fixed at 30 dBm. If P_2 also changes proportionally to P_1 such that their ratio stays fixed, then we have found that P_C remains constant for a given value of τ_c , for all the schemes.

In comparing the various schemes, the correlated SBS strategy achieves any given P_C with the lowest values of P_1 and highest values of τ_c out of all the schemes, meaning that UEs are the most likely to be in coverage with that strategy. There is a general trend that, except for low values of P_C , the β -GPP approximated coverage probability tends to become closer to that of the correlated SBS strategy as P_1 increases. The same trend holds true for the PPP approximation for all values of P_C . As P_1 increases, the SINR provided by the MBS also increases relative to that provided by the SBSs, making it more likely for a UE to associate with the MBS tier. Since the MBS tier is modeled by a PPP in all three cases, it is reasonable to expect that coverage probabilities for the three schemes converge to each other as more UEs associate with the MBS tier. In contrast, the non-uniform scheme from [105] diverges from the other three cases as P_1 increases. Since SBSs are restricted from being within a certain distance of an MBS in the non-uniform scheme, it is less likely for a UE nearer to an SBS to switch over to the MBS as P_1 grows. This makes the coverage probability change considerably smaller as P_1 varies for the non-uniform scheme than for the other schemes.

Figure 3.9 provides contours of coverage probability as the association bias factor B and

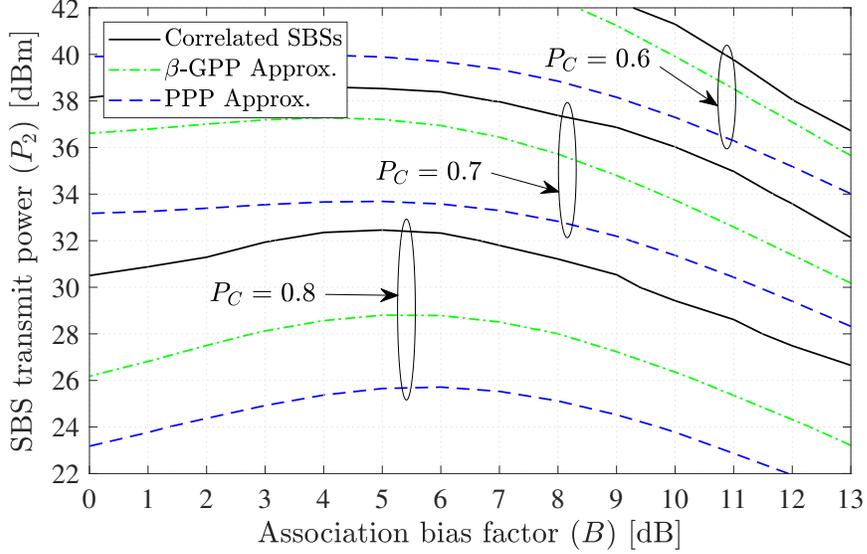


Figure 3.9: Contours of coverage probability vs. association bias factor B and SBS transmit power P_2 , comparing correlated SBS placement strategy with β -GPP and PPP approximations. $N_M = 100$, $U = 10$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $\tau_c = 5$ dB.

SBS transmit power P_2 are varied while the other parameters are constant. We also note that it is the ratio of P_1 to P_2 that determines the performance of the system; different values for the two powers with the same ratio give the same value of P_C . So, as an example, $P_1 = 43$ dBm and $P_2 = 30$ dBm would have the same performance as $P_1 = 37$ dBm and $P_2 = 24$ dBm. For this figure, we do not show results for the non-uniform scheme from [105], since that scheme tends to require significantly higher ratios of $P_1 : P_2$ to achieve a given value of P_C , especially as P_C increases. As such, the non-uniform scheme's results would lie mostly off the bottom of the graph in Fig. 3.9. The three schemes all display the same type of trend as B changes. Initially, as B increases from 1 (0 dB), the contours move to higher values of P_2 , but then eventually change direction such that P_2 decreases as B continues to increase. This means that there is a value of B for which a given coverage probability can be obtained with the highest ratio of $P_2 : P_1$, or equivalently, the lowest ratio of $P_1 : P_2$. More importantly, this value of B is optimal in that it yields the highest possible coverage probability for a given ratio of $P_1 : P_2$. For all the schemes in Fig. 3.9, this value lies at about $B = 3\text{--}6$ dB (or a factor of about 2 to 4 on a linear scale). We also observe that

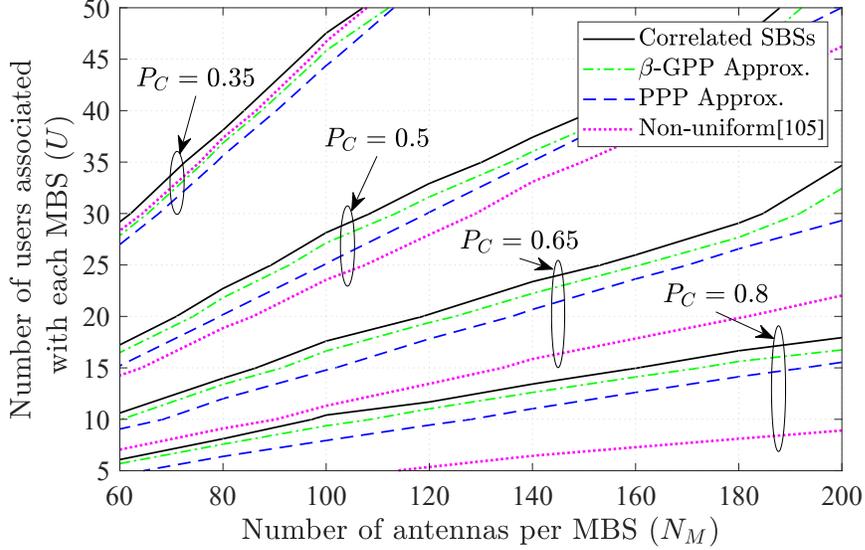


Figure 3.10: Contours of coverage probability vs. number of transmit antennas per MBS (N_M) and number of users associated with each MBS (U), comparing correlated SBS placement strategy, β -GPP and PPP approximations, and SBS deployment scheme from [105]. $B = 5$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm, $\tau_c = 5$ dB.

P_C is higher for any given values of B and P_2 with the proposed correlated SBS placement strategy than with the other two schemes, similar to previous results.

In Fig. 3.10, we depict contours of coverage probability as the number of antennas per MBS N_M and the number of users U associated with each MBS are varied while the other parameters are constant. The expected increase in P_C with increasing N_M can be observed, due to the increase in spatial diversity and hence SINR for MBS users when more MBS antennas are deployed. Similarly, as expected, P_C decreases with increasing U due to less power being allocated per MBS user and the increase in multiuser interference on SBS users. (These trends can also be predicted by the increase or decrease in the shape parameter of the Gamma-distributed channel power gains as N_M or U change.) It can also be observed that the contour lines are close to being linear for all the schemes. The implication of this is that if N_M and U are both scaled by the same factor, the coverage probability will not change by much.

We further demonstrate the advantages of the correlated SBS placement strategy in Fig. 3.11 by plotting the joint PDF of the received SINR at the user and the distance from the user to

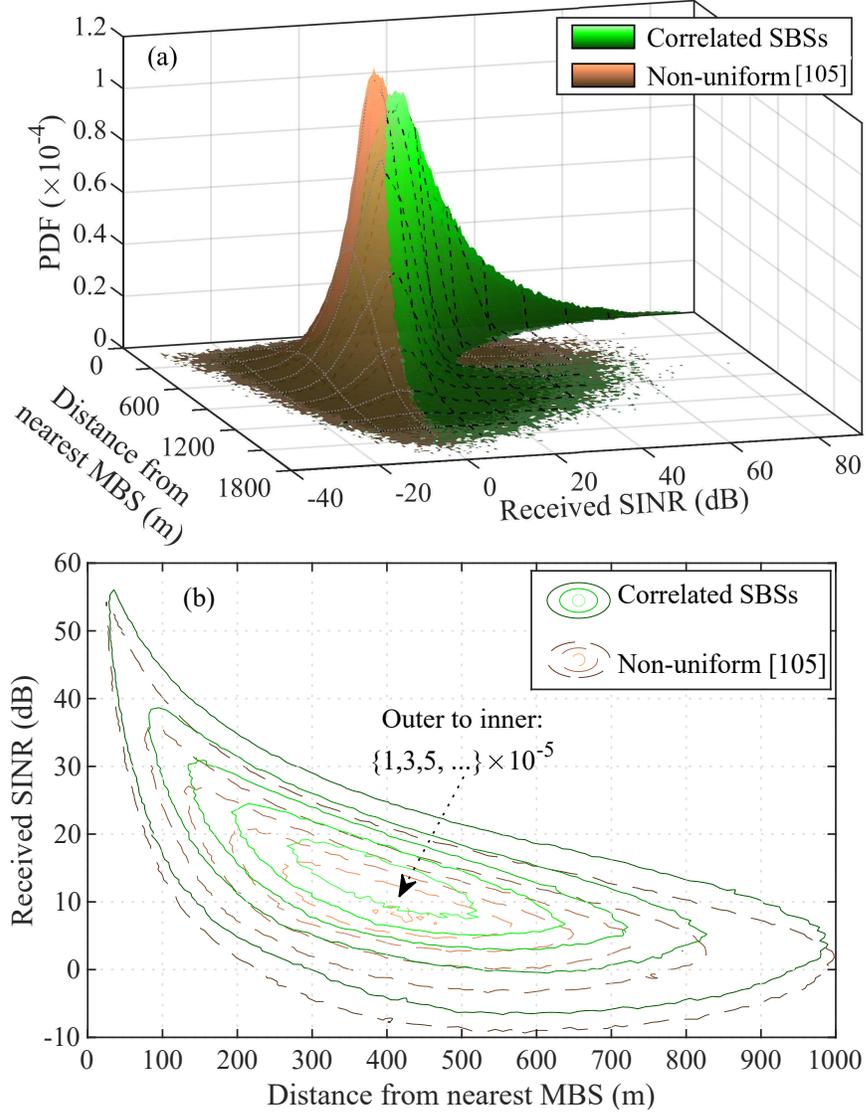


Figure 3.11: Joint PDF of received SINR at user and distance from nearest MBS, comparing correlated SBS placement strategy with scheme from [105]. $N_M = 100$, $U = 10$, $\lambda_M = 1/(\pi 500^2)$, $P_1 = 43$ dBm, $P_2 = 30$ dBm. (a) 3-dimensional PDF. (b) Contour plots showing where PDF equals $\{1, 3, 5, \dots\} \times 10^{-5}$.

its nearest MBS. As expected, the probability of a given SINR initially drops off rapidly as a user gets farther from the MBS, but eventually starts increasing again as a user approaches an SBS. Comparing the correlated SBS strategy with the non-uniform scheme from [105], it can be seen in Fig. 3.11(b) that the PDF contours for the latter scheme are shifted downwards by about 3–4 dB and a bit to the left. This implies that at any given position in the network, the probability of receiving a larger SINR is higher for the correlated SBS strategy

than with the scheme from [105]. Alternatively, in order to achieve a certain probability of a given SINR, the user needs to be located closer to either an MBS or an SBS in the scheme from [105] than in the correlated SBS strategy.

3.7 Summary

In this chapter, we have proposed a novel HetNet StoGeo model where the locations of SBSs and MBSs are correlated, with SBSs placed at the corners of the macrocells (modeled by the cells of the Poisson-Voronoi tessellation). We have developed an analytical framework for the proposed network with StoGeo tools. We have derived a new distribution for the distance between a user and its nearest SBS, approximated the placement of SBSs using two different point processes, and derived approximated distance distributions to derive simpler analytical expressions. These distance distributions are then used to analyze the downlink coverage probability of a typical user. Results suggest performance improvement up to 21% in the coverage probability and up to 28% in the rate coverage.

Chapter 4

Downlink Channel Estimation For FDD Massive MIMO Using Conditional Generative Adversarial Networks

4.1 Introduction

To fully realize the advantages of massive MIMO, the availability of DL CSI is essential [45, 132]. TDD mode of transmission is most typically considered in the research literature on massive MIMO systems, because obtaining CSI at the BS is much simpler in TDD mode than in FDD mode. Following the 3GPP standards [133], in massive MIMO field deployments, TDD mode should be used in frequency range 2 (FR2, also known as the mmWave frequency range), whereas in frequency range 1 (FR1, also commonly referred to as the sub-6 GHz frequency range), both TDD and FDD may be used. In TDD mode, the UEs can send pilot signals to the BS, which uses them to estimate the UL channels. Then, under the assumption of UL-DL radio channel reciprocity, the BS uses these estimates on the DL [3]. However, estimating the CSI in FDD mode is a challenging task, because UL-DL radio channel reciprocity does not hold due to significantly different UL and DL carrier frequencies in this mode. Conventional methods such as the use of orthogonal pilot sequences result in high overhead associated with DL training and UL feedback, along with a shortage of orthogonal sequences themselves on account of the large size of the antenna arrays [134]. Nevertheless, for the evolution of 5G cellular systems, using FDD mode for massive

MIMO is an important consideration for two reasons. First, most current cellular networks use FDD mode, and thus the infrastructure for it is readily available to network operators [135]. Second, FDD mode can provide both higher data rates and greater coverage [136].

Although full UL-DL channel reciprocity does not exist in FDD mode, it has been observed that partial channel reciprocity exists in the form of angular, shadow fading, and path loss reciprocity, because signals on the UL and DL follow the same paths (albeit in opposite directions) [137, 138]. Angular reciprocity means that the angles of arrival (AoAs) are the same for UL and DL signals. Two commonly-used algorithms to estimate the AoAs are MULTiple SInal Classification (MUSIC) [139] and Estimation of Signal Parameters using Rotational Invariance Techniques (ESPRIT) [140]. Extensive research has been conducted to exploit angular reciprocity when estimating DL CSI [141–143]. However, this approach can perform unreliably when the number of paths is large, as the AoAs for so many paths cannot be evaluated accurately [144].

4.1.1 Prior Research

Existing work on obtaining DL CSI for FDD massive MIMO can be broadly divided into two classes: a) estimation of DL CSI using DL pilots and UL feedback, and b) estimation of DL CSI using UL CSI. The first approach relies on reducing the overhead associated with DL pilot transmission and UL feedback of the estimated DL CSI. Techniques to reduce the DL pilot overhead typically utilize channel sparsity in the angular/beamspace domain [145, 146] or the low-rank nature of the channel covariance matrices (CCMs) [147–150]. Both of these properties ease the DL channel estimation process and enable estimation using fewer pilot sequences than the number of antennas at the BS. Furthermore, compressive sensing has been used to reduce the UL feedback overhead in [151–153]. However, application of these methods may be limited because channel sparsity can vary from user to user, especially in a quickly-varying environment or at sub-6 GHz frequency bands¹. Therefore, assuming

¹Beamspace sparsity would more likely be experienced at mmWave frequencies [154]. However, we note that our proposed CGAN method can be used whether or not channel sparsity exists.

the existence or prior knowledge of the channel sparsity or of the rank of the CCMs might not be justified in practice [136]. In [155], machine learning has been used to train regression models on the correlation between BS antennas. Pilots are then sent from a subset of BS antennas, and CSI is estimated for the remaining antennas from codebook-based UE feedback. Recently, multiple articles have considered reducing DL channel feedback using a DNN [156–158]. However, the dataset needed to train a DNN is very large, on the order of tens of thousands [159]. Understandably, it may be difficult to obtain such a large dataset in a practical environment, and if the environment changes then a dataset of similar size is required to retrain the DNN (although not necessarily from scratch).

In the second approach, DL CSI is estimated using UL CSI, thereby avoiding the overhead issues faced by the first approach. These methods can be further divided into two subclasses, namely “indirect” vs. “direct”. “Indirect” methods typically use second-order channel statistics, i.e., the CCM, to provide additional knowledge about the channel and thus increase the estimation accuracy [160]. The UL CSI is first used to estimate UL CCMs, map them to DL CCMs, and then estimate DL CSI using those DL CCMs. A large body of work in the literature explores this UL-to-DL CCM mapping approach. One of the earliest works was [161], in which the authors have defined a frequency calibration matrix accounting for the difference in UL/DL carrier frequencies. [162] has used a method of cubic splines to interpolate magnitude and phase values of the UL CCM to estimate the DL CCM. In [163], the authors have developed a dictionary-based interpolation algorithm that stores pairs of UL and DL CCMs measured at different UE locations and uses the dictionary to estimate the DL CCM for a new UL CCM not in the dictionary. In [164], the authors have first estimated the UL power angular spectrum (PAS), then obtained the DL PAS by resampling the Fourier transform of the UL PAS, and finally extrapolated the DL PAS to obtain the DL CCM. More recently, the method in [165] performed CCM mapping by solving an optimization problem using infinite-dimensional Hilbert spaces. Among these works on UL-to-DL CCM mapping, [161, 162, 164, 165] are somewhat model-dependent, and [163] requires

a huge dataset. Notably, to our knowledge, no existing CCM mapping method employs machine learning techniques, probably due to no direct mapping function nor parametric model being available for supervised learning to either map from UL CCMs to DL CCMs or express the output for the DL as a function of the UL input.

In comparison, more recent “direct” methods attempt to directly estimate DL CSI from UL CSI by employing neural networks. [166] and [167] have examined DL CSI extrapolation from UL CSI using a DNN, and considered both single-input single-output (SISO) and MIMO scenarios and CSI data obtained from both simulated models and field measurements. In [168–170] mapping between UL and DL CSI in space and frequency has been developed; in these approaches, transmission of DL pilots is no longer required. The authors of [171] have extracted path delay and AoA/AoD information along with using a DNN to improve the performance of DL CSI estimation. Their results indicate this method performs better than a typical DNN, but it considers a small number of clustered paths in the channels; this may not be true in the case of sub-6 GHz channels. Moreover, to reiterate, training a DNN generally requires a large dataset and thousands of training epochs [159]. Nevertheless, the above papers demonstrate that UL-to-DL CSI mapping using machine learning methods is feasible.

4.1.2 Motivation

These challenges have motivated us to employ a GAN [67], a promising machine learning framework, to solve the UL-to-DL CCM and CSI mapping problems. One of the primary advantages of a GAN over typical neural network architecture is the encoder-decoder structure, which helps it to learn common features shared between the input and output, and thereby closely estimate and generate an output having those features. (In this case, one of those common features would be angular reciprocity.) In this chapter, we use a specific type of GAN, called a CGAN used so far for image translation and generation applications [69]. To apply a CGAN in the context of wireless channels (where its use is relatively novel), we

convert the UL and DL CCMs and CSI to images and train the CGAN using these images. Thereafter, the trained CGAN framework is used to estimate the DL CCMs or CSI from the UL CCMs or CSI. Some similar techniques were used in [172], but in the context of only estimating/predicting CCMs from training signals in a single frequency band, not mapping CCMs/CSI from one band to another, the latter being a more difficult task.

In our conference paper [25], we used a CGAN to estimate CCMs; however, there are several limitations to using estimated CCMs. To begin, some time is required to first obtain numerous UL CSI samples, which are then averaged to estimate the UL CCM. This delays the operation of the network somewhat. More significantly, in our CGAN algorithm, although it is trained for all UEs, only one UE CCM is input at a time. This issue could be solved with parallel processing, although with some added complexity and perhaps duplicated functionality. Most notably, the input data can be somewhat large; if there are M elements in the antenna array (itself a relatively large number due to massive MIMO), the CCM for each UE has dimensions $M \times M$.

These problems can be avoided if the CSI can be directly estimated. With the same CGAN structure, the CSI of all UEs associated with the BS can be jointly estimated in one go. The amount of input data for the CGAN is also reduced; if there are K single-antenna UEs, then there are only KM complex values rather than KM^2 when using CCMs². Also, precoding vectors are most typically obtained from the CSI itself. However, there are downsides to direct CSI estimation as well. CSI estimation can be prone to instantaneous channel fluctuations that change the small-scale fading parameters significantly and are difficult to predict. This is not an issue in CCM estimation as changes to the CCM occur on a much longer time scale. Moreover, if UEs become active and inactive, then the input and output dimensions and overall properties of the CSI matrix change, and it can be difficult for machine learning techniques to handle this situation. This problem can be avoided in CCM estimation as well. Therefore, in this article we examine the performance of both CCM and

²However, one could in principle only input about half of this number of values. The CCMs are Hermitian in structure, so the (j, i) th matrix entry is just the complex conjugate of the (i, j) th entry.

CSI estimation since either could be useful depending on the situation and system requirements.

Organization: The chapter is organized as follows. In Section 4.2, we describe the analytical approach to obtain the exact CCMs, and also the method of estimating the CCMs when specific parameters required for calculating exact CCMs are unavailable. Similarly, the system model to obtain simulated UL and DL CSI data is discussed in Section 4.3. The details of our machine learning technique and the CGAN framework are discussed in Section 4.4. A performance evaluation and discussion for exact and imperfect UL CCMs are given in Section 4.5, whereas the performance of CSI estimation is examined in Section 4.6. Finally, we conclude the paper in Section 4.7.

4.2 CCM Modeling

Let us first consider a single-cell massive MIMO system where the BS is equipped with a large antenna array with M antennas and serves K single-antenna UEs. Using ray tracing, the UL CSI (channel vector) $\mathbf{h} \in \mathbb{C}^{M \times 1}$ can be expressed as

$$\mathbf{h} = \int_{\vartheta_{\min}}^{\vartheta_{\max}} \alpha(\vartheta) e^{j\phi(\vartheta)} \mathbf{a}(\vartheta) d\vartheta, \quad (4.1)$$

where $\alpha(\vartheta)$ and $\phi(\vartheta)$ are respectively the amplitude and phase of the complex UL channel gain corresponding to the AoA ϑ , and $\mathbf{a}(\vartheta) \in \mathbb{C}^{M \times 1}$ is the antenna array response in the direction of ϑ . The minimum and maximum AoAs are represented by ϑ_{\min} and ϑ_{\max} . In a sub-6 GHz cellular network with a large number of scatterers, ϑ_{\min} and ϑ_{\max} can be expressed as

$$\vartheta_{\min} = \bar{\vartheta} - \Delta, \quad (4.2)$$

$$\vartheta_{\max} = \bar{\vartheta} + \Delta. \quad (4.3)$$

$\bar{\vartheta}$ is the mean AoA for a given UE, and Δ is the angular spread of AoAs (assumed to be the same for all UEs). For this chapter, we assume a uniform linear array (ULA) at the BS,

although the techniques herein are applicable to an arbitrary type of antenna array. The angle steering (or array response) vector for a ULA is given by

$$\mathbf{a}(\vartheta) = \frac{1}{\sqrt{M}} [1, e^{-j\chi \cos(\vartheta)}, \dots, e^{-j(M-1)\chi \cos(\vartheta)}]^T, \quad (4.4)$$

where $\chi = 2\pi fd/c$; f is the carrier frequency, d is the distance between antenna elements, and c is the speed of light.

In FDD mode, the unknown random quantity is the complex channel gain $g(\vartheta) = \alpha(\vartheta) e^{j\phi(\vartheta)}$. For a Rayleigh fading channel, the phases are uniformly distributed, $\phi(\vartheta) \in [-\pi, \pi]$, and $\mathbf{h} \sim \mathcal{CN}(0, \mathbf{R})$, where $\mathbf{R} \in \mathbb{C}^{M \times M}$ is the CCM. It can also be assumed that paths with different AoAs ϑ_1 and ϑ_2 have uncorrelated channel gains. As a consequence,

$$\mathbb{E} \{g(\vartheta_1)g^*(\vartheta_2)\} = \begin{cases} \mathbb{E} \{|g(\vartheta_1)|^2\} & \text{for } \vartheta_1 = \vartheta_2, \\ \mathbb{E} \{g(\vartheta_1)\} \mathbb{E} \{g^*(\vartheta_2)\} & \text{for } \vartheta_1 \neq \vartheta_2. \end{cases} \quad (4.5)$$

The zero-mean complex Gaussian distribution of \mathbf{h} implies that $\mathbb{E}\{g(\vartheta)\} = 0, \forall \vartheta$. Thus, (4.5) can be further simplified to $\mathbb{E} \{g(\vartheta_1)g^*(\vartheta_2)\} = \mathbb{E} \{\alpha^2(\vartheta_1)\}$ for $\vartheta_1 = \vartheta_2$, and 0 otherwise.

Using (4.1)–(4.5), the UL CCM $\mathbf{R}_{\text{UL}} \in \mathbb{C}^{M \times M}$ can be presented as follows:

$$\mathbf{R}_{\text{UL}} \triangleq \mathbb{E} \{\mathbf{h}_{\text{UL}} \mathbf{h}_{\text{UL}}^H\} = \int_{\bar{\vartheta}-\Delta}^{\bar{\vartheta}+\Delta} \mathbb{E} \{\alpha_{\text{UL}}^2(\vartheta)\} \mathbf{a}_{\text{UL}}(\vartheta) \mathbf{a}_{\text{UL}}^H(\vartheta) d\vartheta. \quad (4.6)$$

The term $\mathbb{E} \{\alpha^2(\vartheta)\}$ is also known as the power angular spectrum (PAS). According to channel model studies, the shape of the PAS is the same as that of the Laplace distribution [173]:

$$\mathbb{E} \{\alpha^2(\vartheta)\} = \frac{C}{\sqrt{2}\Delta} \exp\left(-\frac{\sqrt{2}|\vartheta - \bar{\vartheta}|}{\Delta}\right). \quad (4.7)$$

The value of the constant C is set so as to give the channel the desired large-scale fading properties. Furthermore, based on several measurement tests [174–176], it has been observed that the PAS is similar for the UL and DL and can be related as

$$\mathbb{E} \{\alpha_{\text{DL}}^2(\vartheta)\} \approx \mu(f) \mathbb{E} \{\alpha_{\text{UL}}^2(\vartheta)\}, \quad (4.8)$$

where $\mu(f)$ is a frequency-dependent real-valued quantity. The relation in (4.8) can be physically interpreted as the DL PAS having the same shape as the UL PAS. Using (4.8), the DL CCM $\mathbf{R}_{\text{DL}} \in \mathbb{C}^{M \times M}$ can be expressed as³

$$\mathbf{R}_{\text{DL}} \approx \int_{\bar{\vartheta}-\Delta}^{\bar{\vartheta}+\Delta} \mu(f) \mathbb{E} \{ \alpha_{\text{UL}}^2(\vartheta) \} \mathbf{a}_{\text{DL}}(\vartheta) \mathbf{a}_{\text{DL}}^H(\vartheta) d\vartheta. \quad (4.9)$$

It should be noted that the array response vectors for the UL and DL are different because the carrier frequencies are different for the UL and DL in FDD mode. In our performance evaluation simulations, we use three different types of CCM for performance evaluation: a) exact CCMs, b) CCMs estimated from perfect CSI, and c) CCMs estimated from imperfect CSI.

a) Exact CCMs: By specifying values for the UL and DL carrier frequencies and antenna spacing in (4.4), using random values generated by the distributions over the UEs of $\bar{\vartheta}$ and $\mathbb{E} \{ \alpha^2(\vartheta) \}$, using a constant value of Δ , and then calculating the mathematical expressions in (4.6) and (4.9), exact CCM data are generated.

b) CCMs estimated from perfect CSI: Here we assume the system has no direct knowledge of the CCMs, but can measure the CSI perfectly. A number N of CSI samples are generated for each UE using the exact \mathbf{R} matrices (known by the channel simulator, but not the system) by

$$\mathbf{h} = \mathbf{R}^{1/2} \mathbf{v}, \quad (4.10)$$

where⁴ $\mathbf{v} \sim \mathcal{CN}(0, \mathbf{I}_M)$ models uncorrelated Rayleigh fading. Multiplying by $\mathbf{R}^{1/2}$ creates the desired spatial correlation. Then, the estimated CCM $\hat{\mathbf{R}}$ used by the system is determined by

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{i=1}^N \mathbf{h}_i \mathbf{h}_i^H. \quad (4.11)$$

³As the CGAN learns to estimate \mathbf{R}_{DL} , it inherently subsumes the value of the scalar multiplier $\mu(f)$ within \mathbf{R}_{DL} .

⁴The values of \mathbf{v} and ν are different for each sample.

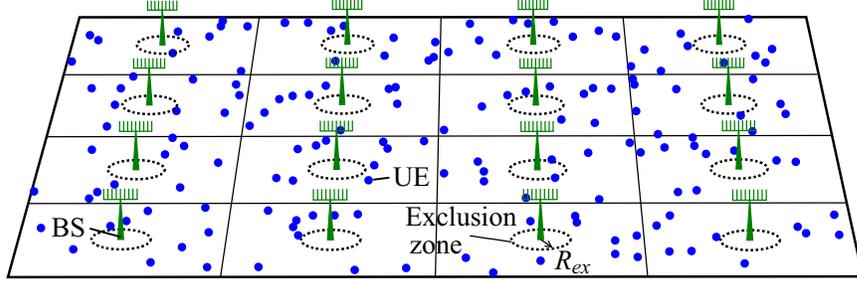


Figure 4.1: Network layout considered for UL and DL CSI dataset generation.

c) **CCMs estimated from imperfect CSI:** Unreliable channel estimation results in imperfect CSI. We start with a simplified error model here; one based on channel estimation from pilot sequences is used in the next section. To generate imperfect CSI samples, we use [132]

$$\tilde{\mathbf{h}} = \zeta \mathbf{h} + \sqrt{1 - \zeta^2} \boldsymbol{\nu}, \quad (4.12)$$

where $\zeta \in [0, 1]$ is the channel estimation reliability, and $\boldsymbol{\nu} \sim \mathcal{CN}(0, \mathbf{I}_M)$. We first generate perfect \mathbf{h} samples from (4.10) for the channel simulator, then use (4.12) to generate imperfect $\tilde{\mathbf{h}}$ samples for system knowledge. Then, N samples of $\tilde{\mathbf{h}}$ are used to estimate the CCM using (4.11).

To train our CGAN, we need DL CCM data as well, to compare the DL CCMs estimated by the CGAN with their “true” values. To obtain exact DL CCMs, we use the UL PAS with the same $\bar{\vartheta}$ and Δ values, scaled by $\mu(f)$. For imperfect DL CCMs, we first generate the exact \mathbf{R}_{DL} and thereafter follow the same steps used for generating imperfect UL CCMs.

4.3 System Model for Simulated CSI Dataset Generation

For direct estimation of DL CSI from UL CSI, we consider a more sophisticated system model, largely based off of one in [3, Ch. 4]. We assume a geographical area with L square-shaped cells. Each cell consists of one BS with M antennas at the cell center and K single-antenna UEs distributed uniformly over the area of the cell outside of an exclusion radius R_{ex} from the BS. Fig. 4.1 illustrates the network layout of the system. We assume

that DL transmissions are enabled using a total resource consumption of τ_c symbols, sent simultaneously but split between the UL and DL at their respective frequencies. Of these, τ_p^{UL} pilot symbols are sent on the UL for channel estimation. The remaining τ_d symbols out of τ_c are data symbols sent on the DL.

As before, we assume the UL channel between UE k and the BS in cell ℓ is modeled as correlated Rayleigh fading, i.e., $\mathbf{h}_{\ell k}^{\text{UL}} \sim \mathcal{CN}(0, \mathbf{R}_{\ell k}^{\text{UL}})$, where $\mathbf{R}_{\ell k}^{\text{UL}}$ is the CCM of this UL channel. The CCM incorporates the large-scale fading coefficient $\beta_{\ell k}$ according to⁵ $\beta_{\ell k} = \text{tr}(\mathbf{R}_{\ell k}^{\text{UL}})$. $\beta_{\ell k}$ includes the effects of path loss and shadow fading and is given (in dB) by

$$\beta_{\ell k} [\text{dB}] = \Upsilon(f) - 10\gamma \log_{10} \left(\frac{d_{\ell k}}{1 \text{ km}} \right) + \Omega_{\ell k}. \quad (4.13)$$

$\Upsilon(f)$ is the frequency-dependent path loss at the reference distance of 1 km, γ is the path loss exponent, $d_{\ell k}$ is the distance between the BS in cell ℓ and UE k , and $\Omega_{\ell k}$ represents log-normal shadow fading distributed $\sim \mathcal{N}(0, \sigma_{\text{sf}}^2)$. Similarly, the DL CSI is modeled as $\mathbf{h}_{\ell k}^{\text{DL}} \sim \mathcal{CN}(0, \mathbf{R}_{\ell k}^{\text{DL}})$. $\Upsilon(f)$ may be considered to incorporate the value of $\mu(f)$ — from (4.6)–(4.9), at the angle $\bar{\vartheta}$, $\frac{\text{tr}(\mathbf{R}_{\ell k}^{\text{DL}})}{\text{tr}(\mathbf{R}_{\ell k}^{\text{UL}})} = \mu(f)$, and from (4.13), $\frac{\text{tr}(\mathbf{R}_{\ell k}^{\text{DL}})}{\text{tr}(\mathbf{R}_{\ell k}^{\text{UL}})} = \Upsilon(f_{\text{DL}}) - \Upsilon(f_{\text{UL}})$ dB. Thus, in linear units, $\mu(f)$ can be given by $10^{[\Upsilon(f_{\text{DL}}) - \Upsilon(f_{\text{UL}})]/10}$.

The channel vectors $\mathbf{h}_{\ell k}^{\text{UL}}$ and $\mathbf{h}_{\ell k}^{\text{DL}}$ are generated from (4.10), using $\mathbf{R}_{\ell k}^{\text{UL}}$ and $\mathbf{R}_{\ell k}^{\text{DL}}$ respectively. The CSI used by the system to train the CGAN, perform precoding, etc. is estimated based on pilot sequences. UL CSI is estimated at the BSs from pilots sent by the UEs. During the training of the CGAN, the corresponding DL CSI is also required, which is estimated by the UEs from pilots sent by the BS, then fed back to the BS⁶. The estimated UL CSI for all UEs in cell ℓ can be expressed jointly as $\hat{\mathbf{H}}_{\ell}^{\text{UL}} = \begin{bmatrix} \hat{\mathbf{h}}_{\ell 1}^{\text{UL}} & \hat{\mathbf{h}}_{\ell 2}^{\text{UL}} & \dots & \hat{\mathbf{h}}_{\ell K}^{\text{UL}} \end{bmatrix} \in \mathbb{C}^{M \times K}$; the estimated (from pilots) DL channel matrix $\hat{\mathbf{H}}_{\ell}^{\text{DL}}$ and the “true” channel matrices $\mathbf{H}_{\ell}^{\text{UL}}$ and $\mathbf{H}_{\ell}^{\text{DL}}$ are defined similarly. The purpose of the CGAN is to estimate $\check{\mathbf{H}}_{\ell}^{\text{DL}}$ ($\check{[\cdot]}$ denoting estimates from the CGAN) from $\hat{\mathbf{H}}_{\ell}^{\text{UL}}$ ($\hat{[\cdot]}$ denoting estimates from pilots).

⁵[3] has $\beta_{\ell k} = \text{tr}(\mathbf{R}_{\ell k}^{\text{UL}})/M$, but also does not have the $1/\sqrt{M}$ normalization of the ULA response vector as we do in (4.4). Since we normalize (4.4), we do not need to normalize $\text{tr}(\mathbf{R}_{\ell k}^{\text{UL}})$ to express $\beta_{\ell k}$.

⁶These DL pilots and DL CSI are only required during CGAN training and not during the normal operation of the system.

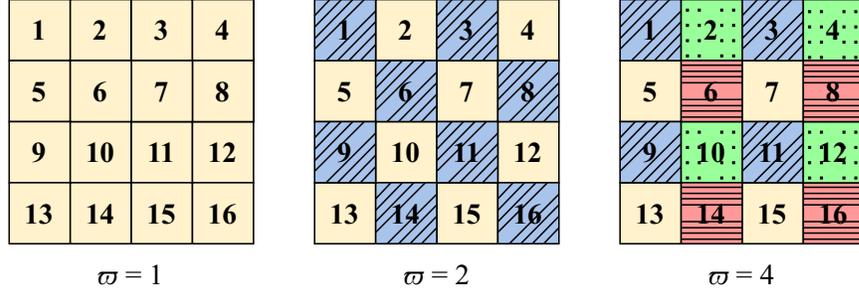


Figure 4.2: Illustration of pilot reuse factor ϖ (cf. [3]).

The pilot sequence for UE k in cell ℓ is given by $\phi_{\ell k} \in \mathbb{C}^{\tau_p^{\text{UL}} \times 1}$ with $\|\phi_{\ell k}\|^2 = \tau_p^{\text{UL}}$, and is transmitted with power $\eta_{\ell k}^{\text{UL}}$. The available pilot sequences are orthogonal to each other, but there may be an insufficient number of these sequences for every UE in the system to be assigned a unique sequence. Consequently, pilot sequences may be reused between cells; let $\mathcal{P}_{\ell k}$ denote the set of all BS-UE pairs (ℓ', k') that use the same pilot sequence $\phi_{\ell k}$. Additionally, let $\mathbf{Y}_\ell^p \in \mathbb{C}^{M \times \tau_p^{\text{UL}}}$ be the received pilot signal at the BS in cell ℓ from all UEs in all cells. The pilot signal $\mathbf{y}_{\ell k}^p \in \mathbb{C}^{M \times 1}$ for UE k in cell ℓ can be extracted by $\mathbf{y}_{\ell k}^p = \mathbf{Y}_\ell^p \phi_{\ell k}^*$. An MMSE estimate⁷ for the UL CSI can then be obtained by [3]

$$\hat{\mathbf{h}}_{\ell k}^{\text{UL}} = \sqrt{\eta_{\ell k}^{\text{UL}}} \mathbf{R}_{\ell k}^{\text{UL}} \Psi_{\ell k}^{-1} \mathbf{y}_{\ell k}^p, \quad (4.14)$$

where

$$\Psi_{\ell k} = \sum_{\forall (\ell', k') \in \mathcal{P}_{\ell k}} \eta_{\ell' k'}^{\text{UL}} \tau_p^{\text{UL}} \mathbf{R}_{\ell' k'}^{\text{UL}} + \sigma_n^2 \mathbf{I}_M. \quad (4.15)$$

Here, $\mathbf{R}_{\ell k'}^{\text{UL}}$ is the CCM of the channel between user k' located in cell ℓ' and the BS in cell ℓ (not ℓ'), and σ_n^2 is the noise power, which we assume is the same for all UEs and BSs on both the UL and DL. $\Psi_{\ell k}$ is the covariance matrix of $\mathbf{y}_{\ell k}^p$, which BS ℓ can obtain from the overall pilot signal without needing to know the individual CCMs $\mathbf{R}_{\ell k'}^{\text{UL}}$ of UEs in other cells.

For the network layout shown in Fig. 4.1, we consider specifically $L = 16$ cells. The distribution of pilot sequences among the cells is determined by the pilot reuse factor ϖ , meaning pilot sequences are reused every ϖ cells. The physical interpretation of the pilot

⁷In practice, the BS can obtain knowledge of $\mathbf{R}_{\ell k}^{\text{UL}}$ and $\Psi_{\ell k}$ in (4.14) and (4.15) from averaging multiple samples of $\hat{\mathbf{h}}_{\ell k}^{\text{UL}}$ and $\mathbf{y}_{\ell k}^p$, respectively, similar to what is shown in (4.11).

reuse factor is illustrated in Fig. 4.2 for $\varpi = 1, 2$, and 4. A larger value of ϖ means more orthogonal pilot sequences are required with larger τ_p^{UL} , and thus the number of symbols used for DL data transmission decreases ($\tau_d = \tau_c - \tau_p^{\text{UL}}$, where we assume τ_c stays constant) as the trade-off of obtaining more accurate UL CSI. The length of the pilot sequences as a function of the pilot reuse factor is given by $\tau_p^{\text{UL}} = \varpi K$. Only during CGAN training (so τ_c is unaffected), each BS also sends pilot sequences from each of its M antennas with power η_ℓ^{DL} and length $\tau_p^{\text{DL}} = M\varpi$, which also are reused every ϖ cells. For training, the UEs estimate and feedback the CSI from these pilots to their BS, similar to (4.14)–(4.15). However, the sum in (4.15) now becomes over the antennas m' in the cells ℓ' that share a sequence.

After the DL CSI is estimated by the CGAN, we perform DL precoding using the resulting channel vectors. The precoding vector $\mathbf{w}_{\ell k} \in \mathbb{C}^{M \times 1}$ depends on the precoding scheme used. We consider regularized zero-forcing (RZF), zero-forcing (ZF), and maximum ratio (MR) precoding, where the (arbitrarily-scaled) precoding vectors $\overline{\mathbf{W}}_\ell = [\overline{\mathbf{w}}_{\ell 1} \ \overline{\mathbf{w}}_{\ell 2} \ \dots \ \overline{\mathbf{w}}_{\ell K}]$ for cell ℓ are given by [3, Eq. (4.9)–(4.11)]:

$$\overline{\mathbf{W}}_\ell^{\text{RZF}} = \check{\mathbf{H}}_\ell^{\text{DL}} \left((\check{\mathbf{H}}_\ell^{\text{DL}})^H \check{\mathbf{H}}_\ell^{\text{DL}} + \sigma_n^2 \mathbf{P}_\ell^{-1} \right)^{-1}, \quad (4.16a)$$

$$\overline{\mathbf{W}}_\ell^{\text{ZF}} = \check{\mathbf{H}}_\ell^{\text{DL}} \left((\check{\mathbf{H}}_\ell^{\text{DL}})^H \check{\mathbf{H}}_\ell^{\text{DL}} \right)^{-1}, \quad (4.16b)$$

$$\overline{\mathbf{W}}_\ell^{\text{MR}} = \check{\mathbf{H}}_\ell^{\text{DL}}. \quad (4.16c)$$

In (4.16a), $\mathbf{P}_\ell = \text{diag}(\rho_{\ell 1}, \rho_{\ell 2}, \dots, \rho_{\ell K})$, where $\rho_{\ell k}$ is the transmit power assigned in cell ℓ to the signal to UE k . The precoding vectors are normalized by $\mathbf{w}_{\ell k} = \overline{\mathbf{w}}_{\ell k} / \sqrt{\mathbb{E}\{\|\overline{\mathbf{w}}_{\ell k}\|^2\}}$.

The received DL data signal $y_{\ell k}$ at UE k in cell ℓ is

$$y_{\ell k} = \sum_{j=1}^L \sum_{i=1}^K \sqrt{\rho_{ji}} (\mathbf{h}_{jk}^{\text{DL}})^H \mathbf{w}_{ji} \varsigma_{ji} + n_{\ell k}, \quad (4.17)$$

where ς_{ji} is the data symbol from the BS in cell j for its UE i , with $\mathbb{E}\{|\varsigma_{ji}|^2\} = 1, \forall j, i$, and $n_{\ell k} \sim \mathcal{CN}(0, \sigma_n^2)$ is noise. The useful part of the double sum in (4.17) for UE k in cell ℓ

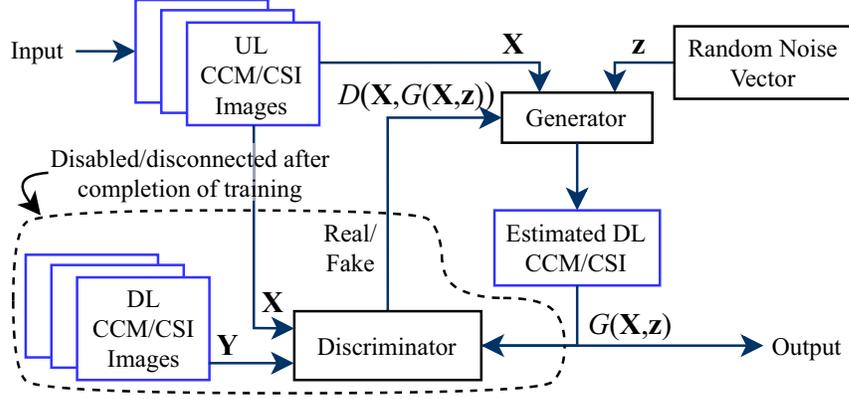


Figure 4.3: Block diagram of our CGAN algorithm.

is $\sqrt{\rho_{\ell k}}(\mathbf{h}_{\ell k}^{\text{DL}})^H \mathbf{w}_{\ell k \zeta_{\ell k}}$; the remainder is multi-user interference. The resulting DL effective SINR is given by [3]

$$\text{SINR}_{\ell k}^{\text{DL}} = \frac{\rho_{\ell k} |\mathbb{E} \{ \mathbf{w}_{\ell k}^H \mathbf{h}_{\ell k}^{\text{DL}} \}|^2}{\sum_{j=1}^L \sum_{i=1}^K \rho_{ji} \mathbb{E} \{ |\mathbf{w}_{ji}^H \mathbf{h}_{ji}^{\text{DL}}|^2 \} - \rho_{\ell k} |\mathbb{E} \{ \mathbf{w}_{\ell k}^H \mathbf{h}_{\ell k}^{\text{DL}} \}|^2 + \sigma_n^2}. \quad (4.18)$$

An achievable effective SE for DL data transmissions to UE k in cell ℓ , accounting for the resources consumed by pilot signals, may be defined as

$$\text{SE}_{\ell k}^{\text{DL}} = \frac{\tau_d}{\tau_c} \log_2(1 + \text{SINR}_{\ell k}^{\text{DL}}). \quad (4.19)$$

(4.19) includes the factor τ_d/τ_c because to enable the DL transmission of τ_d symbols at the SINR of (4.18), a total of τ_c symbols worth of resources must be consumed on the UL and DL combined. Note also that the above SE will naturally be less than what is maximally achievable since the precoding is performed using estimated CSI $\check{\mathbf{H}}_{\ell}^{\text{DL}}$ rather than the exact CSI $\mathbf{H}_{\ell}^{\text{DL}}$. The precoding is therefore imperfect and results in additional residual interference between UEs compared to if the exact CSI is used, leading to reduced SE.

4.4 CGAN Framework

GANs are a promising machine learning framework that are typically used in image processing applications [67]. A GAN consists of two main components, a generator block and a discriminator block, each with its own neural network. In image processing applications,



Figure 4.4: Example images of exact UL CCMs for $M = 256$.

the generator block generates fake images and the discriminator block, trained with real images, determines if the generated image is fake or real. During training, the generator is penalized if the discriminator identifies the image as fake, and the generator readjusts its neural network weights accordingly. Similarly, when the discriminator fails to identify the image as fake, then the discriminator is penalized and it accordingly updates its neural network weights. However, a traditional GAN generates fake images randomly and does not depend on any specific input (which, if it has one, is typically noise). However, in our case we need to obtain the DL CCMs/CSI for specific UL CCMs/CSI. So, we use a variant of GAN, i.e., a CGAN [69], to solve the UL-to-DL CCM and CSI mapping problems. Unlike a traditional GAN, the output of a CGAN can be controlled by a conditional input. A block diagram of our CGAN algorithm is illustrated in Fig. 4.3. The random noise vectors are used only to initialize the neural network and filter weights.

To employ the CGAN, first we need to generate images from the UL and DL CCMs or CSI. To this end, we separate the real and imaginary data of the matrices and then use those values for red-green-blue (RGB) image channels. Raw RGB images are in the form of an $X \times Y \times 3$ dataset, where X is the width of the image, Y is the height of the image, and three channels are used for red, green, and blue colors. In the case of CCMs, the size of the image is $M \times M \times 3$, whereas for CSI the size is $M \times K \times 3$. Because of potential issues with the operation of the CGAN in downsampling and upsampling rectangular-shaped images, we use zero-padding on the CSI matrices to expand them to the size of $M \times M \times 3$; the operations in question shall be discussed in Section 4.4.2. We use the real values of the CCM or CSI for the red channel or $(M, M, 1)$ matrix, and use the imaginary data for the



Figure 4.5: Example images of UL CSI (rotated by 90°) for $M = 256$, $K = 10$, and several values of ϖ .

green channel or $(M, M, 2)$ matrix. We normalize the real and imaginary data to the range⁸ of $[-127.5, 127.5]$. For the otherwise unused blue channel or $(M, M, 3)$ matrix, we just set all values to 127.5. A few example images of exact UL CCMs, created using the above method, are shown in Fig. 4.4 for $M = 256$. Images corresponding to the DL CCMs are generated in the same fashion and used to train the CGAN. If the CCMs are estimated using imperfect CSI as per (4.12), the images become more noisy. Images corresponding to UL CSI (rotated by 90° to fit better on the page) are illustrated in Fig. 4.5 for several values of ϖ . It can be observed that higher values of ϖ result in finer details in the images (i.e., less blurry/fuzzy), since the UL CSI estimation is better with the resulting longer pilot sequences, being less corrupted by pilot contamination.

4.4.1 Objective of CGAN

To model the objective function of the CGAN, we follow the same approach as [69]. The objective function of our CGAN is a weighted sum of the binary cross-entropy loss function that penalizes the generator if the generated image is identified as fake, plus the mean absolute error that penalizes pixel-wise error. The generator G learns a mapping from input image \mathbf{X} and random noise vector \mathbf{z} to a corresponding “true” output image \mathbf{Y} , and the discriminator D is trained to identify fake images generated by the generator. The objective

⁸These normalized values can be mapped directly to color values between 0 and 255 by simply adding 127.5. For RGB images, the color values are also typically quantized integers, e.g., 8 bits per color, but for our purposes, we use floating point values in the CGAN.

function of the CGAN can be expressed as follows [69]:

$$\begin{aligned} L_{\text{CGAN}}(G, D) = & \mathbb{E}_{\mathbf{X}, \mathbf{z}} \{\log(1 - D(\mathbf{X}, G(\mathbf{X}, \mathbf{z})))\} \\ & + \mathbb{E}_{\mathbf{X}, \mathbf{Y}} \{\log D(\mathbf{X}, \mathbf{Y})\} + \lambda \mathbb{E}_{\mathbf{X}, \mathbf{Y}, \mathbf{z}} \{\|\mathbf{Y} - G(\mathbf{X}, \mathbf{z})\|_1\}. \end{aligned} \quad (4.20)$$

$G(\mathbf{X}, \mathbf{z})$ is the output of the generator (i.e., the DL CCM/CSI image) given inputs \mathbf{X} and \mathbf{z} , and $D(\mathbf{X}, \mathbf{Y})$ is the discriminator’s estimate of the probability that the data of \mathbf{Y} are real for input \mathbf{X} . G tries to minimize the objective function in (4.20), while D tries to maximize it. Therefore, the overall objective of the CGAN is a minimax optimization, given by [69]

$$G^* = \arg \min_G \max_D L_{\text{CGAN}}(G, D). \quad (4.21)$$

The last term in (4.20) represents the pixel-wise difference between the generated image and the corresponding “true” image, and penalizes the generator based on the size of the difference; the parameter λ controls the relative size of the penalty. Higher values of λ decrease the presence of artifacts in the generated image [69]. We have considered $\lambda = 100$ for performance evaluation, as used in [69].

4.4.2 Architecture of CGAN

We design the neural network architectures of the generator and discriminator blocks (illustrated in Figs. 4.6 and 4.7, respectively) similar to [68]. Both the generator and discriminator blocks use modules of convolution—batch-normalization—rectified linear units (ReLU) [177].

The generator block of our CGAN is a U-shaped network (U-net) connection [178] between the encoders and decoders. Seven encoder and seven decoder blocks are used with skip-connections. Each skip-connection simply concatenates all channels of the connected blocks, which helps in mapping a high resolution input to a high resolution output. 2D convolutional filters are used in the encoder blocks; 2D convolutional transpose filters [179] of the same size are used in the decoder blocks, but in the reverse order. For the discriminator block, a convolutional neural network with five layers of 2D convolutional filters are

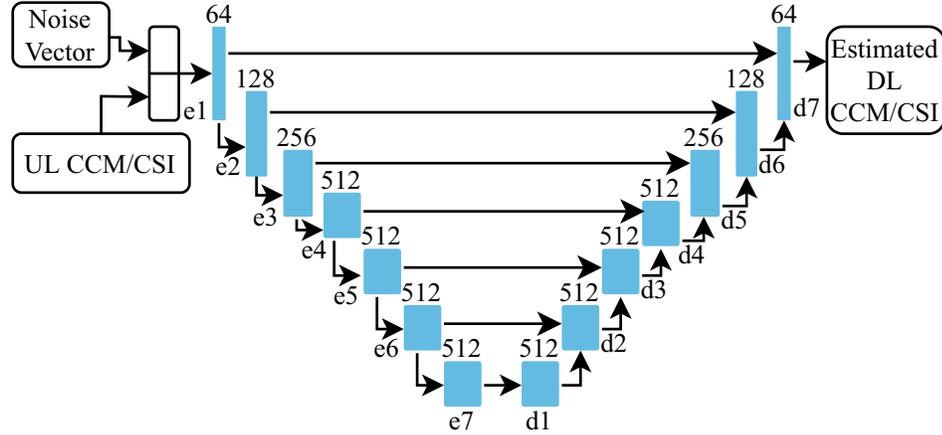


Figure 4.6: Neural network architecture of generator block.

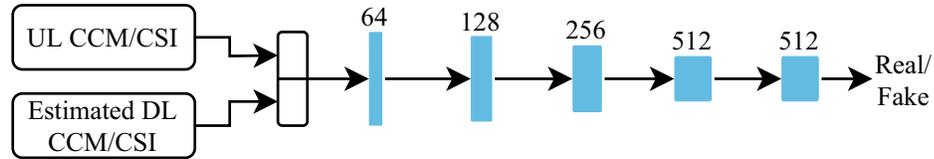


Figure 4.7: Neural network architecture of discriminator block.

used; these are the same as in the first five layers of the generator block. The number of filters in each encoder/decoder layer is shown in Figs. 4.6 and 4.7. In both the generator and discriminator filters, a kernel size of 4×4 and stride size of 2×2 are used⁹. In decoders d1–d4, to reduce the likelihood of generator overfitting¹⁰, we have added a dropout regularization method with retention probability $p = 0.8$ as in [181]; this feature was not present in our conference paper [25]. The method temporarily disconnects neurons at random during training, so each neural network update will be to a slightly different configuration.

In convolutional neural networks, filters extract then map small patches of image data to learn different sections of an input image. The operation of a 2D convolutional filter is shown in Fig. 4.8. For convolution, on each RGB channel, a dot product is taken between a kernel-sized patch of the input and the filter. In Fig. 4.8, the 4×4 square patch that is

⁹These values for the stride size and kernel size might not be optimal, and can be the subject of further experiments to assess their impact on the performance.

¹⁰Generator overfitting takes place when the generator effectively “memorizes” the mapping of a UL CCM image to its corresponding “true” DL CCM image to such an extent that it cannot generalize to new input data or to variations of existing data (e.g., a noisy image) [180]. Overfitting tends to occur in relation to a subset of the training images and can potentially happen for the entire training set if its cardinality is too small.

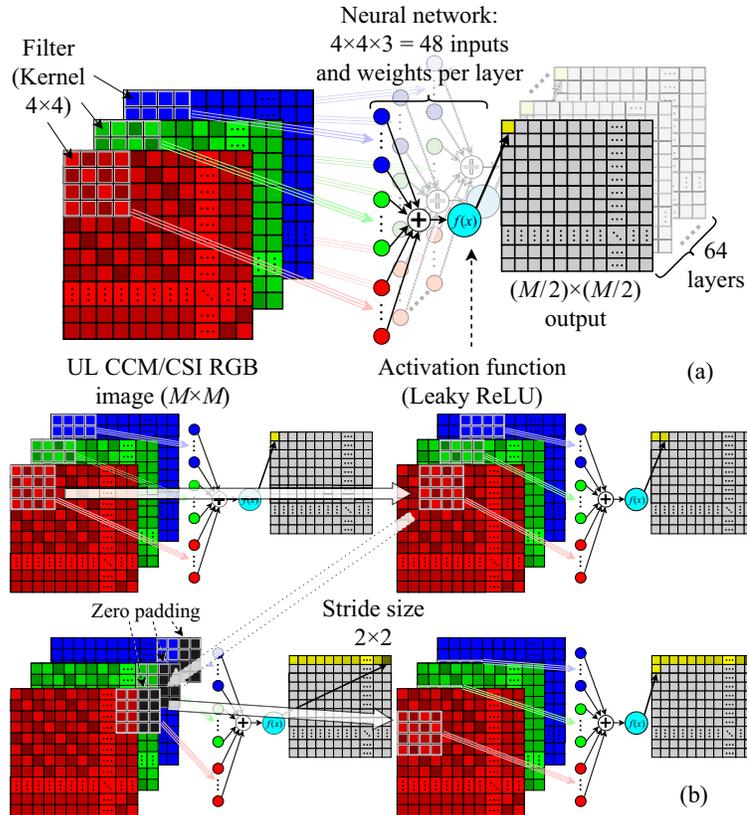


Figure 4.8: Operation of 2D convolutional filter, illustrating the first stage of our CGAN generator for $M \times M$ input image. (a) Structure of filters, neural networks, and outputs on each layer. (b) Movement of filter according to stride size.

convolved in the first step is shown by the grey grid at the top left of the RGB image. The convolved values are used as inputs to the neural network to ultimately obtain a single pixel as the output. The values of the filters and the neural network weights are both initialized randomly using the noise vector, whose entries are Gaussian-distributed with zero mean and a standard deviation of 0.02. Once the convolution of a patch is complete, the filter slides across the image according to the stride size. In our case, the filter moves two pixels for consecutive operations, initially horizontally, as seen at the top right of Fig. 4.8(b). When the filter's right side moves past the end of the row, to obtain the value for the right-most output column, the right-most filter values are padded with zeros to account for their lack of input data (see lower left of Fig. 4.8(b)). Once all the pixels in a row have been convolved, the filter returns to the start of the row, slides two pixels vertically, and starts on the next

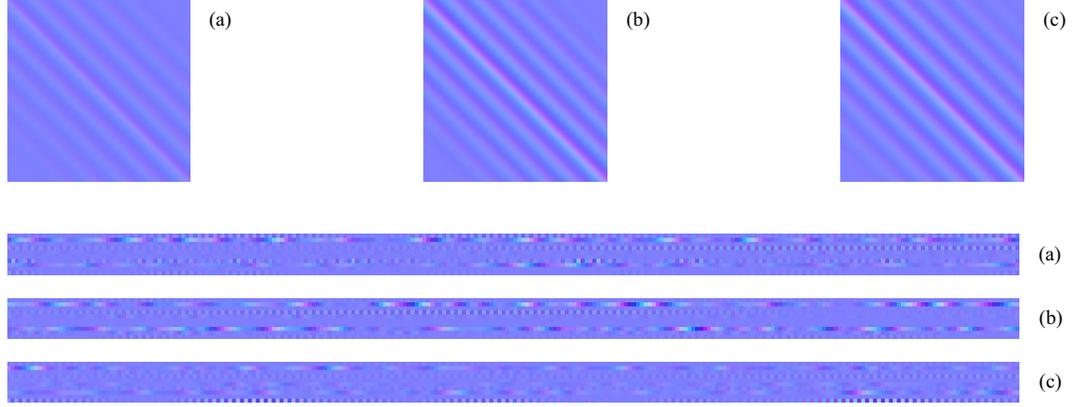


Figure 4.9: Example CCM and CSI images for $M = 256$, $K = 10$, and $\varpi = 4$. (a) Exact UL CCM and MMSE-estimated UL CSI. (b) The corresponding exact CCM and MMSE-estimated CSI for the DL. (c) DL CCM and CSI generated by CGAN.

row (Fig. 4.8(b), lower right). In this manner, the filter slides across the whole image and extracts different features of the image. In each encoder layer, multiple filters are used to map the features of the image more precisely. The outputs of all the filters are stacked together to obtain the final output of each layer. Therefore, in the case of $M = 256$, or an input image of size 256×256 , the output dimensions¹¹ of the first layer with 64 filters are $128 \times 128 \times 64$.

The values obtained from the convolution operation are used as the inputs to the neural network, shown by the red, green, and blue input circles in Fig. 4.8. A weighted sum of the input values is fed to the single neuron (shown by the cyan “ $f(x)$ ” circle), which maps the result to a single pixel based on the activation function. In this chapter, we have used normal ReLU as the activation function for the decoders in the generator, and Leaky ReLU [182, 183] as the activation function everywhere else. These are defined as [182]

$$\text{Normal ReLU: } f(x) = \begin{cases} 0 & \text{for } x < 0, \\ x & \text{for } x \geq 0; \end{cases} \quad (4.22)$$

$$\text{Leaky ReLU: } f(x) = \begin{cases} 0.01x & \text{for } x < 0, \\ x & \text{for } x \geq 0. \end{cases} \quad (4.23)$$

¹¹It is the combination of the 4×4 kernel size and 2×2 stride size that causes the reduction by a factor of two in both dimensions, e.g., 256×256 to 128×128 . As a consequence, if the image was rectangular, one dimension could become incompatibly small before it is processed through all the encoder layers. It is for this reason that we use zero-padding on the CSI matrices to enlarge them to have square dimensions.

The advantage of Leaky ReLU over normal ReLU is the range of the function. In normal ReLU, negative input values are mapped to an output of zero. Hence, the range of normal ReLU is $[0, \infty)$, whereas the range of Leaky ReLU is $(-\infty, \infty)$. Therefore, using Leaky ReLU, negative pixel values can also be processed.

The overall process in encoder blocks is known as downsampling. In decoder blocks, the process is reversed, i.e., the downsampled data are upsampled to generate the estimated DL CCM/CSI image. Each decoder block uses multiple 2D convolutional transpose filters, whose operation is the inverse of convolutional filters: the value of a single pixel is multiplied with each element in the filter, then the values are placed in a new empty matrix. This occurs each time the filter slides according to its stride size. In the final decoder layer, the three final output matrices correspond to the three RGB colors for the estimated DL CCM or DL CSI. A detailed visual representation of a convolutional transpose filter is given in [179]. Example images of CCMs and CSI matrices generated using the CGAN are illustrated in Fig. 4.9.

After initialization, the values in the filters remain constant. In contrast, all the weights in the neural network (for example, $\{w_1, \dots, w_{48}\}$ in the first encoder layer) are updated after each training step as the generator tries to minimize the loss. As one may expect, updating the weights in one layer impacts the other layers in the neural network. Coordination of updating weights across multiple layers is done using the batch-normalization technique [177], which has been designed to optimize the performance of a convolutional neural network [180]. For additional details and information on the operation of the CGAN, we refer the reader to [68, 69, 177–183].

Remark: We assume that DL CCMs or CSI are fed back from the UEs during an initial setup/training phase for the CGAN; the techniques described in Section 4.1.1 could reduce the complexity of this overhead. However, this full feedback¹² is only required during training. After training is complete, the discriminator block is disabled and only the generator

¹²Nevertheless, some less complex periodic feedback may be beneficial to account for UEs entering/leaving the system or their motion.

block operates, estimating DL CCMs/CSI for new input UL CCMs/CSI. Hence, the short-term training complexity of the scheme for training is somewhat high. However, over the longer term after training, the complexity would be much smaller than existing schemes such as [161–164], which still require some form of continuous feedback.

4.5 Performance of CCM Estimation

The performance of our CGAN for CCM estimation is studied for several values of antenna array size M , and both perfect and imperfect CCMs and CSI as described at the end of Section 4.2. For imperfect CCMs and CSI, we also examine the effect of the number of samples N on the performance. The performance of our CGAN is compared against several state-of-the-art algorithms for UL-to-DL CCM mapping.

For our simulations, we consider a UL carrier frequency of 2 GHz and DL carrier frequency of 2.18 GHz [133]. The ULA element spacing is half the wavelength at the UL frequency. To generate the CCMs, the mean AoA over all UEs is assumed to be uniformly distributed as $\bar{\vartheta} \sim \mathcal{U}(-\pi, \pi)$ (i.e., UEs are distributed uniformly omnidirectionally around the BS), and a constant value of $\Delta = 5^\circ$ [173] is used for the angular spread. To generate CCM data for 500 UEs, we first generate a value of $\bar{\vartheta}$ for each UE, then use it in (4.6) and (4.9) to generate CCM data for that UE. Since the exact value of $\mu(f)$ in (4.9) is immaterial to the operation and performance of the CGAN (more details on this below), to generate DL CCM data, we just use a constant value of $\mu(f) = 2$. The CCM data for 400 UEs uniformly randomly chosen¹³ from 500 is used to train the CGAN in each test run. The CGAN performance is evaluated using the data from the remaining 100 UEs after 40 training epochs, where each epoch means processing every image in the training set once. Compared to DNN-based algorithms, this is extremely low, as a DNN requires thousands of epochs to obtain a stable output [159]. The accuracy of the estimated CCM ($\check{\mathbf{R}}$, as output by the CGAN) is evaluated using a normalized mean squared error (NMSE) metric. The

¹³This follows the commonly-used methodology for machine learning of using about 70–90% of the data for training and the remaining 10–30% for testing. We use an 80/20 split recommended in [180].

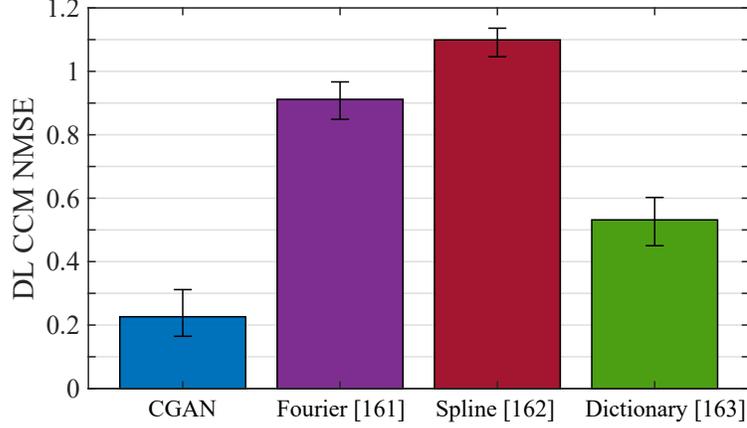


Figure 4.10: NMSE performance comparison of several DL CCM estimation methods, using exact CCMs with $M = 256$. Error brackets represent 95% confidence interval.

NMSE metric¹⁴ can be expressed as

$$\text{NMSE} = \begin{cases} \frac{\|\check{\mathbf{R}} - \mathbf{R}\|_F^2}{\|\mathbf{R}\|_F^2} & \text{for exact CCMs,} \\ \frac{\|\check{\mathbf{R}} - \hat{\mathbf{R}}\|_F^2}{\|\hat{\mathbf{R}}\|_F^2} & \text{for imperfect CCMs.} \end{cases} \quad (4.24)$$

Based on this metric, the performance of the proposed CGAN is studied for three cases: a) exact CCMs, b) CCMs estimated from perfect CSI, and c) CCMs estimated from imperfect CSI. All the performance results are averaged over 100 independent runs.

a) Performance for exact CCMs: The NMSE of the CGAN for exact CCMs with $M = 256$ is illustrated in Fig. 4.10. The CGAN performance is compared against the reference algorithms Fourier-transform-based CCM mapping from [161], cubic-spline-based CCM mapping from [162], and dictionary-learning-based CCM mapping from [163]. The error brackets in the plot represent the 95% confidence interval of the NMSE values, calculated using the bootstrap confidence interval method described in [184]. From the results, we observe that our proposed CGAN outperforms all the reference methods. The average NMSE from the CGAN is about 0.226, whereas the next best performing algorithm (the dictionary-learning-based algorithm from [163]) yields an average NMSE of about 0.532,

¹⁴With the normalization of the NMSE metric, the value of $\mu(f)$ cancels out in the numerator and denominator, which is why its value is irrelevant to the performance.

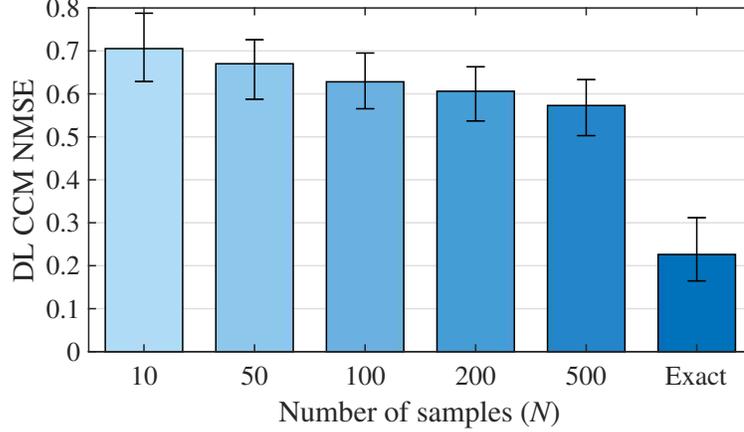


Figure 4.11: NMSE performance comparison when using imperfect CCMs with varying number of samples N ; $M = 256$. Error brackets represent 95% confidence interval.

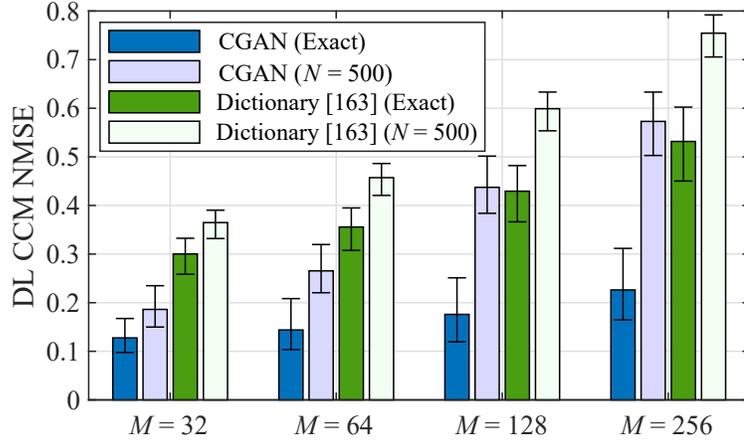


Figure 4.12: NMSE performance comparison between exact and imperfect CCMs for varying antenna array size M . Error brackets represent 95% confidence interval.

which is about 2.35 times higher.

b) Performance for CCMs estimated from perfect CSI: The performance for estimated CCMs is examined for several values of the number of samples N used to estimate the CCM. In Fig. 4.11, we compare the CGAN performance for $N = \{10, 50, 100, 200, 500\}$ with $M = 256$. The results suggest that even with a small number of \mathbf{h} samples, our proposed CGAN provides reasonable performance. This observation is extremely important in terms of practical application; it suggests that our CGAN can be trained using a small number of channel samples and much faster than DNN-based algorithms. With $N = 500$, the NMSE is about 0.573, which is on par with the exact-CCM performance of the dictionary

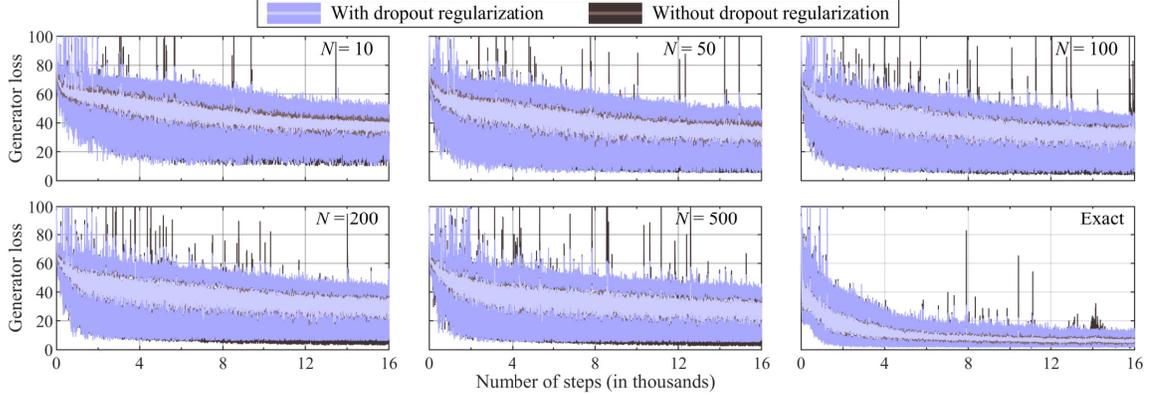


Figure 4.13: Generator loss for varying number of samples N . Light blue line and envelope respectively represent mean value and range of values observed during simulations with dropout regularization; dark grey line and envelope respectively represent mean value and range of values observed during simulations without dropout regularization.

method from [163].

The CGAN dictionary mapping [163] methods’ performance with exact CCMs, compared to using estimated CCMs using $N = 500$ samples, is illustrated in Fig. 4.12 for $M = \{32, 64, 128, 256\}$. We observe that the NMSE for exact CCMs and estimated CCMs is much closer for smaller antenna array sizes. This is expected because estimation for a larger antenna array results in more random values and errors. The CGAN still outperforms the dictionary method when using estimated CCMs, although the relative reduction in NMSE now drops from 2.35 to ranging between 1.32–1.96 (decreasing with increasing M).

The above performance results do not provide insights about the working and learning rate of the CGAN. Therefore, we depict the results for generator loss in Fig. 4.13 with $M = 256$ and several values of N . The metric “generator loss” is calculated from the sum of the first and third terms of (4.20). In these figures, one “step” represents the processing of one CCM image; based on its outcome, the corresponding weights of the neural networks in the generator and discriminator are updated. As stated earlier, there are 400 such evaluations per epoch; hence, 16000 steps equals 40 epochs. From the results in Fig. 4.13, we observe that the generator loss values are much smaller for exact CCMs, whereas when increasing

N the reduction in generator loss is moderate. These results are consistent with the NMSE performance seen in Fig. 4.11. We also observe that the addition of dropout regularization to the generator significantly reduces the frequency of occasional spikes in the generator loss, especially during the later portions of training. This indicates that generator overfitting has been mitigated with dropout regularization. At the same time, while the variance in generator loss values is diminished, there is very little effect on the mean generator loss due to dropout regularization.

c) Performance for CCMs estimated from imperfect CSI: NMSE values when using CCMs estimated from imperfect channel samples are shown in Table 4.1 for several values of the channel estimation reliability factor ζ , with $M = 256$ and $N = 500$ samples. The NMSE values are quite similar to the $N = 500$ case in Fig. 4.11. We observe that the NMSE performance degrades only marginally in the presence of imperfect channel samples. The difference is less than 7% even when the channel estimation reliability is 0.6. This further suggests the robustness of our CGAN-based method, which is able to capture the underlying similarities between UL and DL CCMs even when the CSI obtained from channel estimation is imperfect. We note there are also errors in the estimated DL CCM due to sampling the CSI rather than using exact CCMs, as seen in the difference in the “ $N = 500$ ” and “Exact” bars in Fig. 4.11. However, Table 4.1 shows the incremental loss in performance due to uncertainty in the CSI is small.

4.6 Performance of CSI Estimation

For performance evaluation of CGAN-based DL CSI estimation, we have considered the total area as shown in Fig. 4.1 to be $1 \text{ km} \times 1 \text{ km}$, making each of the borders of the $L = 16$ cells 250 m long. The radius of the exclusion zone is set to $R_{ex} = 35 \text{ m}$. Each cell serves $K = 10$ UEs, and we consider BS antenna array sizes of $M = 32, 64, 128,$ and 256 antennas. Other simulation parameters are given in Table 4.2.

First we compare the NMSE performance for varying antenna array size M . For CSI

Table 4.1: DL CCM NMSE for $N = 500$ Imperfect \mathbf{h} Samples, Antenna Array Size $M = 256$, and Varying Channel Estimation Reliability ζ

Channel estimation reliability (ζ)	DL CCM NMSE
0.6	0.6093
0.7	0.6022
0.8	0.5916
0.9	0.5805
0.99	0.5744
1 (Perfect CSI for \mathbf{h})	0.5729

estimation, the NMSE metric per cell can be expressed as

$$\text{NMSE} = \frac{\|\check{\mathbf{H}}_{\ell}^{\text{DL}} - \hat{\mathbf{H}}_{\ell}^{\text{DL}}\|_F^2}{\|\hat{\mathbf{H}}_{\ell}^{\text{DL}}\|_F^2}. \quad (4.25)$$

CSI data are generated for 10 different instances, where UEs are placed at different locations in each instance. Each time, the CGAN is trained using the 16 UL and DL CSI matrices for 40 training epochs. Thereafter, a new set of 16 UL and DL CSI matrices (with the same UE positions) is generated; the UL CSI is provided to the trained generator of the CGAN, and the estimated DL CSI is compared against the known DL CSI to calculate the NMSE for each cell. The estimated DL CSI is also used for precoding and to calculate the SE per cell. Finally, the NMSE and SE performances are evaluated by averaging the results over all cells per instance, all 10 instances per simulation run, and 100 independent runs.

The NMSE performance when varying the antenna array size M is shown in Fig. 4.14. We compare our CGAN performance against that of a DNN adapted from [169]. Each BS has its own DNN¹⁵ with 4 hidden layers having 256, 256, 128, and 64 neurons, respectively. Each DNN takes the individual UL CSI vectors for that cell’s UEs as inputs (rather than a matrix or RGB image), and uses the mean absolute error between the “true” and estimated DL CSI vectors as its loss function. We also compare performance with a DNN with path gain information (PG-DNN) from [171]. For this, we set the assumed number of path clusters to 15, although since the number of paths in the system model is infinite (with powers

¹⁵We have also investigated one DNN applied to all 16 cells, and found that it performs worse, so we do not include that case here.

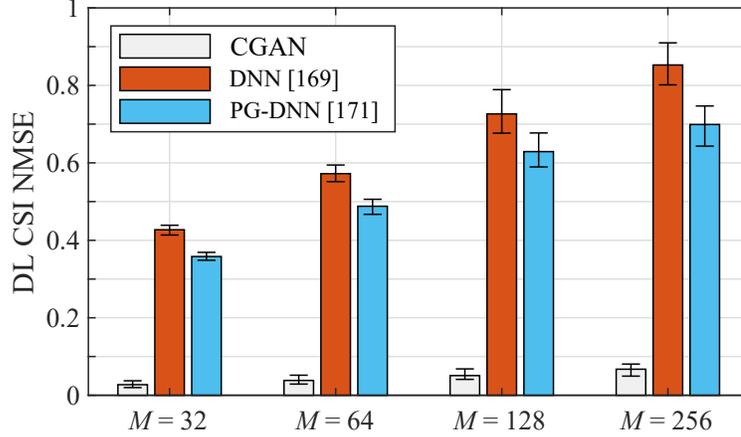


Figure 4.14: NMSE performance comparison of CGAN, DNN adapted from [169], and PG-DNN adapted from [171] for varying antenna array size M with $\varpi = 1$. Error brackets represent 95% confidence interval.

following the shape of the Laplace distribution, as given by (4.7)), any finite number assumed will result in some error. For a fair comparison in terms of training data, the training sets for the CGAN and the DNNs are of the same size, i.e., 10 CSI vectors/ 1 CSI matrix per cell per instance per simulation run, though we allow the DNNs to train for 100 epochs (vs. 40 for the CGAN). From Fig. 4.14, it can be observed that the DNN from [169] performs much worse than our proposed CGAN method, largely due to the very small (for a DNN) training set size. The PG-DNN performs slightly better, but still significantly worse than the CGAN. While the CGAN yields NMSE values ranging from about 0.027–0.067, the DNN and PG-DNN yield about 0.427–0.853 and 0.358–0.699, respectively. To test if either DNN could outperform the CGAN with a larger training set, we increased the number of CSI instances per run ranging from 50–500. The NMSE for the DNNs did decrease initially, but then stabilized around 200 instances, with no significant improvements with larger training sets. The resulting NMSE values were in the range of 0.243–0.574, still significantly higher than those of the CGAN. Much like in Fig. 4.12, it also can be observed that the NMSE again increases when increasing M . However, it is important to note that the CGAN NMSE values for CSI estimation are significantly smaller than those for CCM estimation, which is expected as the effective size of a CSI matrix is much smaller than that of a CCM.

Table 4.2: Simulation Parameters for CSI Dataset Generation

Parameter	Value
UL carrier frequency (f_{UL})	2 GHz
DL carrier frequency (f_{DL})	2.18 GHz
Reference path loss at 1 km (Υ)	-148.1 dB
Path loss exponent (γ)	3.76
Shadow fading standard deviation (σ_{sf})	10
UL pilot power ($\eta_{\ell k}^{\text{UL}}, \forall \ell, k$)	100 mW
DL pilot power ($\eta_{\ell}^{\text{DL}}, \forall \ell$) (only during CGAN training)	$\frac{1000}{M}$ mW
DL transmit power per UE ($\rho_{\ell k}, \forall \ell, k$)	20 dBm
Noise power (σ_n^2)	-94 dBm
$\mu(f)$	2
Total symbols used per DL transmission (τ_c)	200
Pilot reuse factor (ϖ)	1, 2, and 4
Number of UL pilot symbols (τ_p^{UL})	10ϖ
Number of DL pilot symbols (τ_p^{DL}) (only during CGAN training)	$M\varpi$

The NMSE performance for these same values of M and varying pilot reuse factor ϖ is provided in Fig. 4.15. Overall, it can be seen that there is only a small effect on the NMSE with different values of ϖ . The NMSE decreases for greater values of ϖ , which is understandable. Larger values of ϖ correspond to longer training sequences; consequently, the UL CSI and the DL CSI used for training can be estimated better. Correspondingly, the estimates made by the CGAN will also be better when trained with these higher quality data (i.e., less corrupted by pilot contamination), leading to less error in those estimates.

In Fig. 4.16, we examine the average sum SE performance for $M = 32$ and $M = 128$ for RZF, ZF, and MR precoding. We consider three types of DL CSI availability for comparison: 1) genie-aided, in which the system is assumed to know the DL CSI perfectly; 2) feedback-based, where the UEs are assumed to perform MMSE estimation of the DL CSI based on pilot sequences and feed back these channel vectors to their BS; 3) CGAN-based, i.e., the DL CSI as estimated by our CGAN algorithm. (In this last case, the DL CSI used for training is the same type as for the feedback-based case.) The simulation results demonstrate that when using DL CSI estimated by our CGAN algorithm, the average SE achieved is no

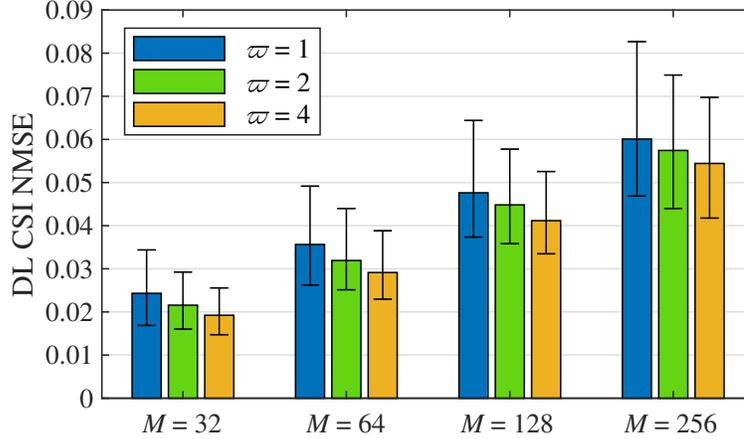


Figure 4.15: CGAN NMSE performance comparison for varying antenna array size with several pilot reuse factors. Error brackets represent 95% confidence interval

worse than about 89–90% of the best possible SE achieved with perfect “genie-aided” channel knowledge. This worst case occurs for all precoding schemes for $M = 32$ and $\varpi = 1$ in Fig. 4.16(a); the mean SE achieved with genie-aided knowledge is about 37.2, 37.0, and 21.7 bits/s/Hz for RZF, ZF, and MR precoding, respectively, whereas the CGAN yields a mean SE of about 33.1, 32.9, and 19.4 bits/s/Hz, respectively. Overall, with MR precoding, the CGAN performance is nearly identical to that of feedback-based CSI availability, which is the best that could be available to a system in practice. With MR precoding, since no inter-UE interference is actively cancelled, the SE is not quite as sensitive to the accuracy of CSI as for RZF or ZF precoding. Better relative performance is seen when increasing M , as the system undergoes more channel hardening [3], so there are smaller fluctuations in the small-scale portion of the CSI around the average. The relative performance also again improves when increasing ϖ , as already seen and explained in regard to Fig. 4.15.

We also investigate the performance of our CGAN-based CSI estimation using measured CSI data. Specifically, we have used the datasets generated as part of [185] and available at [186]. Briefly, the setup for CSI collection in [185] consists of a BS equipped with an 8×8 uniform planar array (UPA) that communicates with single-antenna UEs using orthogonal frequency-division multiplexing (OFDM) transmission in the 2.4 GHz industrial, scientific and medical (ISM) band. CSI is measured at 25 different locations with a mixture of line-

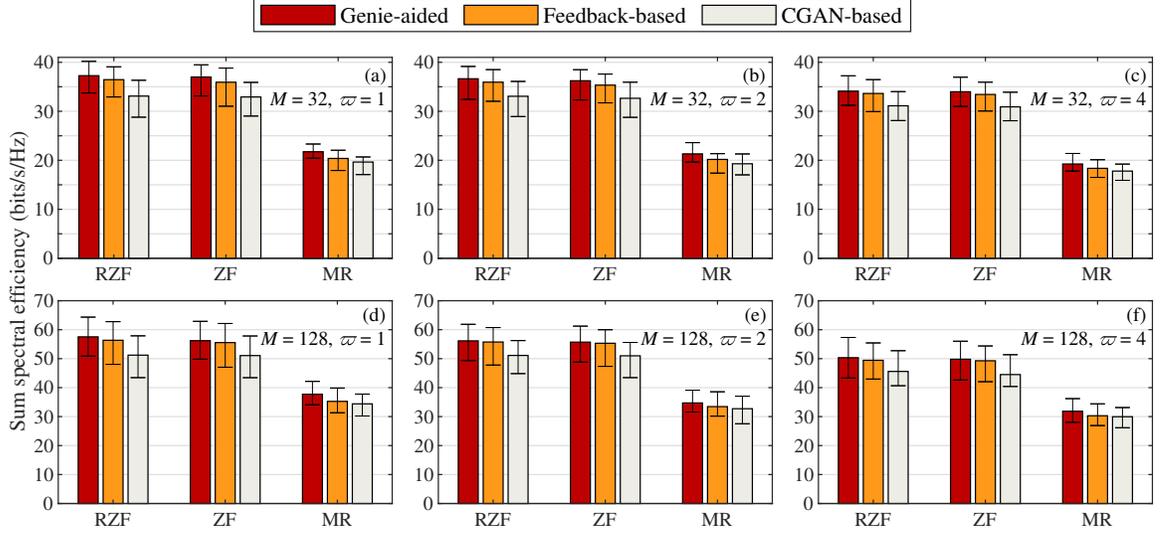


Figure 4.16: Sum spectral efficiency per cell performance results for two different antenna array sizes $M = 32$ and 128 , three pilot reuse factor values $\varpi = 1, 2$, and 4 , and three precoding schemes (RZF, ZF, and MR). Three types of DL CSI availability (genie-aided, feedback-based, and CGAN-based) are compared. Error brackets represent 95% confidence interval.

of-sight and non-line-of-sight conditions; this alternatively can be interpreted that CSI data are available for 25 UEs at those locations. There are 14 subchannels in the 2.4 GHz ISM band, and channel 1 is selected as the UL channel, whereas channel 14 is chosen as the DL channel; the center frequencies of these channels are separated by 72 MHz. Both UL and DL channels have a bandwidth of 20 MHz, and CSI measurements are obtained for 52 OFDM subcarriers. For more details, we refer the reader to [185, 186].

We define UL and DL CSI matrices on a per-subcarrier basis, vectorizing the 8×8 UPA channel gains. With a total of 64 antenna elements, 25 UEs, and 52 subcarriers, this makes the dimensions of $\hat{\mathbf{H}}^{\text{UL}}$ and $\hat{\mathbf{H}}^{\text{DL}}$ 64×25 , and the training dataset contains 52 pairs of UL/DL CSI matrices. From these 52 pairs we randomly select 42 to train the CGAN (following the same methodology mentioned in Footnote 13); the remaining 10 are used to evaluate the NMSE performance of the CGAN. This procedure is repeated 10 times to obtain the average performance. The NMSE results when varying the number of training epochs are illustrated in Fig. 4.17. Initially, the NMSE is comparably high for small epoch values, but it decreases gradually and stabilizes around 85 epochs. The final NMSE is on par with what was seen

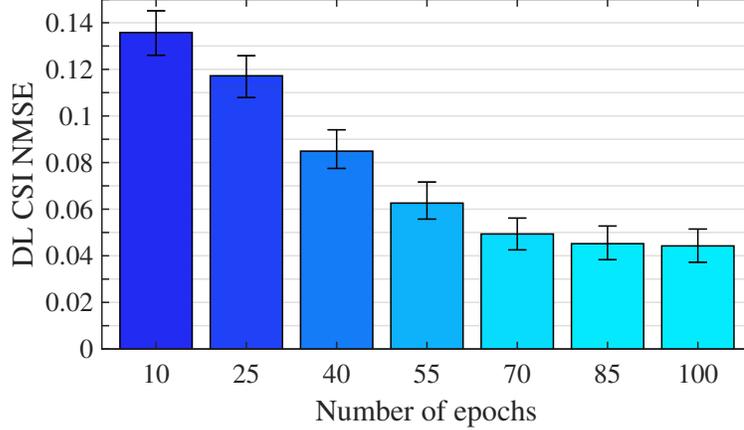


Figure 4.17: NMSE performance for measured CSI data of 8×8 UPA from [185, 186], for varying number of training epochs. Error brackets represent 95% confidence interval.

earlier in Fig. 4.15 for simulated CSI data with 64 antennas (a mean about 0.038 before vs. about 0.045 here, with overlapping confidence intervals). These results demonstrate the robustness of our CGAN approach; it is capable of estimating DL CSI from UL CSI for a variety of operating frequencies (which need not be known to the CGAN), and is not limited to a specific type of antenna array.

4.7 Summary

In this chapter, we have developed CGAN-based UL-to-DL CSI and UL-to-DL CCM mapping methods. We have described analytical frameworks for generating perfect and imperfect CCM data, as well as models to generate CSI data to train the CGAN. The performance of the proposed CGAN has been examined for several antenna array sizes, perfect and imperfect CCMs, and several pilot reuse factors. It has been demonstrated that our proposed CGAN algorithm outperforms several existing algorithms previously described in the literature for CCM estimation. Our CGAN exhibits some sensitivity if the CCMs are estimated from perfect CSI samples instead of having the exact CCMs. However, there is little additional loss observed if the CSI samples themselves are imperfect.

The CGAN also performs well when directly estimating DL CSI, for both simulated and measured CSI data. Simulations have demonstrated that our CGAN approach significantly

outperforms two DNN approaches when all three use the same small-sized training dataset, and still outperforms the DNN approaches when they are allowed to use larger datasets than the CGAN. In terms of SE, when precoding with CSI estimated by the CGAN, the performance loss is at worst about 10–11% compared to if perfect “genie-aided” CSI knowledge is available.

Chapter 5

Access Point Clustering in Cell-Free Massive MIMO Using Conventional and Federated Multi-Agent Reinforcement Learning

5.1 Introduction

Conventionally massive MIMO antenna arrays are assumed to have all their elements co-located at a BS [187–189]. However, in such cases, the massive MIMO cellular network is normally limited by inter-cell interference, resulting in poor cell-edge performance. Therefore, for beyond-5G cellular networks, which put more emphasis on equitable service for all pieces of UE within the coverage area, modifications to the network architecture are necessary. To overcome this shortcoming of conventional massive MIMO, distributed massive MIMO architectures have been studied. In the literature, the core idea of distributed MIMO has been examined under various names, including distributed antenna system (DAS) [190, 191], network MIMO [119, 192, 193], coordinated multipoint (CoMP) transmission [194–198], or C-RAN [199–203].

More recently, distributed architecture has again appeared in the “massive”-sized array regime with the name of cell-free massive MIMO [8]. In cell-free massive MIMO, the APs, each with one or more antennas, are distributed over a geographical area, and multiple APs coordinate to form a virtual massive MIMO array to serve UEs [10]. Cell-free mas-

sive MIMO conceptually removes cell boundaries and therefore suppresses the inter-cell interference problem. This creates more uniformity of service and fairness to UEs over the entire network area [10]. It has been observed in [8] that in a cell-free massive MIMO network, the APs are closer to the UEs than in conventional massive MIMO, yielding higher diversity gain, lower path loss, and better throughput. [8] has also shown that cell-free massive MIMO has significantly better performance than small-cell systems where each UE is served by a single BS.

5.1.1 Prior Research

To the best of our knowledge, distributed architecture specifically in the context of massive MIMO was initially investigated in [204], where a BS selection procedure was developed; the selected BSs coordinated using either maximum ratio combining or MMSE combining to serve the UEs on the UL of hexagonal cells. The authors of [8, 205] first gave the name “cell-free massive MIMO” to the core idea of distributed massive MIMO architecture.

The benefits of cell-free massive MIMO come at the price of increased fronthaul capacity requirements [206]. Existing literature typically assumes infinite-capacity fronthaul links, e.g., [8, 207]. However, prior work in similar contexts has shown that limited fronthaul capacity has a significant performance impact, e.g., for CoMP [208] or C-RAN [209]. The performance of cell-free massive MIMO with capacity-constrained fronthaul links has been studied for some specific scenarios in [206, 210, 211]. Distributed precoding [10] also helps address the issue; we use this approach herein.

Although there are relatively few works on AP clustering or selection, several works on the related problem of antenna selection for massive MIMO are available in the literature, e.g., [212–215]. Antenna selection and AP clustering are fundamentally the same type of problem. However, solutions to the former are most typically centric to the transmit nodes, whereas user-centric solutions are best for the latter. In [213] and [214], the authors have proposed greedy selection algorithms; [213] has maximized the incremental sum rate with

each selected antenna, whereas [214] has used the technique of matching pursuits. The authors of [212] have proposed a branch-and-bound selection algorithm based on the largest minimum singular value of channel submatrices. An ML method for joint antenna selection and user scheduling to maximize the energy efficiency of a single-cell massive MIMO system has been proposed in [215]. The authors of [216] have investigated the related problem of antenna clustering in distributed antenna systems, and [193] has considered cell clustering for network MIMO. In the context of C-RANs, the authors of [199] have considered joint user clustering and sparse beamforming under the constraints of finite-capacity backhaul links, and have obtained a solution by optimizing a weighted MMSE problem. The authors of [203] have framed the user clustering problem as one of a cooperative bargaining game, whose Nash equilibrium has been found in part by a Hungarian method to pair bargaining users.

In the context of cell-free massive MIMO, [211] has proposed two strategies for AP clustering: 1) minimize the number of UE-AP associations subject to the SINR being greater than a threshold, and 2) maximize the minimum SINR subject to a maximum allowable number of APs associated with a UE. More recently, an AP selection method using an ML algorithm based on κ -means clustering has been proposed in [217], a multiple user access scheme using deep RL has been investigated in [218], and a distributed beamforming technique using deep RL has been considered in [219]; [217] has considered DL transmissions, whereas [218] and [219] have considered data transmissions on the UL. Additionally, cell-free massive MIMO DL power control/allocation schemes using deep RL have been developed in [220] and [221].

None of the above antenna selection algorithms considers an environment with UEs in motion. In such a dynamic environment, the association problem needs to be re-solved periodically. Typical DNNs face another challenge in that the input or output state size may vary with the number of nearby and/or active UEs. Therefore, we consider the use of RL to solve the AP clustering problem, as it is suited to handle dynamic environments. Recently,

the authors of [220] have developed RL-based power allocation strategies for a mobile environment. In our MARL algorithm, which we first investigated in [27], we take a more distributed approach with decentralized actors and a centralized critic, inspired by the work in [222]. In this approach, each agent (actor) only has localized environmental information for its AP, whereas the critic has global information. However, a conventional MARL approach such as this has a couple of shortcomings. First, the agents need frequent feedback from the central critic to get their rewards and accordingly update their NN weights. Thus, conventional MARL increases the communication overhead, and to some degree contradicts the basic philosophy of distributed operation in cell-free massive MIMO. Second, the overall policy learned by each agent is strongly contingent on the location of that agent. Therefore, the learned policies are somewhat dependent on the environment, which makes transferring the agents and their policies to a new environment problematic [223, 224].

To overcome these issues, we additionally consider a multi-agent federated reinforcement learning (MAFRL) algorithm. A key consideration in the development of FL was maintaining data privacy between different agents in a system. Agents are only allowed to share learned information (most typically their local NN weights), but not the data with which they train [225]. In the context of communication systems, FL has been used in a variety of scenarios ranging from resource allocation and optimization problems, edge caching and computing, vehicular networks (whether road-based or unmanned aerial vehicles), health care, and the Internet of Things [226–228]. General frameworks for using FL in beyond-5G networks have been proposed in [229, 230], while [231] has surveyed numerous distributed ML techniques for wireless communications, including RL, FL, and other methods that operate in a completely distributed manner with no central coordination. In [232], the authors have considered how best to use the APs of a cell-free massive MIMO system to support and optimize training of an FL framework, where the local NNs being trained are located at the UEs. A related problem has been examined in [233], where massive MIMO and compressive sensing have been used to help reconstruct sparse gradient vectors used for the

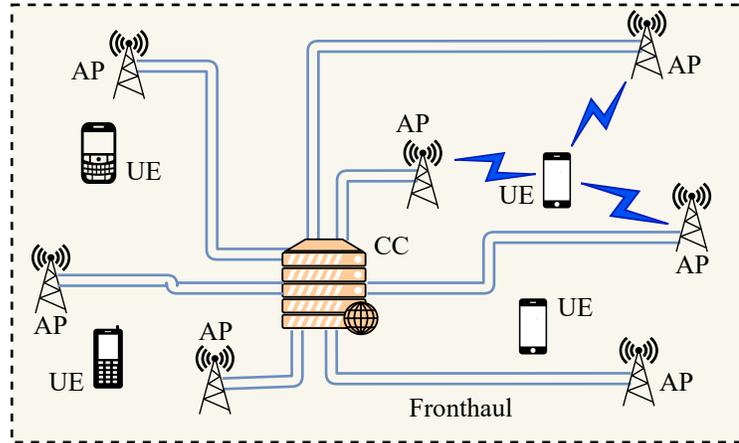


Figure 5.1: Illustration of a cell-free massive MIMO network. (AP: access point, CC: central controller, UE: user equipment.)

FL updates. [234] and [235] have considered FL methods for channel estimation, whereas [236] has used a mixture of deep FL and game theory for dynamic frequency allocation in multicell massive MIMO networks. In [237], the authors have used federated deep RL to tackle the problem of user access control in open radio access networks. However, to the best of our knowledge, our research is the first to combine the advantages of both reinforcement learning and federated learning in the context of optimizing AP clustering in a cell-free massive MIMO network, while also considering the mobility of UEs.

5.1.2 Motivation

We illustrate a typical cell-free massive MIMO network in Fig. 5.1. In canonical cell-free massive MIMO, UEs are served by all the APs [8, 238], which are connected to a CC using fronthaul connections. Therefore, the number of fronthaul connections increases proportionally with the number of APs in the network. Furthermore, as each AP serves all UEs, the overall fronthaul capacity requirement and computing requirements for signal processing also increase. These factors give rise to scalability issues, and thus the canonical form of cell-free massive MIMO is impractical for an arbitrarily large number of APs in the network.

In contrast, in scalable cell-free massive MIMO systems, each UE is served by a subset of

APs [53, 207, 211, 239, 240]. Importantly, the cluster of serving APs should be user-centric and individualized for each UE¹. This challenge was initially addressed in [207], wherein the authors proposed a user-centric AP cluster solution; other works since have also examined the problem (see Section 5.1.1). However, two major issues still remain open: 1) How should the system select which APs to serve a UE in real-time in an environment where the UEs are in motion, where AP selection may need to be updated often? 2) How should the system support the significant fronthaul and computational load in such an environment? We focus mainly on the first question, and address the second through the use of localized precoding.

To tackle these challenges, this chapter focuses on developing machine learning (ML) methods for AP clustering such that each AP can determine the UEs it serves mostly independently of the others. To support the dynamic nature of a mobile environment, reinforcement learning (RL) is a natural choice. In a recent article [220], multi-agent RL (MARL) techniques have been applied to a canonical cell-free massive MIMO network to solve the power allocation problem in a mobile environment, and the performance of the MARL algorithm therein is promising. Those results suggest that MARL algorithms would be suitable for cell-free massive MIMO with mobile UEs, which served as one of our initial motivations for applying MARL to AP clustering. In the conference paper [27] we developed an actor-critic MARL framework that trains the APs to select which UEs to serve; each AP is a distributed agent/actor in the system, and the centralized critic that judges the agents' performance is located at the CC of the network. Because the agents are distributed, with the use of localized precoding, the fronthaul load has been reduced. However, during training, conventional MARL systems require regular information updates regarding rewards from the CC, which can result in significant communication overhead [237]. As an alternative, to train the agents with limited interactions between the CC and agents, federated learning (FL) [225] is a promising technique. Under FL, distributed agents train their neu-

¹This is an important and notable difference from earlier research on distributed architecture, where the clusters of BSs/APs/antennas were typically centric to the serving nodes rather than the UEs.

ral networks (NNs) locally. The CC periodically requests the NN weights from the agents and uses those local weights to compute and distribute a global NN weight update for all agents. FL was initially developed and deployed by Google in their predictive keyboard feature [241]. Later, it has been observed that combining features of FL with those of RL can help reduce the number of interactions between the CC and agents [224, 237].

Organization: The rest of the chapter is organized as follows. In Section 5.2, we describe the model of the cell-free massive MIMO network, the precoding method, and the calculation of SE. The framework and details of our MARL and MAFRL techniques are discussed in Sections 5.3 and 5.4, respectively. We evaluate and discuss the simulated performance of our proposed algorithms in Section 5.5. Finally, we conclude the paper in Section 5.6.

5.2 Cell-Free Massive MIMO System Model

Consider the DL of a cell-free massive MIMO system with L APs that serve a total of K single-antenna UEs, where each AP is equipped with N antennas. We assume that $L \times N \gg K$, which is the typical operating regime for massive MIMO. Each AP can serve any of the UEs, and theoretically can serve any number of them. However, as mentioned earlier, the more UEs served, the more significant the fronthaul load will be. The APs are connected to a CC that forwards UE data symbols to the APs and coordinates the training of ML.

Time-division duplex (TDD) mode is used to alternate between UL and DL transmission. As such, DL channel state information (CSI) may be obtained from the assumption of UL/DL radio channel reciprocity. The UL channel $\mathbf{h}_{k\ell} \in \mathbb{C}^{N \times 1}$ between UE k and AP ℓ is distributed $\sim \mathcal{CN}(0, \mathbf{R}_{k\ell})$, which models correlated Rayleigh fading; $\mathbf{R}_{k\ell} \in \mathbb{C}^{N \times N}$ is the channel covariance matrix. $\beta_{k\ell} = \text{tr}(\mathbf{R}_{k\ell})/N$ is the large-scale fading parameter of the channel, incorporating path loss and shadow fading [10]. The APs make an estimate $\hat{\mathbf{h}}_{k\ell}$ of the UL channels based on pilot sequences sent by the UEs, as follows [10]:

$$\hat{\mathbf{h}}_{k\ell} = \sqrt{\rho_p \tau_p} \mathbf{R}_{k\ell} \mathbf{\Psi}_{k\ell}^{-1} \mathbf{y}_{k\ell}^p, \quad (5.1)$$

where

$$\mathbf{\Psi}_{k\ell} = \mathbb{E}\{\mathbf{y}_{k\ell}^p(\mathbf{y}_{k\ell}^p)^H\} = \rho_p\tau_p\mathbf{R}_{k\ell} + \sigma^2\mathbf{I}_N \quad (5.2)$$

is the $N \times N$ covariance matrix of the received pilot signal $\mathbf{y}_{k\ell}^p \in \mathbb{C}^{N \times 1}$ from UE k at AP ℓ . ρ_p and τ_p are respectively the power and the length of the transmitted pilot sequence, and σ^2 is the variance of the noise (assumed to be distributed $\sim \mathcal{CN}(0, \sigma^2\mathbf{I}_N)$). We also denote the covariance matrix of the error between $\mathbf{h}_{k\ell}$ and $\hat{\mathbf{h}}_{k\ell}$ as $\mathbf{C}_{k\ell} \in \mathbb{C}^{N \times N} = \mathbf{R}_{k\ell} - \rho_p\tau_p\mathbf{R}_{k\ell}\mathbf{\Psi}_{k\ell}^{-1}\mathbf{R}_{k\ell}$ [10]. Here, we assume for simplicity that every UE has its own orthogonal pilot sequence, so interference between pilots does not exist. We also assume the noise variance is the same on the UL and DL and the same² for all UEs and APs.

Let us assume that UE k is served by the APs in set \mathcal{L}_k . We define an $N \times N$ binary diagonal matrix $\mathbf{D}_{k\ell}$ to represent if UE k is associated with AP ℓ :

$$\mathbf{D}_{k\ell} = \begin{cases} \mathbf{I}_N, & \ell \in \mathcal{L}_k; \\ \mathbf{0}_{N \times N}, & \ell \notin \mathcal{L}_k. \end{cases} \quad (5.3)$$

The effective DL channel vector between AP ℓ and UE k can then be considered to be $\mathbf{h}_{k\ell}^H\mathbf{D}_{k\ell}$. We assume that distributed DL precoding is performed, i.e., precoding is done locally at each AP. The data symbol for UE k is given by ζ_k (with $\mathbb{E}\{|\zeta_k|^2\} = 1$), which is sent from the CC to the serving APs over the fronthaul. The received DL signal at UE k is given by [10]

$$y_k = \left(\sum_{\ell=1}^L \mathbf{h}_{k\ell}^H \mathbf{D}_{k\ell} \mathbf{w}_{k\ell} \right) \zeta_k + \sum_{\substack{i=1, \\ i \neq k}}^K \left(\sum_{\ell=1}^L \mathbf{h}_{k\ell}^H \mathbf{D}_{i\ell} \mathbf{w}_{i\ell} \right) \zeta_i + n_k. \quad (5.4)$$

$\mathbf{w}_{k\ell} \in \mathbb{C}^{N \times 1}$ is the precoding vector that AP ℓ uses for UE k , and n_k is the noise. The double summation in the second term of (5.4) represents interference from signals for other UEs, sent from both the serving APs for UE k and the other APs.

To reduce the fronthaul load, we consider localized precoding, where each AP only has the knowledge of its own CSI for the UEs it serves. Thus, no CSI from other APs needs to be exchanged over the fronthaul; only data symbols need to be forwarded. However,

²Even if the noise variances are not the same, due to the network being interference-limited, differences in the variances have a negligible impact on the performance of the system.

with only local CSI knowledge, an AP can't coordinate with any other to serve its UEs. Therefore, it can only create at most N independent spatial streams for its N antennas, meaning it can serve up to N UEs simultaneously³. Specifically, we consider local partial MMSE (LP-MMSE) precoding [10]. Let the set of UEs served by AP ℓ be denoted by \mathcal{D}_ℓ . The (arbitrarily scaled) LP-MMSE precoding vector for UE k at AP ℓ is given by

$$\bar{\mathbf{w}}_{k\ell} = p_{k\ell} \left(\sum_{i \in \mathcal{D}_\ell} p_{i\ell} \left(\hat{\mathbf{h}}_{i\ell} \hat{\mathbf{h}}_{i\ell}^H + \mathbf{C}_{i\ell} \right) + \sigma^2 \mathbf{I}_N \right)^{-1} \mathbf{D}_{k\ell} \hat{\mathbf{h}}_{k\ell}. \quad (5.5)$$

$p_{k\ell}$ ($p_{i\ell}$) is the transmit power assigned by AP ℓ for UE k (i). To normalize the total transmit power, the AP uses the precoding vector $\mathbf{w}_{k\ell} = \bar{\mathbf{w}}_{k\ell} \sqrt{p_{k\ell}} / \sqrt{\mathbb{E}\{\|\bar{\mathbf{w}}_{k\ell}\|^2\}}$. Typically, the transmit power for each UE is determined by a power allocation algorithm, such as in [220] and [221]. However, for simplicity, in this chapter, we have used equal power allocation, i.e., $p_{i\ell} = P_t/|\mathcal{D}_\ell|, \forall i \in \mathcal{D}_\ell$, where P_t is the total transmit power available at the AP (assumed to be the same for all APs). LP-MMSE precoding is scalable to arbitrary network sizes, since the maximum data volume transferred over the fronthaul to AP ℓ is $|\mathcal{D}_\ell| \leq N$ data symbols, which is independent of both K and L .

The effective DL SINR of UE k is given by [10, Eq. (6.22)]

$$\Upsilon_k = \frac{\left| \sum_{\ell=1}^L \mathbb{E}\{\mathbf{h}_{k\ell}^H \mathbf{D}_{k\ell} \mathbf{w}_{k\ell}\} \right|^2}{\sum_{i=1}^K \mathbb{E}\left\{ \left| \sum_{\ell=1}^L \mathbf{h}_{i\ell}^H \mathbf{D}_{k\ell} \mathbf{w}_{i\ell} \right|^2 \right\} - \left| \sum_{\ell=1}^L \mathbb{E}\{\mathbf{h}_{k\ell}^H \mathbf{D}_{k\ell} \mathbf{w}_{k\ell}\} \right|^2 + \sigma^2}. \quad (5.6)$$

An achievable SE for UE k may then be defined as:

$$\eta_k = \log_2(1 + \Upsilon_k). \quad (5.7)$$

This SE of the UEs is used to define the reward functions in our RL algorithms.

5.3 Reinforcement Learning Framework

In this section, we develop the RL framework for solving our AP clustering problem. RL is a very effective ML technique for dynamic environments such as real-time strategic games

³This is in contrast to $C \times N$ UEs that can be jointly served if C APs coordinate with centralized precoding.

and autonomous driving [242–244]. To implement the MARL algorithm, first we define the AP clustering problem as a Markov game [245], represented as a tuple $(\mathcal{L}, \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ [246]. $\mathcal{L} = \{1, 2, \dots, |\mathcal{L}|\}$ is the set of agents, which in our case are the APs. The state of the environment or state space is represented by \mathcal{S} . In our case, the state is based on the received signal strength (RSS) of each UE at the APs. The RSS between AP ℓ and UE k is calculated from the received pilot signal as follows:

$$\text{RSS}_{k\ell} = \|\mathbf{y}_{k\ell}^p\|^2. \quad (5.8)$$

We note that the RSS is directly proportional to $\beta_{k\ell}$, as $\mathbb{E}\{\|\mathbf{y}_{k\ell}^p\|^2\} = \rho_p \tau_p N \beta_{k\ell} + \sigma^2$. Thus, the pilot signals sent by the UEs are used by each AP both to calculate that AP’s RSS values for all UEs, and to estimate the UL CSI for the set of UEs that AP serves. The joint action space \mathcal{A} is the Cartesian product of the action spaces \mathcal{A}_ℓ for all agents. The variable $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ represents the transition probability kernel of moving from one state to another. The reward function is represented by $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, and $\gamma \in [0, 1)$ is called the discount factor. In the MARL algorithm, each agent ℓ has its own parameter vector $\boldsymbol{\theta}_\ell$ (which is basically its NN weights); concatenating $\boldsymbol{\theta}_\ell$ of all agents forms a joint parameter vector $\boldsymbol{\theta}$. In step $t \in \mathbb{N}$, the environment is in state $s_{(t)}$; agent ℓ takes an action $a_{\ell,(t)} \in \mathcal{A}_{\ell,(t)}$ based on the policy $\pi_{\boldsymbol{\theta}_\ell}(a_{\ell,(t)}|s_{(t)})$, where $\mathcal{A}_{\ell,(t)}$ is the action space of agent ℓ at step t . The joint policy of all the agents is

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(a_{1,(t)}, a_{2,(t)}, \dots, a_{L,(t)}|s_{(t)}) = \prod_{\ell=1}^L \pi_{\boldsymbol{\theta}_\ell}(a_{\ell,(t)}|s_{(t)}). \quad (5.9)$$

Training the MARL algorithm consists of groups of steps called “episodes”; the weights of the NNs are updated after each episode following a policy gradient approach [247]. The agents (or actors, in an actor-critic framework) aim to find the optimal policy that, on average, will maximize the cumulative reward in step t , i.e., $R_{(t)} = \sum_{i=0}^{\infty} \gamma^i r_{(t+i)}$. The performance of agent ℓ ’s policy is evaluated using the centralized action-value function $Q_\ell^{\boldsymbol{\pi}^\theta} = (\boldsymbol{x}, a_1, a_2, \dots, a_L)$, where \boldsymbol{x} contains the relevant information about the state of the environment. $Q_\ell^{\boldsymbol{\pi}^\theta}$ defines the algorithm’s critic; essentially, it determines the rewards given

by the critic to agent ℓ depending on the actions of all agents [222]. For a more detailed description of these parameters, we refer the reader to [222, 245–248].

In our considered system, the UEs are mobile, and therefore, the AP clustering should focus on long-term rewards for optimal AP-UE association. The value of the discount factor γ determines over how long of a period an agent’s actions affect its rewards during training [220]. An exponentially-decaying weight γ^i is applied to future rewards; the larger the value of γ , the more emphasis that is placed on long-term rewards. Each AP needs to decide whether it is better to serve a given UE now or wait until later, based on the movement of all UEs. For example, the RSS and SE for a UE and thus the reward for serving that UE will increase over time if said UE is moving towards the AP, and decrease if it is moving away. The emphasis of “waiting until later” on this decision (and how long to wait) depends on the value of γ . Thus, in mobile environments, the discount factor indirectly helps APs learn about possible UE movement and whether serving a specific UE at a given time is good for the cumulative reward.

In this chapter, we have implemented an actor-critic policy gradient MARL-assisted approach, which is efficient in dealing with high-dimensional action spaces [247, 249]. In our case, the size of the action space for each AP is 2^K , as the output state for each UE is either 1 or 0 (i.e., associated with that AP or not). To improve the speed of convergence, we reduce the size of the state space for each agent by first selecting a pool of only Φ UEs with the highest RSS at that AP from the available K . Use of the pool also ensures the algorithm is scalable to arbitrarily large K . This furthermore largely solves the problem of potentially inactive UEs, which would cause the length of the input vector for the NNs to not be constant. We vary the value of Φ to examine its effect on the MARL algorithm’s performance.

However, this approach by itself does not guarantee service to all UEs. For instance, a UE may not be associated with any AP if that UE’s RSS is not within the top Φ RSSs for any AP. To address this issue, at each AP, two additional UEs that are not yet in the AP’s

pool are chosen in a round-robin⁴ fashion and added to the pool; the AP then serves up to N UEs from that enlarged pool. Additionally, we introduce a global penalty to all APs if not all UEs are served. The penalty at time step t is

$$P_{G(t)}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right|\right) = (\varrho + P_{G(t-1)}) \cdot \mathbb{1}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right| < K\right). \quad (5.10)$$

$P_{G(0)}$ is initialized to 0, and a value of ϱ is progressively added to the penalty for each time step that all K UEs are not served. The argument of the indicator function checks if all the UEs in the coverage area have been served. If so, the indicator function resets the penalty to 0. The CC applies the global penalty to the reward of every AP, then forwards the resulting rewards to their corresponding APs over the fronthaul. Enlarging each AP's pool with unserved UEs, combined with the penalty, helps the agents to learn within a few time steps that all the UEs should be served. Overall, in the MARL implementation, the additional overhead is the information shared between the APs and the CC, i.e., the reward for each AP in every time step. The interaction between the cell-free massive MIMO network and the agents' NNs in the MARL algorithm is shown in Fig. 5.2.

The NN for each agent consists of an input layer ($\Phi+2$ nodes containing RSS values of $\Phi+2$ UEs), a hidden layer (20 neurons), and an output layer ($\Phi+2$ neurons that determine the action $a_{\ell,(t)}$ of the agent). The NN weights are initialized randomly with the distribution $\sim \mathcal{N}(0, 0.03^2)$. The activation function of the hidden layer neurons is $\tanh(\cdot)$, whereas for the output layer, it is the $\text{softmax}(\cdot)$ function⁵ [246]. The output of each output node n is the probability χ_n of serving the UE corresponding to input node n . At each time step, using $\{\chi_1, \chi_2, \dots, \chi_{\Phi+2}\}$, the agent calculates the probability of each action from the set of the possible ones. The UE for node n can either be served (with probability χ_n) or not served (with probability $1-\chi_n$), making for $2^{\Phi+2}$ possible actions in total. The agent's action (the

⁴In our conference paper [27], we used uniformly random selection rather than round-robin selection for the two additional UEs, which also worked well. However, random selection does not completely guarantee that all UEs will be considered, although the probability of some UE not being considered eventually is quite low. The choice to consider specifically two additional UEs was made heuristically.

⁵For $\mathbf{z} = [z_1, z_2, \dots, z_N] \in \mathbb{R}^N$, $\text{softmax}(\mathbf{z}) = \frac{[e^{z_1}, e^{z_2}, \dots, e^{z_N}]}{\sum_{i=1}^N e^{z_i}}$.

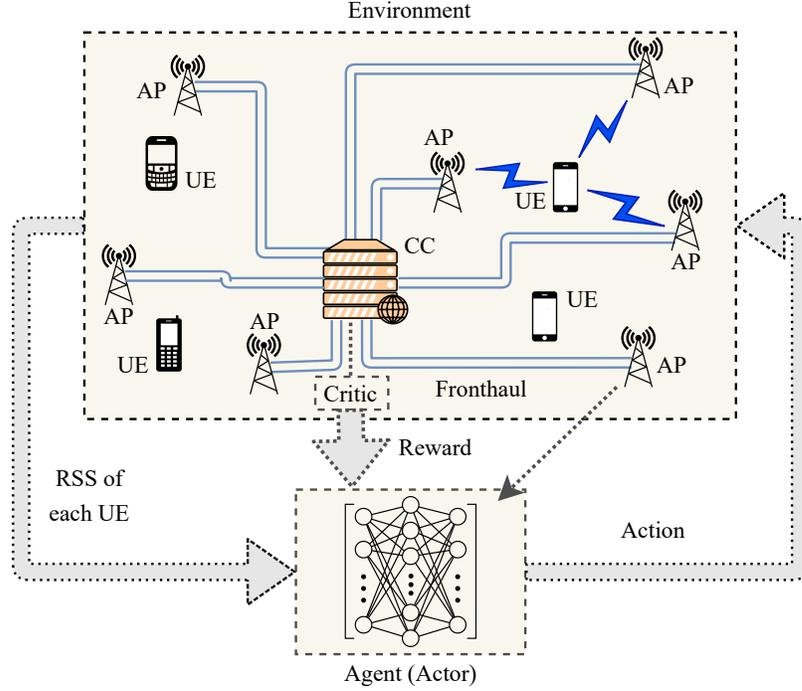


Figure 5.2: Illustration of the decentralized actor, centralized critic MARL algorithm's interactions between the environment and agents.

set of UEs to be served) is then chosen at random as weighted by the action probabilities. This method of choosing an action by weighted random sampling from the set of possible actions is known as stochastic policy gradient-based action selection [222]; it allows for exploration as well as exploitation of acquired knowledge from earlier training.

The (non-convex) optimization problem of determining which APs should serve which UEs in order to maximize the achievable sum SE can be formulated as

$$\max_{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_L} \sum_{k=1}^K \eta_k \quad (5.11a)$$

$$\text{subject to: } \left| \bigcup_{\forall \ell} \mathcal{D}_\ell \right| = K, \quad (5.11b)$$

$$|\mathcal{D}_\ell| \leq N, \forall \ell. \quad (5.11c)$$

Similarly, the optimization problem to maximize the minimum UE SE would replace (5.11a) by

$$\max_{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_L} \min_{k \in \{1, 2, \dots, K\}} \eta_k \quad (5.12)$$

with the same constraints as in (5.11b) and (5.11c). However, finding the globally optimum

solution for either optimization problem would need to be done in a centralized manner. Implementing such a solution in a cell-free scenario would result in higher fronthaul loads because the CC would have to transmit its resulting solution to each AP. Instead, a distributed solution can be found at each AP. Moreover, in a mobile environment, the system should in general optimize the objective function over some interval (such as several sequential time steps t in the set \mathcal{T}) in order to account for the movement of the UEs. Therefore, the optimization problems should be modified. We adjust the max sum SE problem as follows:

$$\max_{\mathcal{D}_{\ell,(t)}, \forall \ell, \forall t \in \mathcal{T}} \sum_{t \in \mathcal{T}} \sum_{\ell=1}^L \sum_{k \in \mathcal{D}_{\ell,(t)}} \eta_{k,(t)} \quad (5.13a)$$

$$\text{subject to: } \left| \bigcup_{\forall \ell} \mathcal{D}_{\ell,(t)} \right| = K, \forall t \in \mathcal{T}, \quad (5.13b)$$

$$\left| \mathcal{D}_{\ell,(t)} \right| \leq N, \forall \ell, \forall t \in \mathcal{T}, \quad (5.13c)$$

whereas for the modified max min SE problem, (5.13a) is replaced by

$$\max_{\mathcal{D}_{\ell,(t)}, \forall \ell, \forall t \in \mathcal{T}} \min_{k \in \mathcal{D}_{\ell,(t)}} \eta_{k,(t)}. \quad (5.14)$$

It is important to note that solutions to these optimization problems do not depend solely on the decisions taken by an individual AP, but rather on the decisions made by the cluster of APs that serve a given UE. Observation of the optimization problem in (5.13) indicates that these decisions depend on the state of the cellular network at the time steps in \mathcal{T} . (If some of these time steps are in the future, the actual state may be replaced by the predicted or expected/average state at that time.) Also, the objective function of the optimization problem exhibits an episodic nature, i.e., the objective function depends on values obtained at multiple time steps. Thus, the overall optimization problem can be extended to a Markov game or Markov decision process (MDP). It has been known for some time that RL can be an efficient methodology to solve MDP problems [250]. This was one of our motivations for using RL in the first place.

We consider four reward policies for our MARL algorithm when evaluating its performance:

Policy 1 — Max sum SE: In this case, the reward function for agent ℓ at time t is defined as

$$r_{\ell,(t)} = \sum_{k \in \mathcal{D}_\ell} \eta_{k,(t)} + P_{\ell,(t)}(|\mathcal{D}_\ell|) + P_{G(t)}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right|\right), \quad (5.15)$$

where η_k is given by (5.7). $P_{\ell,(t)}(|\mathcal{D}_\ell|)$ is a local penalty function that applies if AP ℓ attempts to serve more than N UEs; if so, a penalty of -10 is incurred. $P_{G(t)}(|\bigcup_{\forall \ell} \mathcal{D}_\ell|)$ is the global penalty described in (5.10); we use $\varrho = -20$. The purpose of the penalties is to prevent illegal or undesirable actions by the agents when creating their policies. The penalty values are thus somewhat arbitrary; any large negative value that negates the potential reward of such actions will suffice.

Understandably, since the goal of Policy 1 is to maximize the sum SE of all UEs, the APs will be biased towards associating with the highest RSS (i.e., nearest) UEs. This reduces the system fairness and more distant UEs might not obtain high quality service. Thus, we also consider another reward function that incorporates fairness.

Policy 2 — Max min SE: In this policy, the agents try to maximize the minimum SE of their served UEs, thus providing fairness in the performance. The reward function in this case is expressed as

$$r_{\ell,(t)} = \min_{k \in \mathcal{D}_\ell} \eta_{k,(t)} + P_{\ell,(t)}(|\mathcal{D}_\ell|) + P_{G(t)}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right|\right). \quad (5.16)$$

The penalties in (5.16) are the same as in (5.15). However, if the reward function is expressed as above, without additional constraints such as are typically seen in optimization problems (for example, a constraint that every UE be guaranteed some minimum quality of service), then the agents generally do not learn to each serve multiple UEs. (APs instead prefer serving only one UE each if possible, since that maximizes their minimum, i.e., only, UE SE, although the single UE each AP serves is generally a different one.) To overcome the shortfall of the traditional max min policy, we thirdly use a modified max min SE policy.

Policy 3 — Modified max min SE: The reward function is modified as follows:

$$r_{\ell,(t)} = |\mathcal{D}_\ell| \times \min_{k \in \mathcal{D}_\ell} \eta_{k,(t)} + P_{\ell,(t)}(|\mathcal{D}_\ell|) + P_{G(t)}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right|\right). \quad (5.17)$$

By weighting the minimum UE SE with the number $|\mathcal{D}_\ell|$ of served UEs, the APs learn to serve multiple UEs while still maximizing the minimum SE of the UEs they serve. The penalties in (5.17) are the same as in (5.15).

Policy 4 — Hybrid policy⁶: For the sake of interest, we also examine a policy that is a heuristic hybrid of the max SE and max min SE policies. In this policy, the minimum UE SE is weighted by the sum SE of all the agents’ served UEs. Because of the presence of the sum SE, the agents still learn to serve multiple UEs. The reward function is as follows:

$$r_{\ell,(t)} = \min_{k \in \mathcal{D}_\ell} \eta_{k,(t)} \times \sum_{k \in \mathcal{D}_\ell} \eta_{k,(t)} + P_{\ell,(t)}(|\mathcal{D}_\ell|) + P_{G(t)}\left(\left|\bigcup_{\forall \ell} \mathcal{D}_\ell\right|\right). \quad (5.18)$$

The penalties in (5.18) are the same as in (5.15). It is expected that this reward function should yield a performance somewhere between the performance of the max SE reward and the performance of the max min SE reward by themselves.

At the completion of training, based upon the final probabilities at the NN output nodes, there may remain a very small but non-zero possibility of choosing an action that serves more than N UEs. To ensure that an agent does not take such an action, we force the probability of those actions to be zero.⁷

5.4 Federated Reinforcement Learning Framework

In this section, we develop the MAFRL framework to solve the same AP clustering problem. Unlike in the conventional MARL algorithm, in the MAFRL algorithm the interaction between the CC and the APs is now limited to a periodic exchange of NN weights. Therefore, one can no longer use a centralized critic type of reinforcement learning. Instead, we consider a policy-gradient approach [249] to train the agents. We assume that there is a set \mathcal{L} of agents, with each agent having a local state space and action space and using the same

⁶We called this policy the “max min SE” policy in our conference paper [27]. We have renamed it to be a “hybrid” policy in this chapter, since, as we will show in the simulation results, the reward function of Policy 3 does a much better job of satisfying the max min SE criterion.

⁷We did not encounter any such actions in our simulations, even without forcing the probabilities to be zero. The enforcement therefore mainly guarantees that such actions will not occur over the long-term timescale of the network operation.

reward function. Although the state and action spaces may be different for each agent, the dimensions of the state spaces are the same for every agent, as are the dimensions of the action spaces. The structure of the NN of each agent is the same as in the previous section. Similar to the conventional MARL problem, we formulate the MAFRL problem as a Markov game, represented as a tuple⁸ $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ [251]. The goal of the MAFRL algorithm is to have the $|\mathcal{L}|$ agents jointly learn a policy function π_{θ} that they all use and that performs as close to optimally as possible and uniformly well across the entire environment. This differs from the MARL algorithm, in which each agent has its own (location-dependent) policy. To reduce the communication overhead, agents do not communicate between themselves; instead, they share their parameter vector θ_{ℓ} (i.e., their NN weights) only with the CC.

Similar to the previous section, the agents update the weights of their local NNs after each training episode. Each agent aims to find the optimal policy to maximize its cumulative reward in step t , i.e., $r_{\ell,t} = \sum_{i=0}^{\infty} \gamma^i r_{\ell,t+i}$. The state value function of agent ℓ is defined as $V_{\theta_{\ell}}(s) = \mathbb{E}_{\mathcal{A}_{\ell}, s_{\ell}} \{r_{\ell,0} \mid s_0=s\}$. Mathematically speaking, the goal of each agent is find the policy $\pi_{\theta_{\ell}}^*$ that maximizes the expected state value function:

$$\pi_{\theta_{\ell}}^* = \arg \max_{\pi_{\theta_{\ell}}} \mathbb{E}\{V_{\theta_{\ell}}(s)\}. \quad (5.19)$$

Each AP forms a pool of $\Phi+2$ UEs as the input to its NN in the same fashion as in the MARL algorithm. Furthermore, each agent in the MAFRL algorithm receives information about Υ_k as feedback from the UEs it is serving. Thereafter, it calculates the SE of each UE using (5.7). However, it is not possible for the agents themselves to determine if all UEs have been served or not. In the event one or more UEs have not been served, the CC broadcasts a global penalty to all the APs.

For the MAFRL algorithm, we consider an additional alternative max sum SE reward function, which has the local penalty removed compared to Policy 1. We make the assumption here that if an AP serves more than N UEs, then the resulting inter-user interference

⁸ \mathcal{L} no longer appears in the tuple since every agent is playing a copy of the same game.

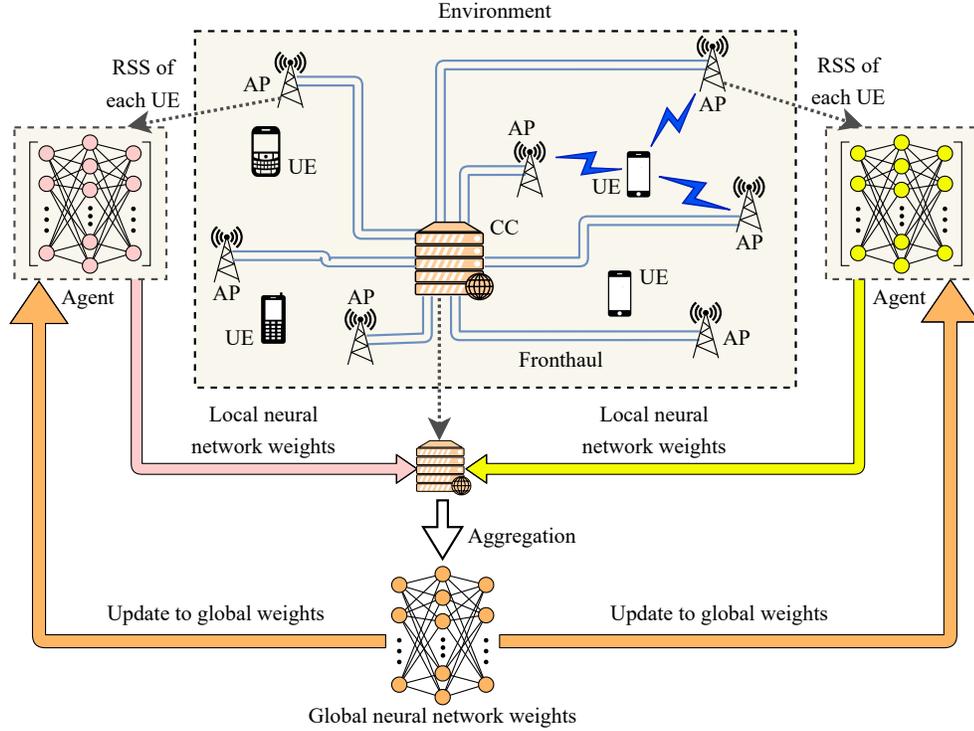


Figure 5.3: Illustration of the MAFRL algorithm’s interactions between the central controller, environment, and agents.

will increase significantly, resulting in smaller SEs for the UEs and thus a lower reward for the AP. Formally, the alternative reward policy is defined as follows:

Policy 5 — Max sum SE for MAFRL: The reward function for agent ℓ at time t is

$$r_{\ell,(t)} = \sum_{k \in \mathcal{D}_{\ell}} \eta_{k,(t)} + P_{G(t)} \left(\left| \bigcup_{\forall \ell} \mathcal{D}_{\ell} \right| \right). \quad (5.20)$$

After T_{FL} episodes, each agent shares its parameter vector θ_{ℓ} (NN weights) with the CC. The CC then aggregates the agents’ parameter vectors and uses them to calculate updated global NN weights. There are various possible methods of doing this (see e.g., [226–228]), but a common way is simply to average θ_{ℓ} over all agents; we use this average in our chapter⁹. The global NN weight update is transmitted back to the APs via fronthaul links. The interaction of different components of the MAFRL system is illustrated in Fig. 5.3.

⁹Averaging may not be the best choice in certain scenarios, such as if there are significant differences in the distribution of data each agent trains with or in the computing capabilities of each agent. In the case of agents training very large DNNs, they may instead send only a portion of their weights to the CC, e.g., for the last few layers. However, since the NNs in our agents are quite small and the network architecture quite homogeneous, we simply average the entire NN weight vectors for the agents at the CC.

Even though our MARL and MAFRL algorithms have many similarities, there are several significant differences between them as well, mostly during training. Notably, APs trained using our MAFRL algorithm will all end up with the same NN weights, whereas each AP trained using our MARL algorithm will end up with different localized NN weights. During MARL training, the CC distributes individual rewards (including possible penalties) to each AP. In contrast, during MAFRL training, in every episode the CC broadcasts the global penalty to all APs. Every T_{FL} episodes, the agents sent their NN weights to the CC, which aggregates them and then broadcasts the updated weights to be used by all APs.

5.4.1 Complexity of MARL and MAFRL decisions

Concerning the complexity of an AP making a decision on which UEs to serve, we note again that the NN of each agent has only three layers: the input and output layers and a single hidden layer. The number of floating point operations (FLOPs) for each AP to make a decision can be calculated as follows. In the hidden layer, at each of the $\Phi+2$ neurons, the $\Phi+2$ input values are multiplied by a weight, then the weighted values are summed. The sum then passes through the $\tanh(\cdot)$ activation function, which requires a bit shift operation (which is simpler than a FLOP) and c_e+3 FLOPs, where c_e is the complexity of calculating e^x of a scalar x (an $\mathcal{O}(1)$ operation). In the output layer, at each of the $\Phi+2$ neurons, the 20 outputs of the hidden layer are again weighted and summed. Then, the vector of those $\Phi+2$ sums is input into the $\text{softmax}(\cdot)$ activation function, which uses $(\Phi+2)(c_e + 2)$ FLOPs. Thus, in total, $\Phi(c_e + 82) + 22c_e + 224$ FLOPs are required to make a decision. As this number of FLOPs is quite low, the proposed MARL and MAFRL algorithms should not be a challenge for practical implementation. Additionally, with only three layers, the delay involved in computation should be sufficiently small for real-time operation. There is of course additional complexity that occurs during training, but this would happen off-line and not during the regular operation of the network.

5.5 Performance Evaluation

In this section, we examine the simulation results of our MARL algorithm for a cell-free massive MIMO network. We consider $L = 40$ 10-m-tall APs with $N = 4$ antennas each that are uniformly distributed over a geographical area of $1 \text{ km} \times 1 \text{ km}$. $K = 20$ single-antenna UEs have their locations initialized uniformly over the area. We assume that the UEs move around the simulation area at a speed¹⁰ of $v = 1 \text{ m/s}$. The direction of each UE is initially selected at random isotropically within the range of angles $[0, 2\pi)$; the UEs move in a straight line afterwards, with the movement wrapped around the edges of the simulation area.

We consider a carrier frequency of 2 GHz and channel bandwidth of 20 MHz. The elements of each AP's antenna array are spaced at half a wavelength at the carrier frequency. We neglect any spatial correlation between the antenna elements, i.e., $\mathbf{R}_{k\ell} = \beta_{k\ell} \mathbf{I}_N, \forall k, \ell$. We set (in dB) $\beta_{k\ell} = -30.5 - 36.7 \log_{10}(d_{k\ell}) + \Omega_{k\ell}$, where the distance $d_{k\ell}$ (in m) accounts for the AP height of 10 m, and $\Omega_{k\ell} \sim \mathcal{N}(0, 4^2)$ is log-normal shadowing [238]. When UEs are initialized at a distance δ from one another, and whenever a UE moves a distance δ , $\Omega_{k\ell}$ is created/updated with a correlation of $2^{-\delta/(9 \text{ m})}$ with the earlier value [238]. The transmitted power of each AP is $P_t = 38 \text{ dBm}$ and the noise power is assumed to be $\sigma^2 = -94 \text{ dBm}$. K orthogonal pilot sequences are available to the UEs, each with length $\tau_p = K$ and power $\rho_p = 100 \text{ mW}$. The discount factor for the MARL algorithm is set to $\gamma = 0.95$, which is a typically-used value (e.g., [248, 249]).

We consider a discrete-time system where for the purpose of AP association, the UEs' positions and channels are updated and sampled¹¹ every 63 ms. Thus, the sampling interval

¹⁰In this chapter, we limit the examination to pedestrian speeds, because considering vehicular speeds would result in the channel estimates becoming increasingly inaccurate. Depending on the carrier frequency and UE speed, the channel coherence time could diminish sufficiently so that the channel could no longer be considered constant within a TDD frame. As such, channel prediction would be needed along with CSI estimation. We have begun to investigate networks with UEs moving at vehicular speeds in some of our other chapter, e.g., [29].

¹¹With a sample period of 63 ms and a UE speed of 1 m/s, the UEs thus move a distance $\delta = 63 \text{ mm}$ when updating $\Omega_{k\ell}$ between samples.

is about the same as the channel coherence time $t_c = 0.423\lambda/v$ [252, Eq. (5.40.c)], where λ is the carrier wavelength. In RL terminology, these samples are the steps, and we consider 80 steps during one episode of training. This corresponds to a UE travel distance of 5.04 m at 1 m/s speed. For this relatively small distance, the assumption of UEs moving in a straight line is reasonable¹². After each episode, the UEs' locations and directions are reset randomly, but the AP locations stay the same. The NN weights for each agent are updated after each episode. For the MAFRL algorithm, the global update of weights at the CC occurs every $T_{FL} = 20$ episodes. Agents are trained for 4000 episodes and thereafter their performance is evaluated for 40 test cases (each being a new episode with the NN weights fixed). We repeat this procedure for 10 independent simulation runs. The performance results averaged over the $40 \times 10 = 400$ total test cases are compared against five existing strategies: a) "All": UEs are served by all the APs, and coordinated centralized precoding is done rather than distributed precoding, thus representing the maximum possible performance; b) "Greedy": each AP serves the N highest-RSS (nearest) UEs; c) the max min SINR method proposed in [211]; d) the κ -means clustering ML algorithm proposed in [217]; e) a modified version of the RL-based power control method proposed in [221]. In the case of [221], the authors had originally considered APs equipped with a single antenna each, and UEs served by all APs with centralized precoding. For a fair performance comparison with the other schemes, we modified the method from [221] for multi-antenna APs with LP-MMSE precoding first by adding the global penalty from our reward policies to its reward function. The use of localized precoding implies that each AP should serve no more than N UEs. However, it does not by itself ensure that the APs learn to serve a maximum of N UEs, because if the power allocated to some UE is very small, the resulting effect on the sum SE would be negligible. Hence, the algorithm would not be able to learn properly whether that action is better or worse. Therefore, we additionally defined a threshold such that if the power

¹²If the distance traveled per UE per episode was longer, alternative models for the UE movement could be more appropriate, such as along a grid in an urban area, along some predefined paths, according to a random walk model (see e.g., [253]), or by a machine-learned model [254]. However, such more complicated UE movement models are not necessary in this article and are outside of its main focus.

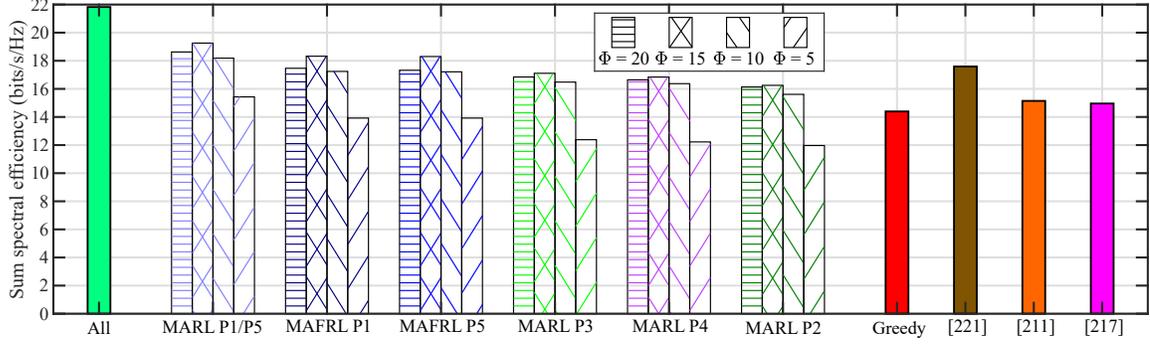


Figure 5.4: Average sum SE performance of MARL and MAFRL algorithms with several values of Φ , compared against “All” and “Greedy” strategies, max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221]. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only.

allocated to a given UE is less than 1% of the AP’s total transmit power, then it is considered that the given UE and that AP are not associated. This threshold for the scheme modified from [221] helps limit the number of served UEs to N .

The average sum SE performance of the MARL and MAFRL algorithms with several values of Φ and the five policies is illustrated in Fig. 5.4. As expected, if UEs are served by all the APs with centralized precoding, then the sum SE is maximum (21.8 bits/s/Hz), but so too is the fronthaul load. Our MARL algorithm under Policy 1 with $\Phi = 10$ and LP-MMSE precoding achieves about 18.2 bits/s/Hz, or about 83.3% of the max SE; this increases to about 88.3% (19.3 bits/s/Hz) using $\Phi = 15$. As seen, increasing Φ improves the sum SE performance, but our algorithms require more training episodes to converge properly. This can be seen in the $\Phi = 20$ result; in this case, 4000 episodes is insufficient for training because of the large action space for $\Phi = 20$. The results of the MAFRL algorithm are similar to, though marginally less than, those provided by MARL algorithm. We first observe that the MAFRL algorithm’s results using Policy 1 and Policy 5 are nearly identical, confirming that the MAFRL algorithm does not need the local penalty as in Policy 1 when maximizing the sum SE. We also observe that the MAFRL algorithm’s sum SE is about 90–95% of (or about 0.95–1.5 bits/s/Hz less than) that of the MARL algorithm. The main reason for the worse MAFRL performance is because the NN weights of the MAFRL

agents are not optimized to their individual locations; rather, the global average is optimized. Thus, the MAFRL algorithm trades off some locally optimized higher performance in favor of consistently good performance over the entire coverage area. Given the similarity in MAFRL performance between Policies 1 and 5, as an additional test, we also checked the performance of Policy 5 when used with the MARL algorithm, even though that policy was designed for the MAFRL algorithm. We found the MARL algorithm performance is also virtually identical for both Policies 1 and 5, which demonstrates that with localized precoding the sum SE reduces when an AP serves more than N UEs. Thus, the agents can learn to serve only N UEs even without the local penalty in the reward term when the goal is to maximize the sum SE. Since the performances of the MARL and MAFRL algorithms are nearly identical under Policy 1 as they are under Policy 5, hereafter we will just depict results for MARL using Policy 1 and MAFRL using Policy 5.

Considering the reference algorithms, we observe that the modified RL-based power control algorithm from [221] performs the best and achieves a sum SE of about 17.6 bits/s/Hz, or about 81% of the “All” case. However, we note that our MARL and MAFRL algorithms employ equal power allocation to all the UEs, yet with $\Phi = 15$ they outperform [221]. This is because the APs learn better to account for UE mobility, whereas the method from [221] does not account for mobility. If we were to incorporate power allocation along with our MARL and MAFRL algorithms, it can be expected that their performance would be further improved. We furthermore note that the higher performance of our algorithms comes via considerably less complex NNs than the NNs used with the method from [221]. For instance, our agent NNs have a single hidden layer with 20 neurons, whereas the (multiple) layers in [221] are of size 400×300 .

We next note that considering just the UEs with the highest RSS values or serving the closest UEs together is far from an optimal AP clustering solution to maximize the sum SE, as observed from the performance for the schemes from [211] and [217], and the $\Phi = 5$ results for MARL under Policy 1 and MAFRL under Policy 5. These respectively achieve

only about 68.5%, 69.4%, 70.7%, and 63.8% of the sum SE of the “All” case. “Greedy” selection performs the worst of the reference schemes, as it is not optimized to maximize the SE globally¹³, though the MAFRL algorithm’s performance with $\Phi = 5$ is the lowest overall among the algorithms intended to maximize the sum SE. The lower the value of Φ , the more localized the selection of UEs is around a given AP, and thus in some sense the whole performance is also based on more localized conditions.

Interestingly, Policies 2 and 3 provide sum SEs not much below that of Policies 1 and 5. They also provide a considerably better sum SE than the scheme from [211], despite all three nominally having a “max min” goal. With $\Phi = 15$, Policy 3 yields a sum SE of about 18.3 bits/s/Hz, while Policy 2 yields a sum SE of about 17.3 bits/s/Hz. The sum SE of Policy 4 is in between that of Policies 2 and 3, about 17.8 bits/s/Hz. It should be noted, though, that Policies 2–4 perform particularly poorly with low values of Φ . For these three policies, Φ must be large enough so that the APs can find the UE with the minimum SE in a wider area around their vicinity. Φ being too small (e.g., $\Phi = 5$) results in the APs considering too small of a neighborhood around their respective locations for the algorithm to properly increase the minimum UE SEs, and thus the sum SE by extension. We lastly note that we also tested Policies 2, 3, and 4 with the MAFRL algorithm, and they displayed a similar small drop in performance relative to the MARL algorithm as was seen with Policies 1 and 5. Therefore, we do not depict the MAFRL results for Policies 2–4, since the drop in performance with Policies 1 and 5 is representative of all the policies.

Fig. 5.4 does not provide insight about the fairness of the MARL and MAFRL algorithms. Therefore, in Fig. 5.5 we examine the cumulative distribution functions (CDFs) of the UEs’ instantaneous SEs for the five reward policies with $\Phi = 15$, compared with

¹³The “Greedy” algorithm selects UEs with the highest RSS values for a given AP. This selection would be the optimal one in terms of maximizing the sum SE if each AP only had one antenna and served a single UE each [255]. However, it is no longer optimal when each AP has multiple antennas and serves multiple UEs, and each UE is served by multiple APs. Instead, other factors such as the orthogonality between the channel vectors of UEs must be considered, on account of the interference their signals cause on each other. A greedy algorithm that maximizes the incremental SE provided by each UE it selects in succession can still provide a near-maximal sum SE. However, using only RSS values, serving the UEs with highest RSS often results in higher multiuser interference, and correspondingly lower sum SE.

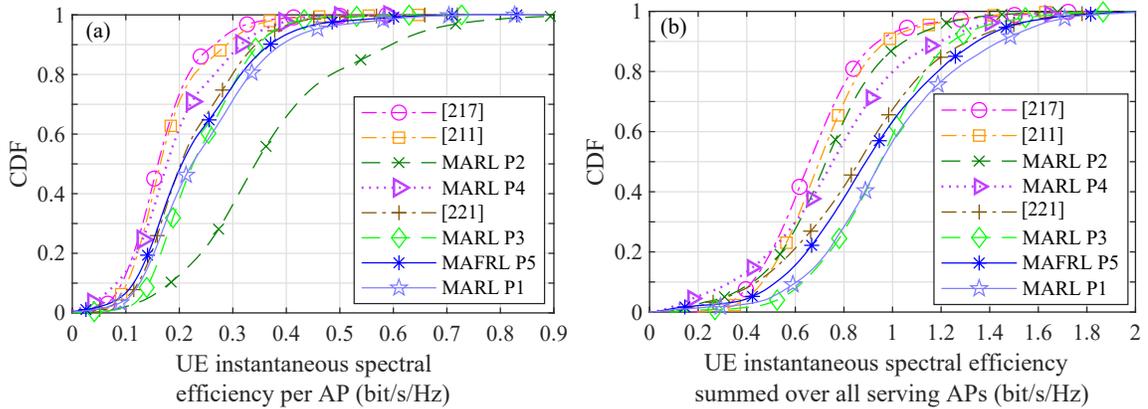


Figure 5.5: CDFs of UEs' instantaneous SEs for MARL and MAFRL algorithms with $\Phi = 15$ under five reward policies, compared with schemes from [211], [217], and [221]. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only. (a) UE SEs per AP. (b) UE SEs summed over their serving APs.

the CDFs for the schemes from [211], [217], and [221]. Fig. 5.5(a) shows the UE SEs per AP, whereas Fig. 5.5(b) shows the UE SEs summed over their respective serving APs. We differentiate between the two because our RL reward policies apply separately at each individual AP, as do the power allocations from [221], whereas [211] and [217] apply more to the system as a whole. We observe that Policies 1 and 5, which maximize the sum SE, provide the highest median and 95th percentile total UE SEs in Fig. 5.5(b), as expected. When each individual AP maximizes the SE of the UEs it serves, that also maximizes the total SE they receive from all their serving APs. The MAFRL algorithm's CDF is slightly to the left of the one for the MARL algorithm, reflecting its slightly worse performance seen in Fig. 5.4. The MARL algorithm under Policy 1 provides median and 95th percentile UE SEs per AP of about 0.22 bits/s/Hz and 0.45 bits/s/Hz, respectively; the median and 95th percentile of the UE total SEs are about 0.95 bits/s/Hz and 1.59 bits/s/Hz, respectively. In comparison, the MAFRL algorithm under Policy 5 yields median and 95th percentile UE SEs per AP of about 0.20 bits/s/Hz and 0.42 bits/s/Hz, respectively; for UE total SEs, they are about 0.89 bits/s/Hz and 1.48 bits/s/Hz, respectively. We also note Policy 3 yields about the same median SE as Policy 1.

The results for Policy 2 are rather unusual compared to the other schemes. Fig. 5.5(a) indicates that this policy results in the highest UE SEs per AP out of any of the schemes. However, this is because under Policy 2, the APs only learn to serve a single UE each at a time, and UEs are served by only one or two APs in total. This behaviour is largely due to the fact that, although the reward of Policy 2 is based on the total SE a UE receives, each AP does not know the specific actions taken at other APs, only the net result of all those actions (including its own). It therefore cannot properly differentiate whether or not the reward is solely due to its own actions. This distinction is irrelevant when maximizing the sum SE as in Policies 1 and 5, but is more important when maximizing the minimum SE. Note that the policy does indeed achieve its goal of each AP maximizing the minimum SE provided to its served UEs (its 5th percentile UE SE per AP is 0.14 bits/s/Hz), but it does so by allocating all its resources to that single UE. This could potentially be problematic with more UEs in the system. In examining Fig. 5.5(b), it can be observed that Policy 2 generally yields the lowest total UE SEs out of our five policies. However, the CDF of UE total SEs for Policy 2 is also one of the steepest out of our five policies, meaning that it provides lower variation / more uniformity in total SEs among the UEs.

In comparison, the modification made in Policy 3 allows the agents to learn to serve multiple UEs, and also results in the highest of the minimum total UE SEs out of all the examined schemes. The 5th percentile total UE SE of Policy 3 is about 0.55 bits/s/Hz. This performance is achieved by trading off the SE given to the higher SE UEs; the upper percentiles are worse for Policy 3 than for Policies 1 and 5. Like Policy 2, Policy 3 also has a steep CDF, meaning that there again is lower variation / higher uniformity in total SEs among the UEs.

The results for Policy 4 indicate that the hybrid reward function does not end up working particularly well at either the lower or the upper end of the CDFs. Neither the sum SE nor the minimum SE ends up maximized. However, Policy 4 does in general perform better than Policy 2 in terms of total UE SEs; their CDFs cross each other at about the 35th percentile.

Policy 4 provides better performance at the middle and upper end of the CDFs, whereas Policy 2 provides better performance at the lower end. We finally note that in regard to our policies, much like for Policies 1 and 5, the MAFRL algorithm provides slightly worse performance than the MARL algorithm under Policies 2–4 as well. (We do not depict the MAFRL algorithm’s performance with Policies 2–4 in Fig. 5.5 in order to avoid obscuring the other results.)

In terms of instantaneous SE performance, among the reference algorithms, the modified RL-based algorithm from [221] provides the highest SEs. Its performance for instantaneous SE per AP in Fig. 5.5(a) is quite close to that of MAFRL under Policy 5, but with a somewhat fairer (less varied) distribution of SEs among UEs. This is likely an indication of the power allocation part of the scheme from [221] diverting power from certain UEs and APs to other ones. The results for SE summed over all APs in Fig. 5.5(b) show more of a difference between the MAFRL algorithm and the modified method from [221]; the summed SEs provided by the former are larger than those of the latter, even for low-SE UEs. This indicates that the modified scheme from [221] likely allocates most power to a UE at a single “best” AP, and significantly less at other APs. In both Figs. 5.5(a) and (b), the SEs provided by the MARL algorithm under Policy 1 are higher than those provided by the modified scheme from [221] across almost the entire distribution. It can lastly be observed from Fig. 5.5 that the CDFs for the schemes from [211] and [217] lie mostly to the left of those of our MARL and MAFRL algorithms. The exception is at the bottom-left of the UE total SE CDFs; those two schemes provide better lower total UE SEs than our Policies 2 and 4. Hence, the schemes from [211] and [217] provide the lowest SEs to most of the UEs out of any of the examined schemes. This reflects the fact that those two schemes also provide the lowest sum SE for the system, as seen in Fig. 5.4.

The convergence of the MARL algorithm can be proven following steps similar to those in [247]. Similarly, [256] provides upper bounds on the convergence rate of FL when global NN updates are calculated as the average of the local NN weights. It is therefore unnecessary

Table 5.1: Sum SE for $\Phi = 10$, Max Sum SE Policies, and Varying Number of Training Episodes

Number of training episodes	MARL Policy 1 Sum SE (bits/s/Hz)	MAFRL Policy 5 Sum SE (bits/s/Hz)
500	10.6	11.1
1000	12.5	12.7
1500	14.2	14.2
2000	14.7	14.5
2500	17.7	16.3
3000	16.9	16.8
3500	17.9	17.2
4000	18.2	17.2
4500	18.2	17.3
5000	18.3	17.3

to duplicate such proofs of convergence here. Instead, in Table 5.1, we compare the MARL and MAFRL performance with $\Phi = 10$ and their max sum SE policies when varying the number of training episodes, to investigate how many episodes are required for the NNs to converge. From the results, we observe that the MARL sum SE oscillates initially, but slowly it stabilizes around 18.2–18.3 bits/s/Hz after about 4000 episodes. The same oscillatory behaviour is seen with the MAFRL algorithm, but it too stabilizes after roughly the same amount of training, this time to around 17.2–17.3 bits/s/Hz. The convergence with Policies 2–4 is similar for both algorithms.

We have also observed that a UE is served by a mean of 3.82 APs with a standard deviation of 0.52 by the MARL algorithm under Policy 1; the results for the MAFRL algorithm under Policy 5 are similar. Under Policy 3, UEs are served by a mean of 3.8 APs with a standard deviation of 0.35 by the MARL algorithm, whereas under Policy 4, UEs are served by a mean of 3.76 APs with a standard deviation of 0.68 by the MARL algorithm. Hence, Policies 3 and 4 are comparable in this regard to Policies 1 and 5, but with Policy 3 having a bit less variation, and Policy 4 a bit more. In sharp contrast, under Policy 2, UEs are only

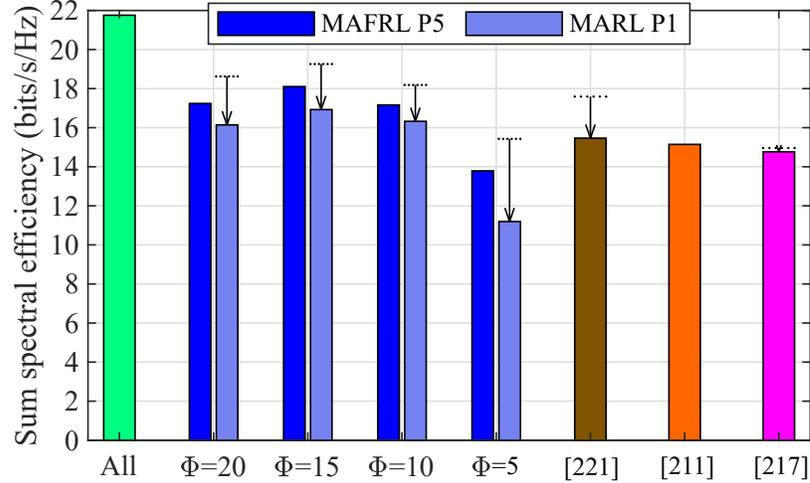


Figure 5.6: Average sum SE performance of MARL and MAFRL algorithms under max sum SE policies with several values of Φ , max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221], when the AP locations during testing differ from those during training. Arrows show drop in MARL performance compared to when the training and testing environments match; performance of MAFRL algorithm is mostly unaffected. Performance of “All” strategy also shown for reference. “P1”: Max sum SE policy having local and global penalties, “P5”: Max sum SE policy having global penalty only.

served by a mean of 1.71 APs, with a standard deviation of 0.24, for the reasons explained earlier. We have additionally confirmed that the set of APs that serves each UE changes as the UEs move through the coverage area. This further demonstrates that the proposed MARL and MAFRL algorithms can indeed properly handle UE mobility while ensuring near-optimal performance, enabling UEs to be connected to 4 APs most of the time (with the exception of Policy 2).

Up to now, the results have suggested that the MARL algorithm outperforms the MAFRL algorithm. This is indeed true when the algorithms are used in the exact same environment they are trained in. However, we next consider a scenario where, after training, the algorithms are transferred to a different environment where the APs’ locations are different than those during training. Specifically, both the APs’ and UEs’ initial locations are randomized, with the results averaged over 400 test cases. This lets us examine how well the algorithms transfer their learning. Fig. 5.6 shows the performance of the MARL and MAFRL algorithms with max sum SE policies, where it can be observed that the MAFRL algorithm now

has a significant advantage. The performance of the MARL algorithm drops considerably in the new environment, between about 1.9 to 4.2 bits/s/Hz, or by 10–27%, compared to its performance in the training environment. In contrast, there is almost no change in the MAFRL performance. Its sum SE drops by at most about 0.2 bits/s/Hz, which is too small for a clear depiction in the figure. Thus, the MARL algorithm’s performance can be highly dependent on the agents’ locations. The $\Phi = 5$ case is particularly vulnerable, because the chosen UEs in that case tend to be those closest to the AP. As such, the agents do not get as broad a sense of the overall UE conditions as they do with higher values of Φ . For the reference algorithms, the scheme from [211] experiences no change in the different environment, which is understandable — its max min SINR metric considers the global environment to begin with. There is also a very small drop in the performance of the scheme from [217] in the new environment, just slightly larger than for our MAFRL algorithm. However, there is a significant dependency¹⁴ of the modified deep RL-based algorithm from [221] on the training environment similar to that of our MARL algorithm. Consequently, that scheme also sees a similar loss in performance in the new environment like our MARL algorithm does.

We next are interested in examining how fast the UEs can move before the performance of our algorithms significantly deteriorates. The sum SE performance for UE speeds of $v = \{1, 1.5, 2, 2.5, 5\}$ m/s is depicted in Fig. 5.7. The first four values cover the range of walking to jogging, while the highest speed that we considered corresponds to a fast run or leisurely bicycle ride. Importantly, the results shown are all for our algorithms having been trained using a UE speed of 1 m/s, but tested on different speeds. We do not show results for the other reference algorithms in this case, because as they do not explicitly account for UE mobility, their performance does not change significantly at the tested speeds; there is just a slight degradation in performance as v increases. From Fig. 5.7, it can be observed that there is no significant change in the performance of our algorithms up to $v = 2.5$ m/s.

¹⁴The authors of [221] have noted this dependency on the training environment in their paper. Their results have circumvented the issue by using data from multiple environments when training.

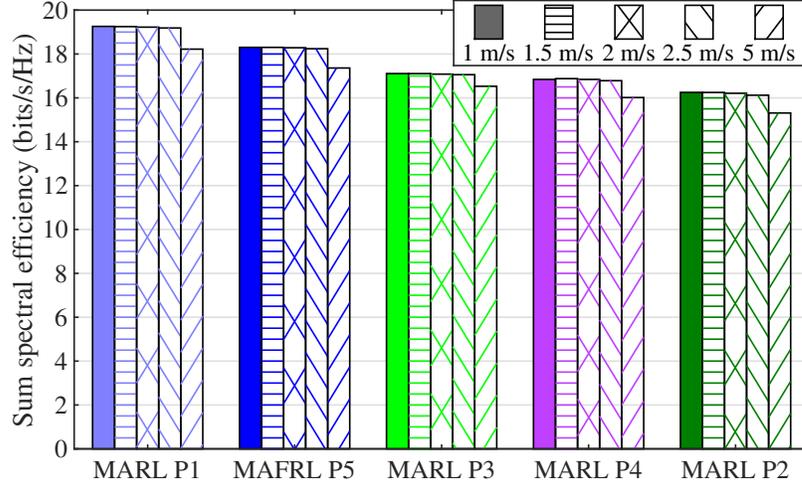


Figure 5.7: Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and several values of UE speed v , having been trained at 1 m/s. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only.

At 2.5 m/s, the drop in performance relative to 1 m/s is less than 1%. It is only at $v = 5$ m/s that a notable deterioration in performance can be seen. In this case, the sum SE drops by about 5–6% for all of the reward policies. Even still, the performance of our MARL algorithm under Policy 1 (18.2 bits/s/Hz at 5 m/s) remains higher than that of the next-best algorithm from [221] (17.6 bits/s/Hz, as seen in previous figures.) The MAFRL performance at 5 m/s under Policy 5 (17.4 bits/s/Hz) is also only slightly worse. We have additionally examined the case of our proposed algorithms being both trained and tested at $v = 5$ m/s. In this event, the algorithms’ performance returns to the same as if they were both trained and tested at 1 m/s. Moreover, training at 5 m/s and testing at 1 m/s again results in a decline in performance; the MARL performance under P1 drops to 18.9 bits/s/Hz, for example. This indicates that the degradation in performance is a result of the mismatch between the training and testing environments (much like what was seen in Fig. 5.6), rather than an inability of the proposed algorithms to handle higher UE speeds. It also suggests that training with a variety of UE speeds ought to result in a bit better performance.

Next, we investigate the impact of varying the ratio of the total number of antenna elements at the APs to the number of UEs on the sum SE performance. For this, we examine

the sum SE performance for two different scenarios: a) varying the number of antennas N per AP while keeping the number of UEs fixed, and b) varying the number of UEs K while keeping the number antennas per AP fixed. The performance when varying N is illustrated in Fig. 5.8(a). We vary N from 2 to 6, and compare the performance of the MARL and MAFRL algorithms with the five reward policies, both using $\Phi = 15$, against the same existing strategies as in Fig. 5.4. We observe that the performance of both our proposed algorithms under Policies 1 and 5 is very close to the maximum performance of the “All” case when $N = 2$. This is understandable because for smaller antenna array sizes, the agents/APs serve fewer UEs; thus, the likelihood of making an optimal UE selection is considerably higher since the search space is much smaller. For $N = 2$, the MAFRL and MARL algorithms obtain about 97–98% of the maximum possible SE of the “All” strategy, whereas for $N = 3$, they obtain about 91–92%. For larger values of N , the slopes of the curves stabilize, such that the SE obtained by our algorithms is consistently about 83–89% of the maximum. Compared to the modified algorithm from [221], the performance of MARL algorithm with Policy 1 is about 7–9% better for $N \geq 4$. We again note that our better performance is with equal power distribution among the UEs; the addition of power allocation should further enhance the MARL (and MAFRL) performance. At the same time, for $N \geq 4$, the MARL algorithm’s SE under Policy 1 is at least about 21–23% larger than the other three reference algorithms, while the MAFRL algorithm’s SE under Policy 5 is about 17–19% larger. Policies 3 and 4 provide about the same sum SE as each other (Policy 3 is marginally better) and their performance improves with N at about the same rate as Policies 1 and 5. Policy 2 continues to have the worst performance among our reward policies, though it remains better than that of all the reference schemes other than the modified one from [221]. Interestingly, although not depicted in the figure, we have observed that the performance of Policy 2 improves slower with N when $\Phi = 10$ is used, such that the scheme from [211] catches up at $N = 6$ in that case.

The performance when varying the number of UEs K is illustrated in Fig. 5.8(b). In this

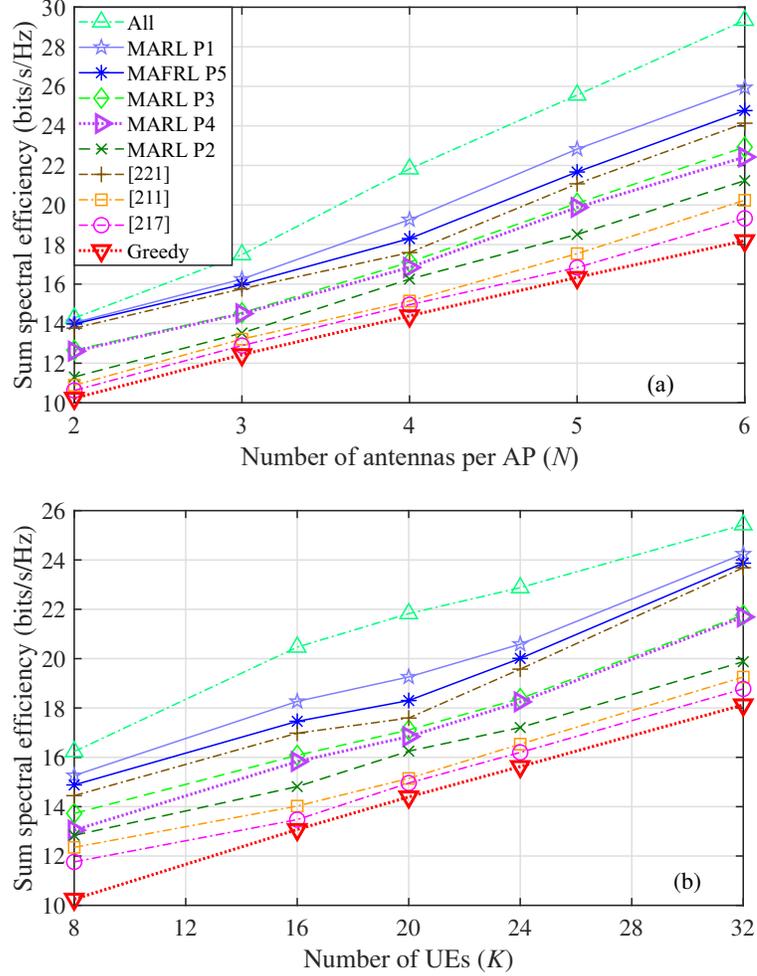


Figure 5.8: Average sum SE performance of MARL and MAFRL algorithms with $\Phi = 15$ and varying ratios of total number of AP antennas to number of UEs. Performance is compared against “All” and “Greedy” strategies, max min SINR strategy from [211], κ -means clustering ML algorithm from [217], and modified RL-based power control algorithm from [221]. “P1”: Max sum SE policy having local and global penalties, “P2”: Max min SE policy, “P3”: Modified max min SE policy, “P4”: Hybrid policy, “P5”: Max sum SE policy having global penalty only. (a) $K = 20$ UEs, varying numbers of antennas per AP (N). (b) $N = 4$, varying number of UEs (K).

case, we vary the number of UEs from 8 to 32 in steps of 8; we also include the previous results for 20 UEs. The relative performance of all the compared algorithms remains the same as in Fig. 5.8(a). Moreover, the rate in increase of SE with K for the SE-maximizing algorithms is roughly the same among that group, as is the rate of increase among the max-min SE algorithms. Unsurprisingly, the SE grows slower with K for the latter group than it does for the former. This simply reflects that the SE-maximizing algorithms can better exploit multiuser diversity. In contrast, the max-min SE algorithms have to trade off some

total SE to improve the performance of additional UEs in the system in relatively poorer channel conditions; hence, their sum SE cannot increase as quickly with K .

5.6 Summary

In this chapter, we have proposed MARL and MAFRL AP clustering algorithms for cell-free massive MIMO systems. We have described the mathematical details for obtaining the CSI and precoding vectors for each AP. The proposed algorithms' performance has been examined for five reward policies and compared with several existing strategies. It has been demonstrated that our MARL algorithm outperforms the other AP clustering strategies, and achieves up to 88.3% of the maximum possible sum SE achievable if all APs were to serve all UEs using centralized precoding. Our MAFRL algorithm performs slightly worse than our MARL algorithm (about 5–10% lower SE) on account of trading off some localized performance gains in favor of uniformly good performance across the entire coverage area. However, that tradeoff also means the MAFRL algorithm can transfer its learning to different environments much better than the MARL algorithm; the latter instead tends to develop dependencies on the training environment. When the AP locations are different during testing than they were during training, the MARL algorithm performance drops significantly (up to 27% lower SE in one case), whereas the MAFRL performance is almost unchanged. A similar but much smaller drop in performance (about 5–6%) occurs for both algorithms when the UE speeds during testing differ significantly from those used for training (e.g., 5 m/s vs. 1 m/s or vice versa). The relative performances of all the examined algorithms also remain about the same when the number of antennas per AP is equal to or greater than 4 and when the number of UEs varies between 8 and 32.

Chapter 6

Machine Learning Assisted DL CSI Estimation for High-Mobility Multi-Antenna Users

6.1 Introduction

In current 5G systems, TDD mode is most commonly considered for massive MIMO systems to exploit radio channel reciprocity between the UL and DL. In TDD mode, the UEs send pilot signals to the BSs, which use those pilots to estimate the UL channels. Thereafter, the BSs use these estimates on the DL channel for purposes such as precoding [3]. Pilot contamination is a known problem in massive MIMO systems [257], usually caused by an insufficient number of orthogonal pilot sequences for all UEs in the system. As a result, certain UEs must share pilot sequences, leading to interference in pilot transmissions between those UEs and corruption of their channel estimates. Moreover, it is most commonly assumed in the massive MIMO literature that the UEs are equipped with a single antenna. However, in practice it is quite common for modern cellphones to be equipped with multiple receive antennas. A unique pilot sequence must be assigned to and sent from each UE antenna in order to differentiate between them at the BS, which exacerbates the pilot contamination problem. Furthermore, normally there are also power constraints on the signals (including the pilots) sent by the UEs. For these reasons, the alternative of incomplete CSI availability may be necessary, such as by UEs not sending pilots sequences from all their

antennas. Also, the evolution from 5G to 5G-Advanced networks is expected to lead to better user mobility support and higher power efficiency [258]. This too can be supported by limiting the auxiliary transmissions used for CSI acquisition.

6.1.1 Background & Motivation

The related topic of acquiring CSI using methods with limited feedback (where the UE measures the DL channel and returns information about it to the BS) has been investigated in numerous articles, e.g., [259–262]. In [259], sparse representation and compressive sensing techniques are discussed. A modulation and beamforming technique has been developed in [260] that exploits the spatial correlation between antennas. The authors of [261] have proposed an antenna selection algorithm based on Thompson sampling and multi-armed bandit ML, where CSI is only available for the selected antennas. [262] and several references therein use deep learning techniques to create the feedback. Most existing methods generally require some sort of processing or encoding of the CSI at the UE side to create the appropriate type of limited feedback, and then decoding said feedback at the BS to obtain the CSI. Understandably, it can be difficult to implement these schemes in a practical network.

Moreover, many of these approaches are not suitable for mobile UEs where, along with spatial correlation of channel gains between antenna elements, temporal correlation between time instances is also involved. Compounded upon this is that in high-mobility scenarios, the channel during the DL part of the TDD frame may no longer quite match the measurements made during the UL part of the frame, if the channel changes quickly enough. These changes should be accounted for and compensated for. Methods of estimating the CSI for mobile UEs have been studied in prior literature, e.g., [263–266]. Many recent works (e.g., [267–270]) have considered various machine learning methods for high-mobility scenarios. However, the combination of estimating the DL CSI of highly mobile multi-antenna UEs in a massive MIMO system with limited CSI availability is something that has yet to be

well studied. Upon surveying the literature, we have found only one closely similar work [271], where the CSI has been estimated for single-antenna mobile UEs with partial CSI acquisition of only a subset of BS antennas and TDD frames. In [271], temporal correlation of CSI has been estimated using an autoregressive (AR) model, while complete CSI has been estimated using a specific spatial correlation model. A key limitation of that approach is a lack of robustness of the authors' proposed algorithm — the approach only applies to the one specific correlation model it is designed for. These challenges have motivated us to employ a combination of ML techniques to create a robust algorithm that can be used with multi-antenna UEs as well as with any spatial correlation model.

These challenges have led us to employ a combination of ML techniques to estimate DL CSI while compensating for both the change in the CSI since the UL part of the TDD frame and for incomplete CSI from multi-antenna UEs.

In this chapter, we consider limited CSI availability resulting from UEs sending pilots from only a subset of their antennas. For the spatial dimension, we use a CGAN [67], a promising ML framework, to estimate and interpolate the complete CSI for all UE antennas from the information available. The problem of estimating complete CSI from partial CSI is similar to the problem of complete image generation from a masked image [272, 273]. It has been observed that CGANs perform quite well for this type of problem [274], which has motivated us to employ a CGAN in our scenario. For the temporal dimension, we use LSTM cells, a type of recurrent neural network, to capture the temporal correlation from past UL CSI measurements and predict the CSI at the time of DL transmission. Earlier work has shown that LSTM is very effective for time-series forecasting problems in capturing the dependency between values from different time instances [275, 276]. More recently, a combination of LSTM and CGAN has been employed to generate replicated network attacks [277], which further proves the robustness of the CGAN and LSTM approach. Finally, using simulations, we evaluate the performance of the proposed ML-assisted CSI estimation method using NMSE and sum SE metrics.

6.2 Channel Dataset Generation Methodology

Our massive MIMO system model is largely based on the one in [278]. We assume a single-cell massive MIMO network where the BS is equipped with M antennas and serves K UEs (with $M \gg K$), each equipped with N antennas. Each UE is mobile and moves in a straight line at a constant speed. We assume the channels undergo spatially and temporally correlated Rayleigh fading. The UL channel gain matrix $\mathbf{G}_k \in \mathbb{C}^{M \times N}$ between UE k and the BS can be defined as

$$\mathbf{G}_k = \sqrt{\beta_k} \mathbf{R}_{b,k}^{1/2} \mathbf{G}_{w,k} \mathbf{R}_{u,k}^{1/2}, \quad (6.1)$$

where $\mathbf{R}_{b,k} \in \mathbb{C}^{M \times M}$ is the spatial correlation at the BS side and $\mathbf{R}_{u,k} \in \mathbb{C}^{N \times N}$ is the spatial correlation at UE k 's side. $\mathbf{G}_{w,k} \in \mathbb{C}^{M \times N}$ represents the Rayleigh small-scale fading channel between the BS and UE k , with each element having an independent and identically distributed (i.i.d.) zero-mean, unit-variance, circularly-symmetric complex Gaussian distribution. β_k includes the large-scale fading parameters for UE k , i.e., path loss and lognormal shadow fading. It is assumed UE k knows its covariance matrix $\mathbf{R}_{u,k}$. Let the eigen decomposition of $\mathbf{R}_{u,k}$ be denoted $\mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^H$, where $\mathbf{U}_k \in \mathbb{C}^{N \times N}$ is a unitary matrix containing the eigenvectors and $\mathbf{\Lambda}_k \in \mathbb{R}^{N \times N}$ is a diagonal matrix containing the associated eigenvalues.

For complete pilot-based CSI acquisition at the BS, $\tau = KN$ orthogonal pilot sequences of length τ are required. Let us assume that the pilot sequence matrix for UE k is $\mathbf{F}_k \in \mathbb{C}^{N \times \tau}$, and the maximum pilot transmit power for UE k is ρ_k . Under the power constraint for pilot transmission $\text{tr}(\mathbf{F}_k \mathbf{F}_k^H) \leq \tau \rho_k$, we assume the pilot matrix has the form $\mathbf{F}_k = \mathbf{U}_k \mathbf{L}_k^{1/2} \mathbf{V}_k^T$, where the diagonal matrix $\mathbf{L}_k \in \mathbb{R}^{N \times N}$ defines how the power is distributed to the N dimensions, and $\mathbf{V}_k \in \mathbb{C}^{\tau \times N}$ satisfies $\mathbf{V}_k^H \mathbf{V}_k = \tau \mathbf{I}_N$ and $\mathbf{V}_k^H \mathbf{V}_\ell = \mathbf{0}_N$ if $k \neq \ell$ [278]. From [279], this form for \mathbf{F}_k minimizes the mean squared error of the channel estimation.

The exact CSI can be expressed as [278]

$$\mathbf{H}_k = \sqrt{\beta_k} \underbrace{\mathbf{R}_{b,k}^{1/2} \mathbf{G}_{w,k}}_{\bar{\mathbf{H}}_k} \mathbf{U}_k, \quad (6.2)$$

where $\bar{\mathbf{H}}_k$ represents the spatially-correlated small-scale part of the CSI. Under a temporal AR model for the CSI, we assume the matrix given by (6.2) is the initial value. We then use an AR model to generate the small-scale part of the CSI for successive time instances. For an AR(P) process of order P , the small-scale part of the CSI at a given time instance can be modeled as a function of that of the P previous time instances [280, 281]:

$$\bar{\mathbf{h}}_k^{(t)} = \sum_{p=1}^P \Phi_{p,k} \bar{\mathbf{h}}_k^{(t-p)} + \mathbf{w}_k, \quad (6.3)$$

where $\bar{\mathbf{h}}_k^{(t)} = \text{vec}(\bar{\mathbf{H}}_k^{(t)})$ is the small-scale part of the CSI at time instance t , $\Phi_{p,k}$ is the p th AR coefficient for UE k , and $\mathbf{w}_k \sim \mathcal{CN}(0, \sigma_k^2(\mathbf{I}_N \otimes \mathbf{R}_{b,k}))$ is the innovation process [281].

Note that the AR(P) process for UE k involves $P+1$ unknowns: the P AR coefficients $\Phi_{p,k}$ and the variance σ_k^2 of the innovation process. We use the Yule-Walker equations method to derive the values of these unknowns [282]. The Yule-Walker equations can be represented in vector-matrix form by

$$[R_k(1) \ R_k(2) \ \cdots \ R_k(P)]^T = \bar{\mathbf{R}}_k [\Phi_{1,k} \ \Phi_{2,k} \ \cdots \ \Phi_{P,k}]^T, \quad (6.4)$$

where

$$\bar{\mathbf{R}}_k = \begin{bmatrix} R_k(0) & R_k(1) & \cdots & R_k(P-1) \\ R_k(1) & R_k(0) & \cdots & R_k(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_k(P-1) & R_k(P-2) & \cdots & R_k(0) \end{bmatrix}. \quad (6.5)$$

Here, $R_k(p)$ is the temporal autocorrelation function (ACF) between the CSI at time instance t and $t-p$. For a Rayleigh fading channel, the ACF is given by [283]

$$R_k(p) = J_0(2\pi p f_{D,k} T_s), \quad p \geq 0. \quad (6.6)$$

$J_0(\cdot)$ is the zero-order Bessel function of the first kind, T_s is the sampling period, and $f_{D,k}$ is the Doppler frequency shift for UE k , given by $f_{D,k} = \frac{v_k f_c}{c}$; v_k is the speed of UE k , f_c

is the carrier frequency, and c is the speed of light. The AR coefficients are obtained by solving (6.4), after which σ_k^2 is found by

$$\sigma_k^2 = R_k(0) - \sum_{p=1}^P \Phi_{p,k} R_k(p). \quad (6.7)$$

Plugging these values into (6.3), the small-scale part of the CSI for P successive time instances can then be generated.

Meanwhile, the two components of the large-scale fading β_k are updated separately. The path loss at time instance t is updated based on UE k 's distance d_k to the BS at that time. The lognormal shadowing component $\Omega_k^{(t)}$ with standard deviation σ_Ω is updated using Gudmunson's exponential correlation model [284, Ch. 2.6.1],[285]:

$$\Omega_k^{(t)}[\text{dB}] = \zeta_k \Omega_k^{(t-1)} + \sqrt{1 - \zeta_k^2} v_k^{(t)}, \quad (6.8)$$

where v_k is an independent zero-mean Gaussian random variable with the same standard deviation σ_Ω . ζ_k is the shadowing correlation parameter for UE k , given by

$$\zeta_k = \exp(-\Delta_k/d_{corr}), \quad (6.9)$$

where Δ_k is the distance UE k moves between time instance $t-1$ and t , and d_{corr} is the shadowing correlation distance of the environment.

The UL CSI for the UEs is estimated at the BS using the received pilot transmissions. The pilot signal $\mathbf{Y} \in \mathbb{C}^{M \times \tau}$ received at the BS can be expressed as

$$\mathbf{Y} = \sum_{k=1}^K \mathbf{Y}_k = \sum_{k=1}^K \mathbf{G}_k \mathbf{F}_k + \mathbf{N} = \sum_{k=1}^K \mathbf{H}_k (\mathbf{\Lambda}_k \mathbf{L}_k)^{1/2} \mathbf{V}_k^T + \mathbf{N}, \quad (6.10)$$

where $\mathbf{N} \in \mathbb{C}^{M \times \tau}$ is a noise matrix with each entry being i.i.d. complex Gaussian with zero mean and variance σ_n^2 . The minimum mean squared error (MMSE) estimate of the CSI $\hat{\mathbf{h}}_k = \text{vec}(\hat{\mathbf{H}}_k)$ for UE k is given by [278, 279]

$$\hat{\mathbf{h}}_k = ((\mathbf{\Lambda}_k \mathbf{L}_k)^{1/2} \otimes \mathbf{R}_{b,k}) \left((\mathbf{\Lambda}_k \mathbf{L}_k) \otimes \mathbf{R}_{b,k} + \frac{\sigma_n^2}{\tau} \mathbf{I}_{MN} \right)^{-1} \mathbf{b}_k, \quad (6.11)$$

where $\mathbf{b}_k = \text{vec}(\frac{1}{\tau} \mathbf{Y}_k \mathbf{V}_k^*)$. For UEs that do not send pilot sequences from all their antennas, the columns of their CSI \mathbf{H}_k and $\hat{\mathbf{H}}_k$ corresponding to the antennas without pilots have those values set to all zeros.

Lastly, we define $\hat{\mathbf{H}}^{(t)} = [\hat{\mathbf{H}}_1^{(t)}, \hat{\mathbf{H}}_2^{(t)}, \dots, \hat{\mathbf{H}}_K^{(t)}] \in \mathbb{C}^{M \times KN}$ as the joint CSI matrix for all UEs at time instance t . Such joint matrices for multiple consecutive values of t act as the inputs to our ML algorithm.

6.3 Machine Learning Framework

In this section, the CGAN-LSTM-based ML architecture is discussed. The CGAN is used to capture the spatial correlation between BS and UE antennas, and to generate and interpolate for the CSI missing from UE antennas that don't send pilot sequences. Meanwhile, the LSTM network is used to capture the spatio-temporal correlation¹ between CSI at different time instances and UE locations. Before going into the details of the neural network architecture, we first provide a brief overview of LSTM methods.

In Chapter 4, we successfully used a CGAN algorithm to perform CSI estimation for the DL of a frequency-division-duplex massive MIMO system based on the UL CSI. This has motivated us to employ the same architecture for this problem as well. Details of the CGAN architecture and the method of converting CSI to images and back are identical to that in Chapter 4. The key difference now is that the input to the CGAN generator is images with data missing/masked out from certain pixels (corresponding to UE antennas without pilots), and the discriminator is trained using complete images.

6.3.1 LSTM Overview

LSTM networks have been developed to handle time-series data where correlation between data at nearby time instances exists [287]. Within this context, the architecture of an LSTM

¹If there is low or no correlation between the various variables in question, then no estimation or prediction method of any kind would work well, since the information the system has would say next to nothing about the information that is missing.

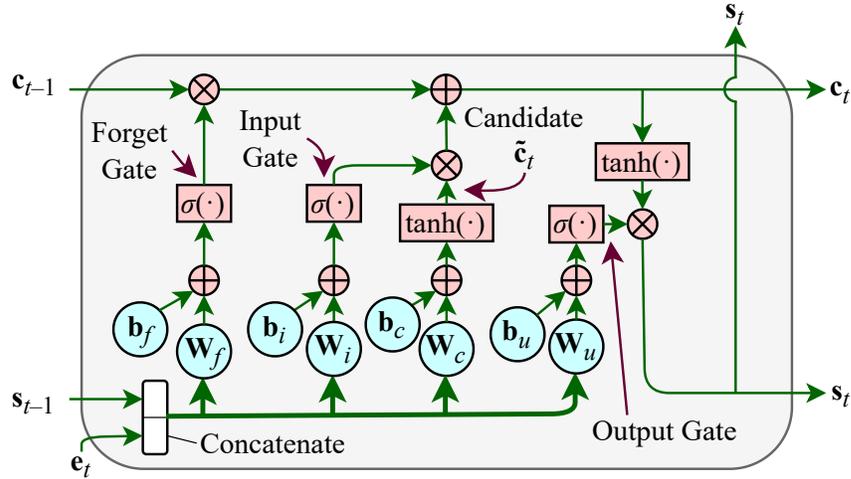


Figure 6.1: Block diagram of LSTM unit (cf. [286]). The addition, multiplication, $\tanh(\cdot)$, and sigmoid ($\sigma(\cdot)$) functions operate on a per-element basis.

network can learn this data correlation over time, thus giving it the ability to predict future data based on previous observations. LSTM networks have been used to solve several communication problems involving sequential data, e.g., [288, 289]. An LSTM unit (depicted in Fig. 6.1) contains several computational blocks known as gates that control and track the flow of information over time. t and $t-1$ respectively denote the current and previous time instances. $e_t \in \mathbb{R}^F$ denotes the current input² (a.k.a. “event”), where F is the number of “features” of the input (e.g., the number of channel gain values). c_{t-1} and c_t are respectively the previous and current long-term memory states, and s_{t-1} and s_t are respectively the previous and current short-term (hidden) memory states; all four have length L , which is also the length of the output vector of each LSTM unit. $\sigma(\cdot)$ is a sigmoid function, specifically, the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$. The various matrices $\mathbf{W} \in \mathbb{R}^{U \times (U+F)}$ and vectors $\mathbf{b} \in \mathbb{R}^U$ are weights and bias values, respectively. The concatenate operation on the left side of Fig. 6.1 vertically stacks s_{t-1} and e_t .

The input gate and candidate state \tilde{c}_t together determine which parts of the information are relevant. The forget gate determines what information to keep as long-term memory.

²This can also be a matrix $\mathbf{E}_t \in \mathbb{R}^{F \times B}$, if the network processes data in batches of size B . The other vector variables then also become matrices accordingly.

Those three components contribute to update the long-term memory c_t , after which the output gate determines the updated short-term memory s_t . Due to space and novelty considerations, for a more detailed description, we refer the reader to [286]. In this chapter, we have considered the traditional architecture of the LSTM unit. More recently, the interaction between the gates of the LSTM unit have been modified and the results thereof investigated [290].

Table 6.1: Parameters of ML Neural Networks

CGAN	
Type of network	Convolutional
Input/output size	$M \times M \times 3$
Kernel size	4×4
Stride size	2×2
Number of layers (generator)	7 encoder, 7 decoder
Number of layers (discriminator)	5 (encoder)
Number of filters per layer (generator encoders)	64, 128, 256, 512, 512, 512, 512
Number of filters per layer (generator decoders)	512, 512, 512, 512, 256, 128, 64
Number of filters per layer (discriminator)	64, 128, 256, 512, 512
Activation functions	Leaky ReLU [183] (encoders), Normal ReLU (decoders)
LSTM	
Number of features (F)	$2MNP$
Number of LSTM units (U)	300
Input size of dense layer	$L \times 1$ with $L = 300$
Output size of dense layer	$2MN \times 1$
Dense layer activation function	Leaky ReLU

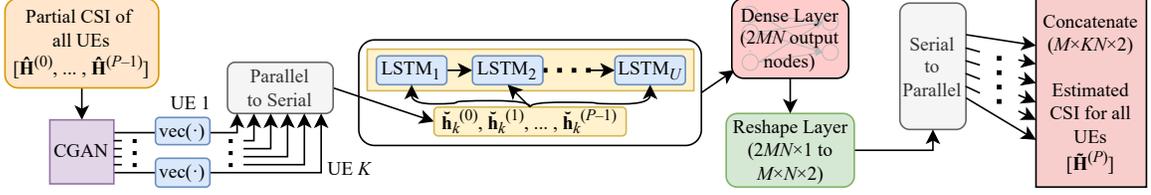


Figure 6.2: Block diagram of CGAN-LSTM-based algorithm for DL CSI estimation. The algorithm works on image pixel values; CSI data matrices are converted to images before the depicted functions, then back to data matrices afterwards. There may alternatively be K identical copies of the LSTM, dense, and reshape layers, with the UE vectors processed in parallel instead of serially.

6.3.2 Overall ML Structure

The overall ML architecture of our CGAN-LSTM-based method is illustrated in Fig. 6.2, while Table 6.1 summarizes the various parameters of the neural networks. The CGAN and LSTM portions are trained separately; thereafter, they are combined to estimate the CSI at the P th time instance. The reason for training the CGAN and LSTM parts separately is due to the unavailability of a suitable loss function to train both together. Once complete CSI for $t = 0$ to $P-1$ is generated by the CGAN, the CSI is split into the matrices for each UE, which are vectorized and sent to the LSTM layer. The LSTM layer and dense layer estimate the CSI for time instance P . To train the LSTM and CGAN, we use an Adam optimizer with a learning rate of 0.001 [291]. The reshape layers convert length- $2MN$ vectors back to two $M \times N$ matrices (the real and imaginary parts of the CSI). Lastly, the $M \times N \times 2$ outputs are concatenated along the 2nd dimension to obtain two $M \times KN$ matrices, the real and imaginary parts of the estimated joint CSI matrix $\tilde{\mathbf{H}}^{(P)}$ for all UEs at time instance P .

6.4 Simulated Performance Evaluation

For performance evaluation, we simulate a circular cellular area with a 500 m radius. The number of BS antennas is varied, with default value $M = 128$, whereas the UEs have a constant $N = 4$ antennas. We consider a carrier frequency of $f_c = 2$ GHz, channel bandwidth of 20 MHz, and two values for the sampling period T_s : 10 μ s and 1 ms. We use the exponential correlation model from [292] for $\mathbf{R}_{b,k}$ and $\mathbf{R}_{u,k}$ with a correlation coefficient of

$0.4e^{j\theta_{b,k}}$ and $0.4e^{j\theta_{u,k}}$, where both $\theta_{b,k}$ and $\theta_{u,k}$ are distributed $\sim \mathcal{U}[0, 2\pi)$. For large-scale fading, we follow the urban microcell non-line-of-sight (UMi NLoS) model in [285]. Thus, the large-scale fading is given in dB by $\Lambda_k = 36.7 \log_{10}(d_k) + 30.5 + \Omega_k$, with d_k in m, and Ω_k having $\sigma_{\Omega} = 4$ dB and correlation distance $d_{corr} = 13$ m; β_k is then given by $10^{-\Lambda_k/10}$. The BS height is assumed to be 10 m when calculating d_k . The BS transmit power is 100 mW and the noise power is $\sigma_n^2 = -94$ dBm. For simplicity, power at each UE is allocated equally to all pilot signals, such that each pilot has power $\rho_k/4 = 0.25$ mW.

The training and test datasets consist of 500 and 100 independent episodes, respectively. In each episode, $K = 10$ UEs are initially placed uniformly over the cell area outside of a circular exclusion area with radius 70 m centered on the BS. Each UE direction is uniformly and isotropically random with a speed chosen at random uniformly between 30–50 km/h. The UEs move in a straight line afterwards, with the movement wrapped around the edges of the simulation area. Complete CSI for $P+1$ time instances ($t = 0$ to P) is generated using the methodology described in Section 6.2. Partial CSI is created by assuming 2 to 4 out of the 10 UEs³, chosen uniformly randomly, each alternate between sending pilots only from antennas #1 and #3, then #2 and #4, in successive time instances. The CGAN is trained with corresponding pairs of partial and complete CSI, whereas the LSTM is trained just with complete CSI. Importantly, note that when training the CGAN, the matrices need not follow a specific order, but when training the LSTM network, the CSI for each UE must be ordered chronologically. Both the CGAN and the LSTM network are trained for 40 epochs.

In Fig. 6.3, the NMSE performance of our proposed CGAN-LSTM method is evaluated. We have plotted the NMSE values for the CGAN-generated CSI as well as the NMSE values of the complete algorithm to investigate which component generates the most error. The

³Only 2 to 4 UEs are chosen to ensure that enough of the CSI image remains for the CGAN to process.

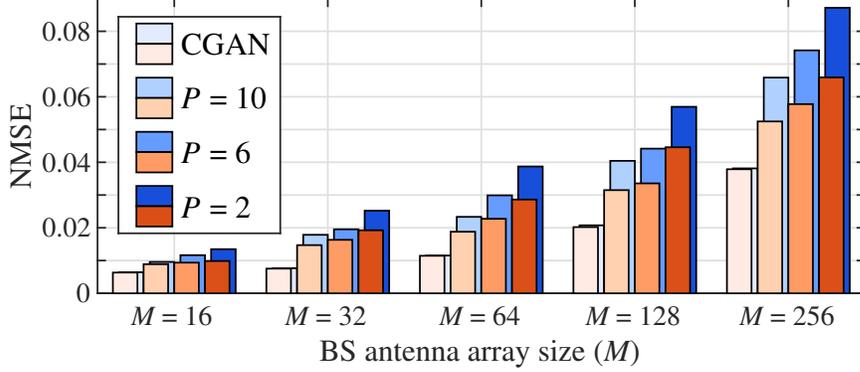


Figure 6.3: NMSE performance of CGAN component alone and of CGAN-LSTM algorithm when an AR(P) process is assumed for the LSTM network. Lower orange bars represent $T_s = 10 \mu\text{s}$; higher blue bars represent $T_s = 1 \text{ ms}$.

NMSE metric per UE is

$$\text{NMSE}_{\text{sys}} = \frac{\|\tilde{\mathbf{H}}_k^{(P)} - \hat{\mathbf{H}}_k^{(P)}\|_F^2}{\|\hat{\mathbf{H}}_k^{(P)}\|_F^2}, \quad (6.12)$$

$$\text{NMSE}_{\text{CGAN}} = \sum_{p=0}^{P-1} \frac{\|\check{\mathbf{H}}_k^{(p)} - \hat{\mathbf{H}}_k^{(p)}\|_F^2}{P \|\hat{\mathbf{H}}_k^{(p)}\|_F^2}, \quad (6.13)$$

where $\tilde{\mathbf{H}}_k^{(P)}$ is the CGAN-LSTM estimated CSI and $\hat{\mathbf{H}}_k^{(P)}$ is the MMSE value of the complete CSI, both at time instance P , and $\check{\mathbf{H}}_k^{(p)}$ is the CGAN's generated CSI for time instance p . The results are generated for varying M and P in the LSTM network. We observe that the NMSE worsens if smaller values of P are used. Please note that CSI for testing is generated always using an AR(10) process, creating CSI for 11 time instances. However, for training and testing the LSTM, we use CSI from the previous P instances. (For example, for $P = 2$, test CSI is from time instance 11 and training CSI is from time instances 9 and 10). The results also show that with $T_s = 10 \mu\text{s}$, the largest portion of the error generally comes from the CGAN module. With $P = 10$, the CGAN accounts for about 51–72% of the NMSE. However, for $P = 2$, this drops to about 39–64%, meaning the LSTM component contributes proportionally more to the increasing NMSE as P decreases. Upon increasing T_s to 1 ms, the NMSE of the CGAN is essentially unaffected, as expected, since the CGAN only estimates missing spatial data for the UE antennas. In contrast, the LSTM contribu-

Table 6.2: NMSE for $U = 300$ LSTM cells, $M = 128$, $T_s = 10 \mu\text{s}$, and varying depth D of LSTM layers

LSTM layer depth (D)	NMSE	LSTM layer depth (D)	NMSE
1	0.0315	5	0.0346
2	0.0316	6	0.1249
3	0.0322	7	0.1327
4	0.0328	10	0.1581

tion to the overall NMSE increases, again as expected due to the decrease in the correlation of CSI realizations spaced further apart in time. With $T_s = 1 \text{ ms}$ and $P = 10$, the CGAN now only accounts for about 43–67% of the total NMSE. Overall, the NMSE is quite low, though it increases with M . This behavior is simply a byproduct of the larger matrices being estimated; the more matrix entries with errors in there are in total, the larger the NMSE will be.

To potentially increase the processing speed, we have investigated varying the depth D of the LSTM layers (a.k.a. stacked LSTM [293]) while keeping the total number of LSTM cells fixed. Table 6.2 shows the NMSE performance of our algorithm at various layer depths. We observe that increasing the depth beyond 5 results in significant performance degradation due to a decreasing number of LSTM layer outputs being mapped in the dense layers. (For depth D , the mapping is from L/D values to $2MN$ output nodes.) The same performance trend with D occurs for other values of M and T_s . As the best result is obtained for a depth of 1, we continue to use that depth for the remainder of the performance evaluation.

In Fig. 6.4, the DL sum SE for varying M is examined, where the BS is assumed to use matched filtering (MF) precoding [278, Eq. (17)–(18)] and equal power allocation to all UEs. We compare the performance of our proposed CGAN-LSTM algorithm with that of having the complete MMSE CSI estimate for time instance P , the method from [271], and mean CSI infilling. Mean CSI infilling means that rather than use the CGAN, the BS fills in the missing CSI for a UE using the mean of the available CSI for its antennas. The

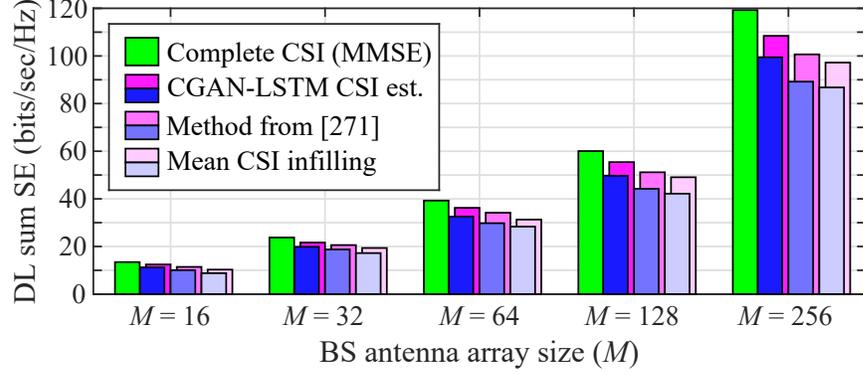


Figure 6.4: DL sum SE performance for MF precoding, comparing complete CSI, our CGAN-LSTM algorithm, the method from [271], and mean CSI infilling. Higher purple bars represent $T_s = 10 \mu s$; lower blue bars represent $T_s = 1 ms$. (Complete CSI is the same for both values of T_s .)

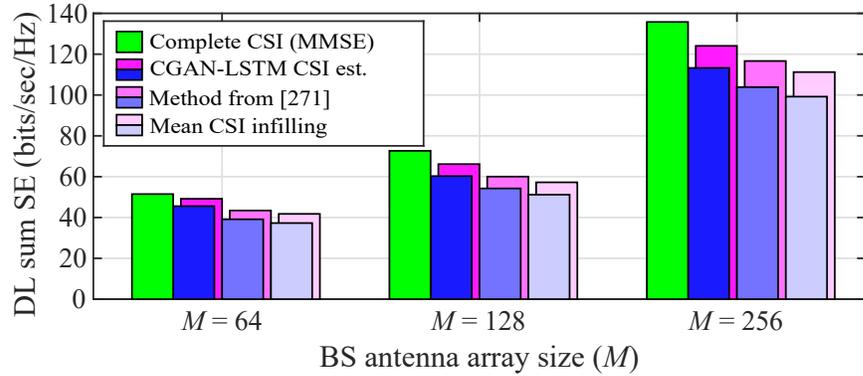


Figure 6.5: DL sum SE performance for BD precoding, comparing complete CSI, our CGAN-LSTM algorithm, the method from [271], and mean CSI infilling. Higher purple bars represent $T_s = 10 \mu s$; lower blue bars represent $T_s = 1 ms$. (Complete CSI is the same for both values of T_s .)

results demonstrate that for $T_s = 10 \mu s$, our CGAN-LSTM-based algorithm achieves about 91–92% of the maximum achievable sum SE when the complete MMSE CSI is available. We also observe that our CGAN-LSTM-based algorithm has a superior performance of about 12–21% higher than mean CSI infilling and about 5–9% higher than the method from [271]. For $T_s = 1 ms$, our algorithm’s performance relative to having complete CSI drops to about 83–84%. However, our performance relative to the other two methods improves slightly (about 15–28% higher than mean CSI infilling and about 6–12% higher than the method from [271]).

More complicated methods of MIMO precoding are known to be more sensitive to CSI errors. Hence, in Fig. 6.5, we also examine the DL sum SE performance of the various

schemes using block-diagonalization (BD) precoding [294] (an extension of zero-forcing precoding for multi-antenna UEs). Interestingly, the simulation results show that although all the SEs increase compared to MF precoding, the relative performance of our algorithm vs. complete CSI is essentially unchanged from that in Fig. 6.4. Our algorithm also gains a few more percentage points relative to the other two methods.

As a final note on complexity, our ML algorithm will initially have a fairly high complexity when training. However, the sizes of the neural networks used are rather small (see Table 6.1). Thus, when in standard operation, the complexity would be comparatively small and capable of supporting real-time operation.

6.5 Summary

In this chapter, we studied a CGAN-LSTM-based algorithm for DL CSI prediction from partial UL CSI for TDD massive MIMO systems with high UE mobility. We have described the details to generate the CSI dataset and outlined the ML architecture. Our algorithm's performance has been examined considering NMSE and DL sum SE metrics. It has been demonstrated that our algorithm achieves low NMSE for its estimates, outperforms the method from [271], and achieves about 91–92% and 83–84% of the maximum possible DL sum SE for CSI sampling periods of $T_s = 10 \mu\text{s}$ and 1 ms, respectively.

Chapter 7

Conclusion

In this chapter, we will discuss some key take-aways of this dissertation and a few promising areas for future work.

7.1 Summary of Contributions

In Chapter 3, we have considered the downlink of a HetNet with massive MIMO MBSs and developed a placement strategy of single-antenna SBSs correlated with the placement of MBSs. Using StoGeo methods, we have developed the analytical model of the HetNet and derived expressions for several performance metrics, i.e., coverage probability, average achievable user rate, and the complementary cumulative distribution function (CDF) for user rates (a.k.a. rate coverage). As part of this process, we have also derived the distributions for the distance between the typical user and its closest SBS, and the distance between the MBS and SBS that are closest to the user. Specifically, the contributions of this chapter are:

- We have developed a StoGeo-based analytical framework for the HetNet with SBSs placed at the vertices of the macrocells. We have also derived the distribution of the distance from the typical user to its closest vertex in a Poisson-Voronoi tessellation. We have also provided two different point process approximations for the placement of SBSs, a simple PPP-based approximation, and a more complicated β -GPP approximation. The latter captures the effect of correlated placement of SBSs.

- We have provided the empirical distribution of the distance between the seed of a Poisson-Voronoi cell and the vertex that is the nearest to the origin of the plane. In the context of a HetNet, this represents the distance between the MBS and SBS that are nearest to the typical user. We have also derived closed-form expressions of the distance distributions for the PPP and β -GPP approximations.
- We have derived analytical expressions for the user association probability, where the association policy is based on the highest biased average received power. We have also derived expressions for the coverage probability and rate coverage.
- We have examined the coverage probability and rate coverage performance of the proposed HetNet model and its approximations when varying several network parameters. Our results indicate an improvement of up to 21% in the coverage probability and up to 28% in the rate coverage compared to the reference SBS placement scheme used in [105], and at least 1.4 times higher coverage probability and 1.22 times higher rate coverage than [39].

In Chapter 4, we have developed a CGAN system for DL channel information estimation from UL channel information using two methods, one using DL CCM estimation and another using DL CSI estimation. Specifically, the contributions of this chapter are:

- We have developed a CGAN framework for DL CCM/CSI estimation from UL CCM/CSI. We have also provided a methodology to convert the UL and DL CCMs and CSI to images that are used to train and operate the CGAN. Furthermore, we have modified the CGAN structure from our earlier work in [25] to improve its operation.
- We have described the analytical details of determining exact CCMs when the mean AoA and angular spread are known. We have also described how to estimate the CCM from the CSI if these parameters are unavailable.
- We have evaluated by simulation the NMSE performance of the proposed CGAN

against existing CCM mapping algorithms for both perfect and imperfect CCM knowledge and observed a significant reduction in NMSE with our method. We have also evaluated both the NMSE and SE performance for CSI estimation and compared it against a typical DNN. In this latter case of CSI estimation, we have performed the evaluation using both simulated and measured CSI data.

In Chapter 5 we have developed conventional and federated multi-agent reinforcement learning for AP clustering in cell-free massive MIMO systems. The specific contributions of this chapter are:

- We have developed a MARL framework for AP clustering in an environment with mobile (i.e., non-stationary) UEs. We have considered UE mobility at pedestrian speeds when creating the simulation environment. We have formulated the problem as a Markov game and then solved it using the “decentralized actor, centralized critic” variant of reinforcement learning. We have developed multiple reward policies to incorporate fair performance.
- We have extended the MARL system to a MAFRL system by introducing FL features. We have described how implementing a MAFRL-based solution can further reduce the communication overhead fronthaul load.
- We have examined the performance of the proposed actor-critic MARL and MAFRL algorithms for UE association and AP clustering via simulations. We have also compared the SE of the MARL and MAFRL algorithms with those of greedy-based AP clustering, ML-based clustering algorithms proposed in [211] and [217], and a modified RL-based downlink (DL) power control algorithm from [221]. As part of this examination, we have illustrated the differences in the performance obtained by extending our MARL algorithm to a MAFRL algorithm. We have demonstrated that the MAFRL performance is somewhat inferior to that of the MARL algorithm in the trained environment, but the MAFRL algorithm also transfers its learning to new en-

vironments more readily and without a notable performance loss, in contrast to the MARL algorithm.

In Chapter 6 we have developed an ML-based system for DL CSI estimation when users are moving at vehicular speed and limited UL CSI is available at the BS. We have assumed that the BSs are equipped with a massive MIMO antenna array and TDD mode of communication is used. Specifically, the contributions of this chapter are:

- We have developed an LSTM-based CSI prediction algorithm for mobile UEs where DL CSI at a given time instance is predicted using UL CSI for earlier instances. This algorithm is suitable when the entire CSI matrices are available for UEs.
- We have also developed a CGAN-based algorithm that can estimate and interpolate the complete CSI matrix for all of a UE's antennas given CSI for only some of its antennas.
- We have combined the LSTM-based and CGAN-based algorithms to develop a framework capable of predicting full DL CSI from partial UL CSI of high-mobility UEs.
- We have simulated and evaluated the performance of our proposed method using NMSE and SE metrics, and have demonstrated it is close to that achievable with full CSI knowledge.

7.2 Possible Future Research Directions

Future research directions for Chapter 3 may involve development of interference mitigation techniques and examination of the network with multi-antenna SBSs. It might also be interesting to extend this SBS placement strategy when HetNets with more than two tiers are considered. This work can be extended to check how different SBS placement strategies perform when interference mitigation or coordinated transmission is used.

The research in Chapter 4 can lead to further analysis and implementation of FDD massive MIMO. While several extensions of this work are possible, immediate follow-up work

could involve tweaking the parameters or structure of the neural networks for better performance. For instance, when converting data to images, the blue color channel for the RGB images is currently unused in our work. The CGAN methodology was initially developed for image-to-image translation; leaving the channel present simplified translating the use of a CGAN to our context of CSI estimation. In the future, the blue channel can be utilized to carry additional information about the channels with the goal of further improving the estimation. The scenario of limited feedback can also be explored; the unused blue channel could play an important role in this case. Similarly, the raw UL pilot symbols could also be used as inputs/data for machine learning. Furthermore, we envision that precoding vectors could be directly estimated, rather than estimating CCMs/CSI as an intermediate step, then precoding from those estimates. Lastly, in the future, it will be important to address how to handle a non-constant number of active UEs when designing, training, and operating the CGAN for direct CSI estimation.

Several extensions of the research in Chapter 5 are possible, such as tweaking the hyperparameters of the agents' NNs for better performance. The operation of the MARL and MAFRL algorithms using measured channel data or a ray-traced environment model could be examined. The performance of other types of ML algorithms can also be studied. In our simulations, the ability to adapt the UE SEs to maximize the rewards of the RL policies is somewhat lessened by the use of equal power allocation for the UEs. It would therefore be useful to examine implementing a power allocation method along with our proposed AP clustering algorithms, possibly together within the same type of MARL/MAFRL framework. Another interesting extension might be investigating a higher-mobility environment, i.e., with much higher UE speeds than the pedestrian speeds considered herein.

Several extensions of the research in Chapter 6 are possible, such as modifying the ML loss function(s), examining the convergence of the proposed algorithm, and investigating how much UL CSI can be missing without the algorithm's performance suffering significantly.

Bibliography

- [1] Ericsson, “Ericsson Mobility Report,” Tech. Rep., Nov. 2022.
- [2] W. Tong and P. Zhu, Eds., *6G: The Next Horizon — From Connected People and Things to Connected Intelligence*. Cambridge, UK: Cambridge Univ. Press, 2021. DOI: 10.1017/9781108989817.
- [3] E. Björnson, J. Hoydis, and L. Sanguinetti, “Massive MIMO networks: Spectral, energy, and hardware efficiency,” *Foundations and Trends in Signal Process.*, vol. 11, no. 3–4, pp. 154–655, 2017.
- [4] J. Hoydis, S. ten Brink, and M. Debbah, “Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013. DOI: 10.1109/JSAC.2013.130205.
- [5] O. Souihli and T. Ohtsuki, “Benefits of rich scattering in MIMO channels: A graph-theoretical perspective,” *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 23–26, Jan. 2013. DOI: 10.1109/LCOMM.2012.120312.121184.
- [6] 3GPP, “Base station (BS) radio transmission and reception (release 15),” 3rd Generation Partnership Project, Tech. Specification TS 38.104 V15.14.0, Sep. 2021.
- [7] Ericsson, “Massive MIMO for 5G networks,” White Paper, Feb. 2023.
- [8] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cells,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017. DOI: 10.1109/TWC.2017.2655515.
- [9] J. Zhang *et al.*, “Cell-free massive MIMO: A new next-generation paradigm,” *IEEE Access*, vol. 7, pp. 99 878–99 888, Jul. 2019. DOI: 10.1109/ACCESS.2019.2930208.
- [10] Ö. T. Demir, E. Björnson, and L. Sanguinetti, “Foundations of user-centric cell-free massive MIMO,” *Foundations and Trends in Signal Process.*, vol. 14, no. 3–4, pp. 162–472, Jan. 2021. DOI: 10.1561/2000000109.
- [11] S. Elhoushy, M. Ibrahim, and W. Hamouda, “Cell-free massive MIMO: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 492–523, Oct. 2022. DOI: 10.1109/COMST.2021.3123267.
- [12] D. Lopez-Perez *et al.*, “Enhanced intercell interference coordination challenges in heterogeneous networks,” *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011. DOI: 10.1109/MWC.2011.5876497.
- [13] Y. Xu and S. Mao, “User association in massive MIMO HetNets,” *IEEE Syst. J.*, vol. 11, no. 1, pp. 7–19, Mar. 2017. DOI: 10.1109/JSYST.2015.2475702.

- [14] H. Ye, G. Y. Li, and B.-H. Juang, “Power of deep learning for channel estimation and signal detection in OFDM systems,” *IEEE Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018. DOI: 10.1109/LWC.2017.2757490.
- [15] M. B. Mashhadi and D. Gündüz, “Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6315–6328, Oct. 2021. DOI: 10.1109/TWC.2021.3073309.
- [16] P. Dong *et al.*, “Deep CNN-based channel estimation for mmWave massive MIMO systems,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019. DOI: 10.1109/JSTSP.2019.2925975.
- [17] Y. Yang *et al.*, “Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities,” *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 22–27, Mar. 2019. DOI: 10.1109/MCOM.2019.1800635.
- [18] B. Cao *et al.*, “Accsinet: Asymmetric convolution-based autoencoder framework for massive MIMO CSI feedback,” *IEEE Commun. Lett.*, vol. 25, no. 12, pp. 3873–3877, Dec. 2021. DOI: 10.1109/LCOMM.2021.3116864.
- [19] H. Shan *et al.*, “Joint sparse autoencoder based massive MIMO CSI feedback,” *IEEE Commun. Lett.*, vol. 27, no. 4, pp. 1150–1154, Apr. 2023. DOI: 10.1109/LCOMM.2023.3250716.
- [20] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, 2nd. Bradford Books, 2018. [Online]. Available: <https://mitpress.mit.edu/9780262039246/reinforcement-learning/>.
- [21] J. Boyan and M. Littman, “Packet routing in dynamically changing networks: A reinforcement learning approach,” in *Advances in Neural Information Processing Systems*, vol. 6, 1993, pp. 671–678. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1993/file/4ea06fbc83cdd0a06020c35d50e1e89a-Paper.pdf.
- [22] S. Singh and D. Bertsekas, “Reinforcement learning for dynamic channel allocation in cellular telephone systems,” in *Advances in Neural Information Processing Systems*, vol. 9, MIT Press, 1996, pp. 974–980. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1996/file/3948ead63a9f2944218de038d8934305-Paper.pdf.
- [23] T. Brown, “Low power wireless communication via reinforcement learning,” in *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, 1999, pp. 893–899. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/54f5f4071faca32ad5285fef87b78646-Paper.pdf.
- [24] B. Banerjee, R. C. Elliott, W. A. Krzymień, and J. Melzer, “Correlated placement of small cell base stations: A coverage enriched HetNet with massive MIMO,” in *Proc. IEEE 91st Veh. Technol. Conf. (VTC2020-Spr.)*, Antwerp, Belgium, May 2020, pp. 1–6. DOI: 10.1109/VTC2020-Spring48590.2020.9128531.

- [25] B. Banerjee, R. C. Elliott, W. A. Krzymień, and H. Farmanbar, “Towards FDD massive MIMO: Downlink channel covariance matrix estimation using conditional generative adversarial networks,” in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Helsinki, Finland (held virtually), Sep. 2021, pp. 940–946. doi: 10.1109/PIMRC50174.2021.9569379.
- [26] B. Banerjee, R. C. Elliott, W. A. Krzymień, and H. Farmanbar, “Downlink channel estimation for FDD massive MIMO using conditional generative adversarial networks,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 122–137, Jan. 2023. doi: 10.1109/TWC.2022.3191605.
- [27] B. Banerjee, R. C. Elliott, W. A. Krzymień, and H. Farmanbar, “Access point clustering in cell-free massive MIMO using multi-agent reinforcement learning,” in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Kyoto, Japan, Sep. 2022, pp. 794–800. doi: 10.1109/PIMRC54779.2022.9977980.
- [28] B. Banerjee, R. C. Elliott, W. A. Krzymień, and M. Medra, “Access point clustering in cell-free massive MIMO using conventional and federated multi-agent reinforcement learning,” *IEEE Trans. Machine Learning Commun. Netw.*, vol. Early Access, pp. 1–1, 2023. doi: 10.1109/TMLCN.2023.3283228.
- [29] B. Banerjee, R. C. Elliott, W. A. Krzymień, and M. Medra, “Machine learning assisted DL CSI estimation for high-mobility multi-antenna users with partial UL CSI availability in TDD massive MIMO systems,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM) Workshops*, Rio de Janeiro, Brazil, Dec. 2022, pp. 1579–1585. doi: 10.1109/GCWkshps56602.2022.10008644.
- [30] B. Błaszczyszyn, M. Haenggi, P. Keeler, and S. Mukherjee, *Stochastic Geometry Analysis of Cellular Networks*. Cambridge, UK: Cambridge Univ. Press, 2018. doi: 10.1017/9781316677339.
- [31] E. Salbaroli and A. Zanella, “Interference analysis in a Poisson field of nodes of finite area,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1776–1783, May 2009. doi: 10.1109/TVT.2008.2003960.
- [32] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, “Modeling and analysis of K -tier downlink heterogeneous cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 550–560, Apr. 2012. doi: 10.1109/JSAC.2012.120405.
- [33] A. He, L. Wang, Y. Chen, M. ElKashlan, and K.-K. Wong, “Massive MIMO in K -tier heterogeneous cellular networks: Coverage and rate,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6. doi: 10.1109/GLOCOM.2015.7417732.
- [34] H. S. Dhillon, M. Kountouris, and J. G. Andrews, “Downlink MIMO HetNets: Modeling, ordering results and performance analysis,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5208–5222, Oct. 2013. doi: 10.1109/TWC.2013.090513.130142.

- [35] A. K. Gupta, H. S. Dhillon, S. Vishwanath, and J. G. Andrews, “Downlink multi-antenna heterogeneous cellular network with load balancing,” *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4052–4067, Nov. 2014. doi: 10.1109/TCOMM.2014.2350504.
- [36] L. Yang, J. Zhao, and Z. Zhang, “Comparative analysis of distance distributions for CoMP-aided cellular networks using Thomas cluster process,” *IEEE Communications Letters*, vol. 25, no. 11, pp. 3503–3507, Nov. 2021. doi: 10.1109/LCOMM.2021.3105283.
- [37] M. Afshang, C. Saha, and H. S. Dhillon, “Nearest-neighbor and contact distance distributions for Matérn cluster process,” *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2686–2689, Dec. 2017. doi: 10.1109/LCOMM.2017.2747510.
- [38] C. Saha, M. Afshang, and H. S. Dhillon, “3GPP-inspired HetNet model using Poisson cluster process: Sum-product functionals and downlink coverage,” *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2219–2234, May 2018. doi: 10.1109/TCOMM.2017.2782741.
- [39] C. Saha, H. S. Dhillon, N. Miyoshi, and J. G. Andrews, “Unified analysis of Het-Nets using Poisson cluster processes under max-power association,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 3797–3812, Aug. 2019. doi: 10.1109/TWC.2019.2917904.
- [40] M. Afshang, C. Saha, and H. S. Dhillon, “Nearest-neighbor and contact distance distributions for Thomas cluster process,” *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 130–133, Feb. 2017. doi: 10.1109/LWC.2016.2641935.
- [41] G. Alfano, M. Garetto, and E. Leonardi, “New directions into the stochastic geometry analysis of dense CSMA networks,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 324–336, Feb. 2014. doi: 10.1109/TMC.2012.248.
- [42] B. Matérn, *Spatial Variation*. New York, USA: Springer, 1986.
- [43] J. Møller, M. L. Huber, and R. L. Wolpert, “Perfect simulation and moment properties for the Matérn type III process,” *Stochastic Processes and their Applications*, vol. 120, no. 11, pp. 2142–2158, Nov. 2010. doi: <https://doi.org/10.1016/j.spa.2010.06.002>.
- [44] M. Kountouris and N. Pappas, “Approximating the interference distribution in large wireless networks,” in *11th Int. Symp. Wireless Commun. Sys. (ISWCS)*, Aug. 2014, pp. 80–84. doi: 10.1109/ISWCS.2014.6933324.
- [45] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.
- [46] S. Willhammer, J. Flordelis, L. Van Der Perre, and F. Tufvesson, “Channel hardening in massive MIMO: Model parameters and experimental assessment,” *IEEE Open J. Commun. Soc.*, vol. 1, pp. 501–512, Apr. 2020. doi: 10.1109/OJCOMS.2020.2987704.

- [47] Z. Chen and E. Björnson, “Channel hardening and favorable propagation in cell-free massive MIMO with stochastic geometry,” *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5205–5219, Nov. 2018. doi: 10.1109/TCOMM.2018.2846272.
- [48] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, “Aspects of favorable propagation in massive MIMO,” in *2nd European Signal Process. Conf. (EUSIPCO)*, Sep. 2014, pp. 76–80.
- [49] B. Hochwald, T. Marzetta, and V. Tarokh, “Multiple-antenna channel hardening and its implications for rate feedback and scheduling,” *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 1893–1909, Sep. 2004. doi: 10.1109/TIT.2004.833345.
- [50] O. Elijah *et al.*, “A comprehensive survey of pilot contamination in massive MIMO-5G system,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 905–923, Nov. 2016. doi: 10.1109/COMST.2015.2504379.
- [51] R. W. Heath Jr. and A. Lozano, *Foundations of MIMO Communication*. Cambridge, UK: Cambridge Univ. Press, 2018. doi: 10.1017/9781139049276.
- [52] H. Q. Ngo *et al.*, “On the total energy efficiency of cell-free massive MIMO,” *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 25–39, Mar. 2018. doi: 10.1109/TGCN.2017.2770215.
- [53] E. Björnson and L. Sanguinetti, “Scalable cell-free massive MIMO systems,” *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020. doi: 10.1109/TCOMM.2020.2987311.
- [54] J. Shi *et al.*, “Power-efficient transmission for user-centric networks with limited fronthaul capacity and computation resource,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5649–5660, Sep. 2020. doi: 10.1109/TCOMM.2020.3002942.
- [55] H. T. Dao and S. Kim, “Effective channel gain-based access point selection in cell-free massive MIMO systems,” *IEEE Access*, vol. 8, pp. 108 127–108 132, Apr. 2020. doi: 10.1109/ACCESS.2020.3001270.
- [56] S. Chen *et al.*, “Structured massive access for scalable cell-free massive MIMO systems,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1086–1100, Apr. 2021. doi: 10.1109/JSAC.2020.3018836.
- [57] C. Wei *et al.*, “User-centric access point selection in cell-free massive MIMO systems: A game-theoretic approach,” *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2225–2229, Sep. 2022. doi: 10.1109/LCOMM.2022.3186350.
- [58] M. Freitas *et al.*, “Matched-decision AP selection for user-centric cell-free massive MIMO networks,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6375–6391, May 2023. doi: 10.1109/TVT.2023.3235980.
- [59] N. Ghiasi *et al.*, “Energy efficient AP selection for cell-free massive MIMO systems: Deep reinforcement learning approach,” *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 29–41, Mar. 2023. doi: 10.1109/TGCN.2022.3196013.
- [60] W. Samek *et al.*, “Explaining deep neural networks and beyond: A review of methods and applications,” *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, Mar. 2021. doi: 10.1109/JPROC.2021.3060483.

- [61] R. Li *et al.*, “Intelligent 5G: When cellular networks meet artificial intelligence,” *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017. DOI: 10.1109/MWC.2017.1600304WC.
- [62] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, “Deep learning in physical layer communications,” *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019. DOI: 10.1109/MWC.2019.1800601.
- [63] L. Dai *et al.*, “Deep learning for wireless communications: An emerging interdisciplinary paradigm,” *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 133–139, Aug. 2020. DOI: 10.1109/MWC.001.1900491.
- [64] M. Chen *et al.*, “Artificial neural networks-based machine learning for wireless networks: A tutorial,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 2019. DOI: 10.1109/COMST.2019.2926625.
- [65] Z. Pan *et al.*, “Recent progress on generative adversarial networks (GANs): A survey,” *IEEE Access*, vol. 7, pp. 36 322–36 333, Mar. 2019. DOI: 10.1109/ACCESS.2019.2905015.
- [66] J. Gui *et al.*, “A review on generative adversarial networks: Algorithms, theory, and applications,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3313–3332, Apr. 2023. DOI: 10.1109/TKDE.2021.3130191.
- [67] I. J. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances Neural Inf. Process. Syst. (NIPS)*, Z. Ghahramani *et al.*, Eds., vol. 27, Curran Associates, Inc., Dec. 2014, pp. 1–9. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- [68] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–16. [Online]. Available: <https://arxiv.org/abs/1511.06434>.
- [69] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5967–5976.
- [70] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *34th Int. Conf. on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, Aug. 2017, pp. 214–223.
- [71] H. Thanh-Tung and T. Tran, “Catastrophic forgetting and mode collapse in GANs,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2020, pp. 1–10. DOI: 10.1109/IJCNN48605.2020.9207181.
- [72] J. Kim, Y. Ahn, and B. Shim, “Massive data generation for deep learning-aided wireless systems using meta learning and generative adversarial network,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1302–1306, Jan. 2023. DOI: 10.1109/TVT.2022.3204835.

- [73] X. Shen, L. Wei, and Y. Xu, "Signal detection method at the OFDM receiver based on conditional GAN," in *IEEE 21st Int. Conf. Commun. and Technology (ICCT)*, Oct. 2021, pp. 1226–1230. DOI: 10.1109/ICCT52962.2021.9658003.
- [74] X. Cui, D. Li, Z. Li, and J. Ou, "A GAN noise modeling based blind denoising method for guided waves," *Elsevier Measurement*, vol. 188, p. 110596, Jan. 2022. DOI: <https://doi.org/10.1016/j.measurement.2021.110596>.
- [75] W. Njima, M. Chafii, and R. M. Shubair, "GAN based data augmentation for indoor localization using labeled and unlabeled data," in *2021 Int. Balkan Conf. on Commun. and Netw. (BalkanCom)*, Sep. 2021, pp. 36–39. DOI: 10.1109/BalkanCom53780.2021.9593240.
- [76] R. M. Schmidt, "Recurrent neural networks (RNNs): A gentle introduction and overview," *CoRR*, vol. abs/1912.05911, pp. 1–16, 2019. [Online]. Available: {<https://arxiv.org/abs/1912.05911>}.
- [77] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ACM 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, 2013, III–1310–III–1318.
- [78] K. Greff *et al.*, "LSTM: a search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017. DOI: 10.1109/TNNLS.2016.2582924.
- [79] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: ACM, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.
- [80] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019. DOI: 10.1162/neco_a_01199.
- [81] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997. DOI: 10.1109/78.650093.
- [82] C. Sun *et al.*, "A convolutional recurrent neural network with attention framework for speech separation in monaural recordings," *Sci. Rep.*, vol. 11, no. 1, p. 1434, Jan. 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-020-80713-3. [Online]. Available: <https://doi.org/10.1038/s41598-020-80713-3>.
- [83] W. Jiang and H. D. Schotten, "Deep learning for fading channel prediction," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, Mar. 2020. DOI: 10.1109/OJCOMS.2020.2982513.
- [84] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *3rd IEEE Int. Conf. on Computer and Commun. (ICCC)*, Dec. 2017, pp. 695–700. DOI: 10.1109/CompComm.2017.8322633.

- [85] S. Solanki *et al.*, “Spectrum sensing in cognitive radio using CNN-RNN and transfer learning,” *IEEE Access*, vol. 10, pp. 113 482–113 492, Oct. 2022. doi: 10.1109/ACCESS.2022.3216877.
- [86] Q. Wu, Z. Sun, and X. Zhou, “Interference detection and recognition based on signal reconstruction using recurrent neural network,” in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6. doi: 10.1109/GCWkshps45667.2019.9024542.
- [87] S. Jaffry and S. F. Hasan, “Cellular traffic prediction using recurrent neural networks,” in *IEEE 5th Int. Symp. on Telecommun. Technol. (ISTT)*, Nov. 2020, pp. 94–98. doi: 10.1109/ISTT50966.2020.9279373.
- [88] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020. doi: 10.1109/MSP.2020.2975749.
- [89] T. Sun, D. Li, and B. Wang, “Decentralized federated averaging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4289–4301, Apr. 2023. doi: 10.1109/TPAMI.2022.3196503.
- [90] Tensorflow, *Tensorflow Federated*, <https://www.tensorflow.org/federated>.
- [91] H.-S. Lee and D.-E. Lee, “Resource allocation in wireless networks with federated learning: Network adaptability and learning acceleration,” *ICT Express*, vol. 8, no. 1, pp. 31–36, Mar. 2022. doi: <https://doi.org/10.1016/j.icte.2022.01.019>.
- [92] Z. Wang, Y. Zhou, Y. Shi, and W. Zhuang, “Interference management for over-the-air federated learning in multi-cell wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2361–2377, Aug. 2022. doi: 10.1109/JSAC.2022.3180799.
- [93] A. M. Elbir and S. Coleri, “Federated learning for hybrid beamforming in mm-Wave massive MIMO,” *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2795–2799, Dec. 2020. doi: 10.1109/LCOMM.2020.3019312.
- [94] Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, “Federated learning-based cooperative spectrum sensing in cognitive radio,” *IEEE Commun. Lett.*, vol. 26, no. 2, pp. 330–334, 2022. doi: 10.1109/LCOMM.2021.3114742.
- [95] H. Tataria *et al.*, “6G wireless systems: Vision, requirements, challenges, insights, and opportunities,” *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, Jul. 2021. doi: 10.1109/JPROC.2021.3061701.
- [96] C. Li, J. Zhang, J. G. Andrews, and K. B. Letaief, “Success probability and area spectral efficiency in multiuser MIMO HetNets,” *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1544–1556, Apr. 2016. doi: 10.1109/TCOMM.2016.2531719.
- [97] T.-X. Zheng, H.-M. Wang, and M. H. Lee, “Multi-antenna transmission in down-link heterogeneous cellular networks under a threshold-based mobile association policy,” *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 244–256, Jan. 2017. doi: 10.1109/TCOMM.2016.2623300.
- [98] G. Hattab and D. Cabric, “Coverage and rate maximization via user association in multi-antenna HetNets,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7441–7455, Nov. 2018. doi: 10.1109/TWC.2018.2867184.

- [99] C. Chen, X. Zhao, and J. Yuan, “Coverage analysis of inter-tier interference cancellation for massive MIMO HetNet with repulsion,” *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 350–353, Feb. 2019. doi: 10.1109/LCOMM.2018.2889328.
- [100] A. Adhikary, H. S. Dhillon, and G. Caire, “Massive-MIMO meets HetNet: Interference coordination through spatial blanking,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1171–1186, Jun. 2015. doi: 10.1109/JSAC.2015.2416986.
- [101] Q. Ye, O. Y. Bursalioglu, H. C. Papadopoulos, C. Caramanis, and J. G. Andrews, “User association and interference management in massive MIMO HetNets,” *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2049–2065, May 2016. doi: 10.1109/TCOMM.2016.2547956.
- [102] B. Li *et al.*, “Energy-efficient resources allocation with millimeter-wave massive MIMO in ultra dense HetNets by SWIPT and CoMP,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4435–4451, Jul. 2021. doi: 10.1109/TWC.2021.3058776.
- [103] K. Khawam, S. Lahoud, M. El Helou, S. Martin, and F. Gang, “Coordinated framework for spectrum allocation and user association in 5G HetNets with mmWave,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1226–1243, Apr. 2022. doi: 10.1109/TMC.2020.3022681.
- [104] H. Wu, X. Tao, J. Xu, and N. Li, “Coverage analysis for CoMP in two-tier HetNets with nonuniformly deployed femtocells,” *IEEE Commun. Lett.*, vol. 19, no. 9, pp. 1600–1603, Sep. 2015. doi: 10.1109/LCOMM.2015.2444836.
- [105] T. X. Tran and K. C. Teh, “Performance analysis of two-tier HetNets with massive MIMO and nonuniformly small cell deployment,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 044–10 054, Nov. 2017. doi: 10.1109/TVT.2017.2744681.
- [106] C. Chen, R. C. Elliott, W. A. Krzymień, and J. Melzer, “Modeling of cellular networks using stationary and nonstationary point processes,” *IEEE Access*, vol. 6, pp. 47 144–47 162, Aug. 2018. doi: 10.1109/ACCESS.2018.2865182.
- [107] N. Deng, W. Zhou, and M. Haenggi, “The Ginibre point process as a model for wireless networks with repulsion,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 107–121, Jan. 2015. doi: 10.1109/TWC.2014.2332335.
- [108] X. Lu, I. Flint, D. Niyato, N. Privault, and P. Wang, “Self-sustainable communications with RF energy harvesting: Ginibre point process modeling and analysis,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1518–1535, May 2016. doi: 10.1109/JSAC.2016.2551538.
- [109] M. Haenggi, “User point processes in cellular networks,” *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 258–261, Apr. 2017. doi: 10.1109/LWC.2017.2668417.
- [110] T. Kwon, H. Ju, and H. Lee, “Performance study for random access-based wireless mutual broadcast networks with Ginibre point processes,” *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1581–1585, Jul. 2020. doi: 10.1109/LCOMM.2020.2987913.
- [111] C. Skouroumounis, C. Psomas, and I. Krikidis, “FD-JCAS techniques for mmWave HetNets: Ginibre point process modeling and analysis,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4352–4366, Dec. 2022. doi: 10.1109/TMC.2021.3076856.

- [112] C. Zhang, F. Yu, Z. Wang, and G. Lyu, “Performance analysis of clustered wireless-powered ad hoc networks via β -Ginibre point processes,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7475–7489, Nov. 2021. doi: 10.1109/TWC.2021.3085247.
- [113] T. Kimura, “Interference analysis in non-Poisson networks under spatially correlated shadowing,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2205–2220, Apr. 2023.
- [114] 3GPP, “Evolved universal terrestrial radio access (E-UTRA); Further advancements for E-UTRA physical layer aspects (release 9),” 3rd Generation Partnership Project, Tech. Rep. 3GPP TR 36.814 V9.2.0, Mar. 2017.
- [115] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, “The multicell multiuser MIMO uplink with very large antenna arrays and a finite-dimensional channel,” *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2350–2361, Jul. 2013. doi: 10.1109/TCOMM.2013.032713.120408.
- [116] H. Huh, A. M. Tulino, and G. Caire, “Network MIMO with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling,” *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2911–2934, May 2012. doi: 10.1109/TIT.2011.2178230.
- [117] H.-S. Jo, Y. J. Sang, P. Xia, and J. G. Andrews, “Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, Oct. 2012. doi: 10.1109/TWC.2012.081612.111361.
- [118] V. Chandrasekhar, M. Kountouris, and J. G. Andrews, “Coverage in multi-antenna two-tier networks,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 10, pp. 5314–5327, Oct. 2009. doi: 10.1109/TWC.2009.090241.
- [119] K. Hosseini, W. Yu, and R. S. Adve, “Large-scale MIMO versus network MIMO for multicell interference mitigation,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 930–941, Oct. 2014. doi: 10.1109/JSTSP.2014.2327594.
- [120] A. He, L. Wang, M. ElKashlan, Y. Chen, and K.-K. Wong, “Spectrum and energy efficiency in massive MIMO enabled HetNets: A stochastic geometry approach,” *IEEE Commun. Lett.*, vol. 19, no. 12, pp. 2294–2297, Dec. 2015. doi: 10.1109/LCOMM.2015.2493060.
- [121] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge, UK: Cambridge Univ. Press, 2012.
- [122] L. Muche, “The Poisson Voronoi tessellation II. edge length distribution functions,” *Mathematische Nachrichten*, vol. 178, no. 1, pp. 271–283, 1996. doi: 10.1002/mana.19961780113.
- [123] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic Geometry and Its Applications*. Chichester, UK: Wiley, 2013.
- [124] J. Møller, “Random tessellations in \mathbb{R}^d ,” *Adv. in Appl. Probability*, vol. 21, no. 1, pp. 37–73, Mar. 1989. doi: 10.2307/1427197.

- [125] I. Nakata and N. Miyoshi, “Spatial stochastic models for analysis of heterogeneous cellular networks with repulsively deployed base stations,” *Perform. Eval.*, vol. 78, pp. 7–17, Aug. 2014. doi: 10.1016/j.peva.2014.05.002.
- [126] G. Chen, L. Qiu, and Z. Chen, “Area spectral efficiency analysis of multi-antenna networks modeled by Ginibre point process,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 6–9, Feb. 2018. doi: 10.1109/LWC.2017.2749572.
- [127] L. Decreusefond, I. Flint, and A. Vergne, “Efficient simulation of the Ginibre point process,” *arXiv*, pp. 1–21, 2013. [Online]. Available: <https://arxiv.org/abs/1310.0800>.
- [128] L. Decreusefond, I. Flint, and A. Vergne, “A note on the simulation of the Ginibre point process,” *J. Appl. Probability*, vol. 52, no. 4, pp. 1003–1012, Dec. 2015. doi: 10.1239/jap/1450802749.
- [129] S. Axler, *Measure, Integration & Real Analysis*. Cham, Switzerland: Springer, 2020. doi: 10.1007/978-3-030-33143-6.
- [130] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 8th, D. Zwillinger and V. Moll, Eds. Waltham, MA, USA: Academic Press, 2014.
- [131] R. W. Heath Jr., M. Kountouris, and T. Bai, “Modeling heterogeneous network interference using Poisson point processes,” *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4114–4126, Aug. 2013. doi: 10.1109/TSP.2013.2262679.
- [132] F. Rusek *et al.*, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.
- [133] 3GPP, “Base station (BS) radio transmission and reception (release 17),” 3rd Generation Partnership Project, Tech. Specification TS 38.104 V17.5.0, Mar. 2022.
- [134] Y. Gu and Y. D. Zhang, “Information-theoretic pilot design for downlink channel estimation in FDD massive MIMO systems,” *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2334–2346, May 2019. doi: 10.1109/TSP.2019.2904018.
- [135] A. Sheikhi, S. M. Razavizadeh, and I. Lee, “A comparison of TDD and FDD massive MIMO systems against smart jamming,” *IEEE Access*, vol. 8, pp. 72 068–72 077, Apr. 2020. doi: 10.1109/ACCESS.2020.2987606.
- [136] W. Peng, W. Li, W. Wang, X. Wei, and T. Jiang, “Downlink channel prediction for time-varying FDD massive MIMO systems,” *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1090–1102, Sep. 2019.
- [137] K. Hugl, K. Kalliola, and J. Laurila, “Spatial reciprocity of uplink and downlink radio channels in FDD systems,” Espoo, Finland, COST 273 TD(02) 066, May 2002, pp. 1–6.
- [138] Z. Zhong, L. Fan, and S. Ge, “FDD massive MIMO uplink and downlink channel reciprocity properties: Full or partial reciprocity?” In *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–5. doi: 10.1109/GLOBECOM42002.2020.9322570.

- [139] R. D. DeGroat, E. M. Dowling, and D. A. Linebarger, "The constrained MUSIC problem," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1445–1449, Mar. 1993.
- [140] F. Li and R. J. Vaccaro, "Analysis of Min-Norm and MUSIC with arbitrary array geometry," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 6, pp. 976–985, Nov. 1990. DOI: 10.1109/7.62249.
- [141] H. Xie, F. Gao, and S. Jin, "An overview of low-rank channel estimation for massive MIMO systems," *IEEE Access*, vol. 4, pp. 7313–7321, Nov. 2016. DOI: 10.1109/ACCESS.2016.2623772.
- [142] H. Xie, B. Wang, F. Gao, and S. Jin, "A full-space spectrum-sharing strategy for massive MIMO cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2537–2549, Oct. 2016.
- [143] H. Xie, F. Gao, S. Zhang, and S. Jin, "A unified transmission strategy for TD-D/FDD massive MIMO systems with spatial basis expansion model," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3170–3184, Apr. 2017.
- [144] E. Björnson, L. Van der Perre, S. Buzzi, and E. G. Larsson, "Massive MIMO in sub-6 GHz and mmWave: Physical, practical, and use-case differences," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 100–108, Apr. 2019. DOI: 10.1109/MWC.2018.1800140.
- [145] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.
- [146] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.
- [147] J. Choi, D. J. Love, and P. Bidigare, "Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 802–814, Oct. 2014.
- [148] Z. Jiang, A. F. Molisch, G. Caire, and Z. Niu, "Achievable rates of FDD massive MIMO systems with spatial channel correlation," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2868–2882, May 2015. DOI: 10.1109/TWC.2015.2396058.
- [149] C. Sun, X. Gao, S. Jin, M. Matthaiou, Z. Ding, and C. Xiao, "Beam division multiple access transmission for massive MIMO communications," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2170–2184, Jun. 2015.
- [150] J. Fang, X. Li, H. Li, and F. Gao, "Low-rank covariance-assisted downlink training and channel estimation for FDD massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1935–1947, Mar. 2017.
- [151] Z. Gao, L. Dai, W. Dai, B. Shim, and Z. Wang, "Structured compressive sensing-based spatio-temporal joint channel estimation for FDD massive MIMO," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 601–617, Feb. 2016.

- [152] P. Liang, J. Fan, W. Shen, Z. Qin, and G. Y. Li, “Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9217–9222, Aug. 2020.
- [153] D. Zhao and T. Han, “Low-complexity compressed sensing downlink channel estimation for multi-antenna terminals in FDD massive MIMO systems,” *IEEE Access*, vol. 8, pp. 130 183–130 193, Jul. 2020. doi: 10.1109/ACCESS.2020.3008175.
- [154] R. W. Heath, Jr., N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave MIMO systems,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.
- [155] P. Dong, H. Zhang, and G. Y. Li, “Machine learning prediction based CSI acquisition for FDD massive MIMO downlink,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647328.
- [156] F. Sahrabi, K. M. Attiah, and W. Yu, “Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4044–4057, Jul. 2021.
- [157] Y. Zhang, X. Zhang, and Y. Liu, “Deep learning based CSI compression and quantization with high compression ratios in FDD massive MIMO systems,” *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2101–2105, Oct. 2021. doi: 10.1109/LWC.2021.3092947.
- [158] J. Zeng *et al.*, “Downlink CSI feedback algorithm with deep transfer learning for FDD massive MIMO systems,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1253–1265, Dec. 2021. doi: 10.1109/TCCN.2021.3084409.
- [159] H. Zhu *et al.*, “Benchmarking and analyzing deep neural network training,” in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Raleigh, NC, USA, Sep. 2018, pp. 88–100. doi: 10.1109/IISWC.2018.8573476.
- [160] H. Xie, F. Gao, S. Jin, J. Fang, and Y.-C. Liang, “Channel estimation for TDD/FDD massive MIMO systems with channel covariance computing,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4206–4218, Jun. 2018.
- [161] Y.-C. Liang and F. P. S. Chin, “Downlink channel covariance matrix (DCCM) estimation and its applications in wireless DS-CDMA systems,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 2, pp. 222–232, Feb. 2001.
- [162] M. Jordan, A. Dimofte, X. Gong, and G. Ascheid, “Conversion from uplink to downlink spatio-temporal correlation with cubic splines,” in *Proc. IEEE 69th Veh. Technol. Conf. (VTC 2009-Spring)*, Barcelona, Spain, Apr. 2009, pp. 1–5.
- [163] A. Decurninge, M. Guillaud, and D. T. M. Slock, “Channel covariance estimation in massive MIMO frequency division duplex systems,” in *Proc. IEEE GLOBECOM Workshops*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

- [164] M. Barzegar Khalilsarai, S. Haghighatshoar, X. Yi, and G. Caire, “FDD massive MIMO via UL/DL channel covariance extrapolation and active channel sparsification,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 121–135, Jan. 2019. doi: 10.1109/TWC.2018.2877684.
- [165] L. Miretti, R. L. G. Cavalcante, and S. Stańczak, “Channel covariance conversion and modelling using infinite dimensional Hilbert spaces,” *IEEE Trans. Signal Process.*, vol. 69, pp. 3145–3159, May 2021. doi: 10.1109/TSP.2021.3082461.
- [166] M. Arnold, S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, “Enabling FDD massive MIMO through deep learning-based channel prediction,” *CoRR*, vol. abs/1901.03664, pp. 1–19, 2019. [Online]. Available: <https://arxiv.org/abs/1901.03664>.
- [167] M. Arnold, S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, “Towards practical FDD massive MIMO: CSI extrapolation driven by deep learning and actual channel measurements,” in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 1972–1976. doi: 10.1109/IEEECONF44664.2019.9048863.
- [168] M. Alrabeiah and A. Alkhateeb, “Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency,” in *Proc. 53rd Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 1465–1470.
- [169] Y. Yang, F. Gao, G. Y. Li, and M. Jian, “Deep learning-based downlink channel prediction for FDD massive MIMO system,” *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019. doi: 10.1109/LCOMM.2019.2934851.
- [170] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, “Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems,” *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7485–7497, Dec. 2020. doi: 10.1109/TCOMM.2020.3019077.
- [171] H. Choi and J. Choi, “Downlink extrapolation for FDD multiple antenna systems through neural network using extracted uplink path gains,” *IEEE Access*, vol. 8, pp. 67 100–67 111, Apr. 2020. doi: 10.1109/ACCESS.2020.2986137.
- [172] X. Li, A. Alkhateeb, and C. Tepedelenlioğlu, “Generative adversarial estimation of channel covariance in vehicular millimeter wave systems,” in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018, pp. 1572–1576. doi: 10.1109/ACSSC.2018.8645463.
- [173] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz (release 16),” 3rd Generation Partnership Project, Tech. Rep. TR 38.901 V17.0.0, Mar. 2022.
- [174] A. Kuchar, M. Tangemann, and E. Bonek, “A real-time DOA-based smart antenna processor,” *IEEE Trans. Veh. Technol.*, vol. 51, no. 6, pp. 1279–1293, Nov. 2002. doi: 10.1109/TVT.2002.801737.
- [175] S. Imtiaz, G. S. Dahman, F. Rusek, and F. Tufvesson, “On the directional reciprocity of uplink and downlink channels in frequency division duplex systems,” in *Proc. 25th IEEE Int. Symp. Pers. Indoor Radio Commun. (PIMRC)*, Washington, DC, USA, Sep. 2014, pp. 172–176. doi: 10.1109/PIMRC.2014.7136154.

- [176] Ericsson, “On channel reciprocity for enhanced DL multi-antenna transmission,” TSG-RAN WG1 Meeting #59b, Valencia, Spain, Tech. Doc. R1-100053, Jan. 2010.
- [177] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Mach. Learn.*, F. Bach and D. Blei, Eds., vol. 37, Lille, France: Proc. Mach. Learn. Res., Jul. 2015, pp. 448–456. [Online]. Available: <http://proceedings.mlr.press/v37/ioffe15.html>.
- [178] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention (MICCAI), Pt. III*, N. Navab et al., Eds., Springer, 2015, pp. 234–241.
- [179] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, “Generating images with recurrent adversarial networks,” *arXiv*, pp. 1–20, 2016. [Online]. Available: <https://arxiv.org/abs/1602.05110>.
- [180] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, <https://www.deeplearningbook.org>.
- [181] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [182] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. Int. Conf. Mach. Learn. Workshop on Deep Learn. for Audio, Speech Lang. Process. (IWCL WDLASL)*, Atlanta, GA, USA, Jun. 2013, pp. 1–6. [Online]. Available: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.
- [183] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” *CoRR*, vol. abs/1505.00853, pp. 1–5, 2015. [Online]. Available: <https://arxiv.org/abs/1505.00853>.
- [184] B. Efron, “Better bootstrap confidence intervals,” *J. Amer. Statistical Assoc.*, vol. 82, no. 397, pp. 171–185, Mar. 1987.
- [185] X. Zhang, L. Zhong, and A. Sabharwal, “Directional training for FDD massive MIMO,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5183–5197, Aug. 2018. doi: 10.1109/TWC.2018.2838600.
- [186] RENEW Project, *RENEW datasets - FDD massive MIMO*. Rice University. <https://renew.rice.edu/dataset-fdd.html> (accessed Jul. 7, 2022).
- [187] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
- [188] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014. doi: 10.1109/JSTSP.2014.2317671.
- [189] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge, UK: Cambridge Univ. Press, 2016.

- [190] W. Choi and J. G. Andrews, “Downlink performance and capacity of distributed antenna systems in a multicell environment,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, pp. 69–73, Jan. 2007.
- [191] X.-H. You *et al.*, “Cooperative distributed antenna systems for mobile communications,” *IEEE Wireless Commun.*, vol. 17, no. 3, pp. 35–43, Jun. 2010. doi: 10.1109/MWC.2010.5490977.
- [192] D. Gesbert *et al.*, “Multi-cell MIMO cooperative networks: A new look at interference,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010. doi: 10.1109/JSAC.2010.101202.
- [193] J.-M. Moon and D.-H. Cho, “Efficient cell-clustering algorithm for inter-cluster interference mitigation in network MIMO systems,” *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 326–328, Mar. 2011. doi: 10.1109/LCOMM.2011.012511.101865.
- [194] R. Irmer *et al.*, “Coordinated multipoint: Concepts, performance, and field trial results,” *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011. doi: 10.1109/MCOM.2011.5706317.
- [195] D. Lee *et al.*, “Coordinated multipoint transmission and reception in LTE-advanced: Deployment scenarios and operational challenges,” *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012. doi: 10.1109/MCOM.2012.6146494.
- [196] J. Lee *et al.*, “Coordinated multipoint transmission and reception in LTE-advanced systems,” *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 44–50, Nov. 2012. doi: 10.1109/MCOM.2012.6353681.
- [197] V. Jungnickel *et al.*, “The role of small cells, coordinated multipoint, and massive MIMO in 5G,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 44–51, May 2014. doi: 10.1109/MCOM.2014.6815892.
- [198] D. Jaramillo-Ramírez, M. Kountouris, and E. Hardouin, “Coordinated multi-point transmission with imperfect CSI and other-cell interference,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 1882–1896, Apr. 2015. doi: 10.1109/TWC.2014.2375874.
- [199] B. Dai and W. Yu, “Sparse beamforming and user-centric clustering for downlink cloud radio access network,” *IEEE Access*, vol. 2, pp. 1326–1339, Oct. 2014. doi: 10.1109/ACCESS.2014.2362860.
- [200] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, “Cloud radio access network (C-RAN): A primer,” *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, Jan. 2015. doi: 10.1109/MNET.2015.7018201.
- [201] A. Checko *et al.*, “Cloud RAN for mobile networks—A technology overview,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 2015. doi: 10.1109/COMST.2014.2355255.
- [202] C. Pan, M. ElKashlan, J. Wang, J. Yuan, and L. Hanzo, “User-centric C-RAN architecture for ultra-dense 5G networks: Challenges and methodologies,” *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 14–20, Jun. 2018. doi: 10.1109/MCOM.2018.1700483.

- [203] M. M. Abdelhakam, M. M. Elmesalawy, K. R. Mahmoud, and I. I. Ibrahim, "A cooperation strategy based on bargaining game for fair user-centric clustering in cloud-RAN," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1454–1457, Jul. 2018. doi: 10.1109/LCOMM.2018.2824320.
- [204] K. T. Truong and R. W. Heath Jr., "The viability of distributed antennas for massive MIMO systems," in *Proc. 47th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2013, pp. 1318–1323. doi: 10.1109/ACSSC.2013.6810508.
- [205] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," in *Proc. 16th Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC 2015)*, Stockholm, Sweden, Jul. 2015, pp. 201–205. doi: 10.1109/SPAWC.2015.7227028.
- [206] G. Femenias and F. Riera-Palou, "Cell-free millimeter-wave massive MIMO systems with limited fronthaul capacity," *IEEE Access*, vol. 7, pp. 44 596–44 612, Mar. 2019.
- [207] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, Dec. 2017. doi: 10.1109/LWC.2017.2734893.
- [208] P. Marsch and G. Fettweis, "Uplink CoMP under a constrained backhaul and imperfect channel knowledge," *IEEE Trans. Wireless Commun.*, vol. 10, no. 6, pp. 1730–1742, Jun. 2011. doi: 10.1109/TWC.2011.041311.100259.
- [209] J. Kang, O. Simeone, J. Kang, and S. Shamai, "Layered downlink precoding for C-RAN systems with full dimensional MIMO," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2170–2182, Mar. 2017. doi: 10.1109/TVT.2016.2572199.
- [210] H. Masoumi and M. J. Emadi, "Performance analysis of cell-free massive MIMO system with limited fronthaul capacity and hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1038–1053, Feb. 2020. doi: 10.1109/TWC.2019.2950316.
- [211] C. F. Mendoza, S. Schwarz, and M. Rupp, "Cluster formation in scalable cell-free massive MIMO networks," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob 2020)*, Thessaloniki, Greece (held virtually), Oct. 2020, pp. 62–67.
- [212] Y. Gao, W. Jiang, and T. Kaiser, "Bidirectional branch and bound based antenna selection in massive MIMO systems," in *Proc. 26th IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC 2015)*, Hong Kong, China, Aug. 2015, pp. 563–568.
- [213] R. Hamdi, E. Driouch, and W. Ajib, "Resource allocation in downlink large-scale MIMO systems," *IEEE Access*, vol. 4, pp. 8303–8316, Nov. 2016.
- [214] M. O. K. Mendonça, P. S. R. Diniz, T. N. Ferreira, and L. Lovisolo, "Antenna selection in massive MIMO based on greedy algorithms," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1868–1881, Mar. 2020. doi: 10.1109/TWC.2019.2959317.

- [215] M. Guo and M. C. Gursoy, “Statistical learning based joint antenna selection and user scheduling for single-cell massive MIMO systems,” *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 471–483, Mar. 2021. doi: 10.1109/TGCN.2020.3033967.
- [216] Y. Xin *et al.*, “Antenna clustering for bidirectional dynamic network with large-scale distributed antenna systems,” *IEEE Access*, vol. 5, pp. 4037–4047, Mar. 2017. doi: 10.1109/ACCESS.2017.2682851.
- [217] S. Biswas and P. Vijayakumar, “AP selection in cell-free massive MIMO system using machine learning algorithm,” in *Proc. 6th Int. Conf. Wireless Commun., Signal Process., Netw. (WiSPNET)*, Chennai, India, Mar. 2021, pp. 158–161. doi: 10.1109/WiSPNET51692.2021.9419450.
- [218] Y. Al-Eryani, M. Akrouf, and E. Hossain, “Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1028–1042, Apr. 2021. doi: 10.1109/JSAC.2020.3018825.
- [219] F. Fredj, Y. Al-Eryani, S. Maghsudi, M. Akrouf, and E. Hossain, “Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1186–1201, Jun. 2022. doi: 10.1109/TCCN.2022.3165810.
- [220] Y. Zhao, I. G. Niemegeers, and S. M. H. De Groot, “Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods,” *IEEE Access*, vol. 9, pp. 102 953–102 965, Jul. 2021. doi: 10.1109/ACCESS.2021.3097243.
- [221] L. Luo *et al.*, “Downlink power control for cell-free massive MIMO with deep reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6772–6777, Jun. 2022. doi: 10.1109/TVT.2022.3162585.
- [222] R. Lowe *et al.*, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proc. 31st Int. Conf. Neural Inf. Process. Systems (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–16. [Online]. Available: <https://arxiv.org/abs/1706.02275>.
- [223] L. Canese *et al.*, “Multi-agent reinforcement learning: A review of challenges and applications,” *Applied Sciences*, vol. 11, no. 11, 2021.
- [224] X. Wang *et al.*, “Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 154–169, Jan. 2021. doi: 10.1109/JSAC.2020.3036946.
- [225] J. Konečný *et al.*, “Federated learning: Strategies for improving communication efficiency,” in *Proc. 30th Conf. Neural Inf. Process. Syst. Workshop on Private Multi-Party Mach. Learn. (NIPS PMPML’16)*, Barcelona, Spain, Dec. 2016, pp. 1–10. [Online]. Available: <https://arxiv.org/abs/1610.05492>.

- [226] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, “Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2021. DOI: 10.1109/COMST.2021.3058573.
- [227] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, “Distributed machine learning for wireless communication networks: Techniques, architectures, and applications,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1458–1493, 2021. DOI: 10.1109/COMST.2021.3086014.
- [228] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 2021. DOI: 10.1109/COMST.2021.3090430.
- [229] R. Ali *et al.*, “A federated reinforcement learning framework for incumbent technologies in beyond 5G networks,” *IEEE Netw.*, vol. 35, no. 4, pp. 152–159, Jul. 2021. DOI: 10.1109/MNET.011.2000611.
- [230] Y. Mu, N. Garg, and T. Ratnarajah, “Federated learning in massive MIMO 6G networks: Convergence analysis and communication-efficient design,” *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4220–4234, Nov. 2022. DOI: 10.1109/TNSE.2022.3196463.
- [231] M. Chen *et al.*, “Distributed learning in wireless networks: Recent progress and future challenges,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021. DOI: 10.1109/JSAC.2021.3118346.
- [232] T. T. Vu *et al.*, “Cell-free massive MIMO for wireless federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020. DOI: 10.1109/TWC.2020.3002988.
- [233] Y.-S. Jeon, M. M. Amiri, J. Li, and H. V. Poor, “A compressive sensing approach for federated learning over massive MIMO communication systems,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1990–2004, Mar. 2021. DOI: 10.1109/TWC.2020.3038407.
- [234] A. M. Elbir and S. Coleri, “Federated learning for channel estimation in conventional and RIS-assisted massive MIMO,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4255–4268, Jun. 2022. DOI: 10.1109/TWC.2021.3128392.
- [235] Y. Guo, Z. Qin, and O. A. Dobre, “Federated generative adversarial networks based channel estimation,” in *Proc. IEEE Int. Conf. Commun. Workshops*, Seoul, Korea, May 2022, pp. 61–66. DOI: 10.1109/ICCWorkshops53468.2022.9814633.
- [236] K.-K. Wong *et al.*, “Truly distributed multicell multi-band multiuser MIMO by synergizing game theory and deep learning,” *IEEE Access*, vol. 9, pp. 30 347–30 358, Feb. 2021. DOI: 10.1109/ACCESS.2021.3059587.

- [237] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, “User access control in open radio access networks: A federated deep reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3721–3736, Jun. 2022. doi: 10.1109/TWC.2021.3123500.
- [238] E. Björnson and L. Sanguinetti, “Making cell-free massive MIMO competitive with MMSE processing and centralized implementation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, Jan. 2020. doi: 10.1109/TWC.2019.2941478.
- [239] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, “Ubiquitous cell-free massive MIMO communications,” *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 197, pp. 1–13, 2019. doi: 10.1186/s13638-019-1507-0.
- [240] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, “User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 611–652, 2022. doi: 10.1109/COMST.2021.3135119.
- [241] A. Hard *et al.*, *Federated learning for mobile keyboard prediction*, arXiv, Feb. 2019. [Online]. Available: <https://arxiv.org/abs/1811.03604>.
- [242] D. Silver *et al.*, “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, pp. 354–359, Oct. 2017. doi: 10.1038/nature24270.
- [243] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015. doi: 10.1038/nature14236.
- [244] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, multi-agent, reinforcement learning for autonomous driving,” *CoRR*, vol. abs/1610.03295, 2016. [Online]. Available: <https://arxiv.org/abs/1610.03295>.
- [245] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Proc. 11th Int. Conf. Mach. Learn.*, W. W. Cohen and H. Hirsh, Eds., Morgan Kaufmann, 1994, pp. 157–163.
- [246] J. G. Kuba *et al.*, “Settling the variance of multi-agent policy gradients,” *CoRR*, vol. abs/2108.08612, 2021. [Online]. Available: <https://arxiv.org/abs/2108.08612>.
- [247] A. Grosnit, D. Cai, and L. Wynter, “Decentralized deterministic multi-agent reinforcement learning,” in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Austin, TX, USA, Dec. 2021, pp. 1548–1553. doi: 10.1109/CDC45484.2021.9683356.
- [248] W. Zhang, X. Wang, J. Shen, and M. Zhou, “Model-based multi-agent policy optimization with adaptive opponent-wise rollouts,” *CoRR*, vol. abs/2105.03363, 2021. [Online]. Available: <https://arxiv.org/abs/2105.03363>.
- [249] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, “Bridging the gap between value and policy based reinforcement learning,” *CoRR*, vol. abs/1702.08892, 2017. [Online]. Available: <https://arxiv.org/abs/1702.08892>.
- [250] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, May 1996. doi: 10.1613/jair.301.

- [251] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, “Federated reinforcement learning with environment heterogeneity,” in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., ser. Proc. Mach. Learn. Res. (PMLR), vol. 151, Mar. 2022, pp. 18–37. [Online]. Available: <https://proceedings.mlr.press/v151/jin22a.html>.
- [252] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [253] S. Hanna, “Random walks in urban graphs: A minimal model of movement,” *Environ. Planning B: Urban Analytics City Sci.*, vol. 48, no. 6, pp. 1697–1711, Jul. 2021. DOI: 10.1177/2399808320946766.
- [254] R. Korbmacher and A. Tordeux, “Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24 126–24 144, Dec. 2022. DOI: 10.1109/TITS.2022.3205676.
- [255] R. Knopp and P. A. Humblet, “Information capacity and power control in single-cell multiuser communications,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 1, Seattle, WA, USA, Jun. 1995, pp. 331–335. DOI: 10.1109/ICC.1995.525188.
- [256] B. Woodworth *et al.*, “Is local SGD better than minibatch SGD?” In *Proc. 37th Int. Conf. Mach. Learn.*, held virtually, Jul. 2020, pp. 10 334–10 343. [Online]. Available: <https://arxiv.org/abs/2002.07839>.
- [257] O. Elijah, C. Y. Leow, T. A. Rahman, S. Nunoo, and S. Z. Iliya, “A comprehensive survey of pilot contamination in massive MIMO—5g system,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 905–923, 2016. DOI: 10.1109/COMST.2015.2504379.
- [258] Nokia, “5G-Advanced: Expanding 5G for the connected world,” White Paper, Mar. 2022. Accessed: Oct. 6, 2022. [Online]. Available: <https://onestore.nokia.com/asset/210984>.
- [259] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, “Sparse representation for wireless communications: A compressive sensing approach,” *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, May 2018. DOI: 10.1109/MSP.2018.2789521.
- [260] M. Maleki, K. Mohamed-Pour, and M. Soltanalian, “Receive spatial modulation in correlated massive MIMO with partial CSI,” *IEEE Trans. Signal Process.*, vol. 67, no. 5, pp. 1237–1250, Mar. 2019. DOI: 10.1109/TSP.2018.2890063.
- [261] Z. Kuai and S. Wang, “Thompson sampling-based antenna selection with partial CSI for TDD massive MIMO systems,” *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7533–7546, Dec. 2020. DOI: 10.1109/TCOMM.2020.3024199.
- [262] G. Fan *et al.*, “Fully convolutional neural network-based CSI limited feedback for FDD massive MIMO systems,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 672–682, Jun. 2022. DOI: 10.1109/TCCN.2021.3119945.
- [263] R. Deng, Z. Jiang, S. Zhou, and Z. Niu, “Intermittent CSI update for massive MIMO systems with heterogeneous user mobility,” *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4811–4824, Jul. 2019. DOI: 10.1109/TCOMM.2019.2911575.

- [264] Y. Zhang, J. Zhang, and L. Yu, "Cluster-based fast time-varying MIMO channel fading prediction in the high-speed scenario," *IEEE Access*, vol. 7, pp. 148 692–148 705, Oct. 2019. doi: 10.1109/ACCESS.2019.2946881.
- [265] X. Xia, K. Xu, S. Zhao, and Y. Wang, "Learning the time-varying massive MIMO channels: Robust estimation and data-aided prediction," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8080–8096, Aug. 2020. doi: 10.1109/TVT.2020.2968637.
- [266] G. Liu, A. Liu, R. Zhang, and M. Zhao, "Angular-domain selective channel tracking and Doppler compensation for high-mobility mmWave massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 2902–2916, May 2021. doi: 10.1109/TWC.2020.3045272.
- [267] H. Hirose, T. Ohtsuki, and G. Gui, "Deep learning-based channel estimation for massive MIMO systems with pilot contamination," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 67–77, Dec. 2021. doi: 10.1109/OJVT.2020.3045470.
- [268] C. Wu *et al.*, "Channel prediction in high-mobility massive MIMO: From spatio-temporal autoregression to deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1915–1930, Jul. 2021. doi: 10.1109/JSAC.2021.3078503.
- [269] W. Luo, W. Ji, Y. Song, and B. Zheng, "Deep learning-based channel estimation approach for 3D massive MIMO millimeter-wave system in time-varying environments," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Xiamen, China, Jul. 2021, pp. 529–533. doi: 10.1109/ICCC52777.2021.9580417.
- [270] S. Zhang *et al.*, "Deep learning-based time-varying channel prediction for MIMO systems," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC 2022-Spring)*, Helsinki, Finland, Jun. 2022, pp. 1–5. doi: 10.1109/VTC2022-Spring54318.2022.9860388.
- [271] J. Qian, C. Masouros, and A. Garcia-Rodriguez, "Partial CSI acquisition for size-constrained massive MIMO systems with user mobility," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 9016–9020, Sep. 2018. doi: 10.1109/TVT.2018.2849263.
- [272] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004. doi: 10.1109/TIP.2004.833105.
- [273] Z. Zhu, H.-Z. Huang, Z.-P. Tan, K. Xu, and S.-M. Hu, "Faithful completion of images of scenic landmarks using Internet images," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 8, pp. 1945–1958, Aug. 2016. doi: 10.1109/TVCG.2015.2480081.
- [274] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, 107:1–107:14, Jul. 2017. doi: 10.1145/3072959.3073659.
- [275] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, Dec. 2018, pp. 1394–1401. doi: 10.1109/ICMLA.2018.00227.

- [276] A. Yadav, C. K. Jha, and A. Sharan, “Optimizing LSTM for time series prediction in Indian stock market,” *Procedia Comput. Sci.*, vol. 167, pp. 2091–2100, 2020. DOI: 10.1016/j.procs.2020.03.257.
- [277] Z. Liu and X. Yin, “LSTM-CGAN: Towards generating low-rate DDoS adversarial samples for blockchain-based wireless network detection models,” *IEEE Access*, vol. 9, pp. 22 616–22 625, Feb. 2021. DOI: 10.1109/ACCESS.2021.3056482.
- [278] X. Li, E. Björnson, S. Zhou, and J. Wang, “Massive MIMO with multi-antenna users: When are additional user antennas beneficial?” In *Proc. 23rd Int. Conf. Telecommun. (ICT)*, Thessaloniki, Greece, May 2016, pp. 1–6. DOI: 10.1109/ICT.2016.7500452.
- [279] E. Björnson and B. Ottersten, “A framework for training-based estimation in arbitrarily correlated Rician MIMO channels with Rician disturbance,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1807–1820, Mar. 2010. DOI: 10.1109/TSP.2009.2037352.
- [280] K. Baddour and N. Beaulieu, “Autoregressive modeling for fading channel simulation,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, Jul. 2005. DOI: 10.1109/TWC.2005.850327.
- [281] H. Kim *et al.*, “Massive MIMO channel prediction: Kalman filtering vs. machine learning,” *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 518–528, Jan. 2021. DOI: 10.1109/TCOMM.2020.3027882.
- [282] S. Haykin, *Adaptive Filter Theory*, 5th. Edinburgh Gate, Essex, England: Pearson, 2013.
- [283] W. C. Jakes, *Microwave Mobile Communications*. New York, NY, USA: Wiley, 1974.
- [284] G. L. Stüber, *Principles of Mobile Communication*, 4th. Cham, Switzerland: Springer, 2017.
- [285] 3GPP, “Further advancements for E-UTRA physical layer aspects (release 9),” 3rd Generation Partnership Project, Tech. Rep. TR 36.814 V9.2.0, Mar. 2017.
- [286] F. M. Salem, *Recurrent Neural Networks: From Simple to Gated Architectures*. Cham, Switzerland: Springer, 2022.
- [287] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.
- [288] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, “Deep learning-based CSI feedback approach for time-varying massive MIMO channels,” *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, Apr. 2019. DOI: 10.1109/LWC.2018.2874264.
- [289] Q. Li, A. Zhang, P. Liu, J. Li, and C. Li, “A novel CSI feedback approach for massive MIMO using LSTM-attention CNN,” *IEEE Access*, vol. 8, pp. 7295–7302, Jan. 2020. DOI: 10.1109/ACCESS.2020.2963896.
- [290] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space odyssey,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017. DOI: 10.1109/TNNLS.2016.2582924.

- [291] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15. [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [292] S. L. Loyka, “Channel capacity of MIMO architecture using the exponential correlation matrix,” *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 369–371, Sep. 2001. doi: 10.1109/4234.951380.
- [293] Z. He *et al.*, “Wider and deeper, cheaper and faster: Tensorized LSTMs for sequence learning,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–15. [Online]. Available: <https://arxiv.org/abs/1711.01577>.
- [294] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, “Zero-forcing methods for down-link spatial multiplexing in multiuser MIMO channels,” *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 461–471, Feb. 2004. doi: 10.1109/TSP.2003.821107.
- [295] J. Reig, L. Rubio, and N. Cardona, “Bivariate Nakagami- m distribution with arbitrary fading parameters,” *Electron. Lett.*, vol. 38, no. 25, pp. 1715–1717, Dec. 2002. doi: 10.1049/el:20021124.
- [296] Z.-H. Yang, Y.-M. Chu, and M.-K. Wang, “Monotonicity criterion for the quotient of power series with applications,” *J. Math. Anal. Appl.*, vol. 428, no. 1, pp. 587–604, Aug. 2015. doi: 10.1016/j.jmaa.2015.03.043.
- [297] National Institute of Standards and Technology. “NIST digital library of mathematical functions.” F. W. J. Olver *et al.*, Eds. version v1.1.9. (Mar. 2023), [Online]. Available: <https://dlmf.nist.gov/>.
- [298] W. Gröbner and N. Hofreiter, *Integraltafel, Zweiter Teil: Bestimmte Integrale*, German, 5th. Wien, Austria: Springer-Verlag, 1973. doi: 10.1007/978-3-7091-7601-6.

Appendix A:

A.1 Proof of Theorem 1 from Chapter 3

The required RV R_{oS} , the distance between the typical user and its nearest SBS, can be derived as a function of R_{oM} and R_{MS} as follows:

$$R_{oS}^2 = R_{oM}^2 + R_{MS}^2 - 2R_{oM}R_{MS} \cos \theta, \quad (\text{A.1})$$

where θ is the angle between the line joining the MBS and the user, and the line joining the MBS and the SBS, respectively. These distances and θ are depicted in Fig. 3.2. The simulated PDF of $\cos \theta$ is shown in Fig. A.1. Note that the distribution of $\cos \theta$ is independent of λ_M because a PPP is isotropic. Varying λ_M has the effect of scaling the distances between points equally in all directions. Therefore, the distribution of angle θ , and by extension the distribution of $\cos \theta$, does not change with λ_M . For similar reasons, the correlation coefficient ρ between R_{oM} and R_{MS} also does not change with λ_M .

In Fig. A.1, a large spike in the distribution can be observed at $\cos \theta = 1$. Consequently, in the following derivations, we will approximate the distribution as a Dirac delta function. This is equivalent to treating $\cos \theta$ as a constant equal to 1.

Now, to determine the distribution of R_{oS} , first we need the joint distribution of R_{oM} and R_{MS} , where R_{oM} and R_{MS} are correlated. As noted in Section 3.3, a Rayleigh distribution is a special case of a Nakagami- m distribution, therefore, the distribution of R_{oM} can be expressed as Nakagami- m with parameters $m_1 = 1$ and $\Omega_1 = 1/(\pi\lambda_M)$. It helps us to use the bivariate Nakagami- m distribution, which is available in the literature [295, Eq. 12] and

is given by

$$\begin{aligned}
f_{R_1, R_2}(r_1, r_2) &= 4(1 - \rho)^{m_2} \sum_{k=0}^{\infty} \left[\frac{(m_1)_k}{k!} \rho^k \left(\frac{m_1/\Omega_1}{(1 - \rho)} \right)^{m_1+k} r_1^{2(m_1+k)-1} \left(\frac{m_2/\Omega_2}{(1 - \rho)} \right)^{m_2+k} \right. \\
&\quad \times r_2^{2(m_2+k)-1} \frac{\exp(-m_1 r_1^2 / [\Omega_1(1 - \rho)]) \exp(-m_2 r_2^2 / [\Omega_2(1 - \rho)])}{\Gamma(m_1 + k) \Gamma(m_2 + k)} \\
&\quad \left. \times {}_1F_1 \left(m_2 - m_1; m_2 + k; \frac{m_2 \rho}{\Omega_2(1 - \rho)} r_2^2 \right) \right], \quad r_1, r_2 \geq 0, \tag{A.2}
\end{aligned}$$

where R_1 has parameters m_1 and Ω_1 and R_2 has parameters m_2 and Ω_2 , with $m_2 \geq m_1$. ρ is the correlation coefficient between R_1 and R_2 , ${}_1F_1(a, b; z)$ is Kummer's confluent hypergeometric function, $(\cdot)_k$ denotes the Pochhammer symbol, and $\Gamma(\cdot)$ is the Gamma function.

Using the joint distribution in (A.2), the cumulative distribution function (CDF) of R_{oS} can be derived as

$$\begin{aligned}
F_{R_{oS}}(r) &= \mathbb{P} \left(\sqrt{R_{oM}^2 + R_{MS}^2 - 2R_{oM}R_{MS} \cos \theta} \leq r \right) \\
&\approx \mathbb{P} \left(\sqrt{R_{oM}^2 + R_{MS}^2 - 2R_{oM}R_{MS}} \leq r \right) \\
&= \mathbb{P} (|R_{MS} - R_{oM}| \leq r) \\
&= \mathbb{P} (-r \leq R_{MS} - R_{oM} \leq r) \\
&= \mathbb{P} (R_{MS} - R_{oM} \leq r) - \mathbb{P} (R_{MS} - R_{oM} \leq -r) \\
&= \int_0^{\infty} \int_0^{r_1+r} f_{R_{oM}, R_{MS}}(r_1, r_2) dr_1 dr_2 - \int_r^{\infty} \int_0^{r_1-r} f_{R_{oM}, R_{MS}}(r_1, r_2) dr_1 dr_2. \tag{A.3}
\end{aligned}$$

To obtain the PDF in Theorem 1, (A.3) is differentiated with respect to r , which leads to

$$f_{R_{oS}}(r) = \int_0^{\infty} f(r_1 + r, r_1) dr_1 + \int_r^{\infty} f(r_1 - r, r_1) dr_1. \tag{A.4}$$

A.2 Proof of Theorem 3 from Chapter 3

To begin, we note the following properties of the regularized lower and upper incomplete Gamma functions, i.e., $\tilde{\gamma}(k, x)$ and $\tilde{\Gamma}(k, x)$, respectively, where $\tilde{\Gamma}(k, x) \triangleq \frac{\int_x^{\infty} e^{-t} t^{k-1} dt}{\Gamma(k)} = 1 - \tilde{\gamma}(k, x)$:

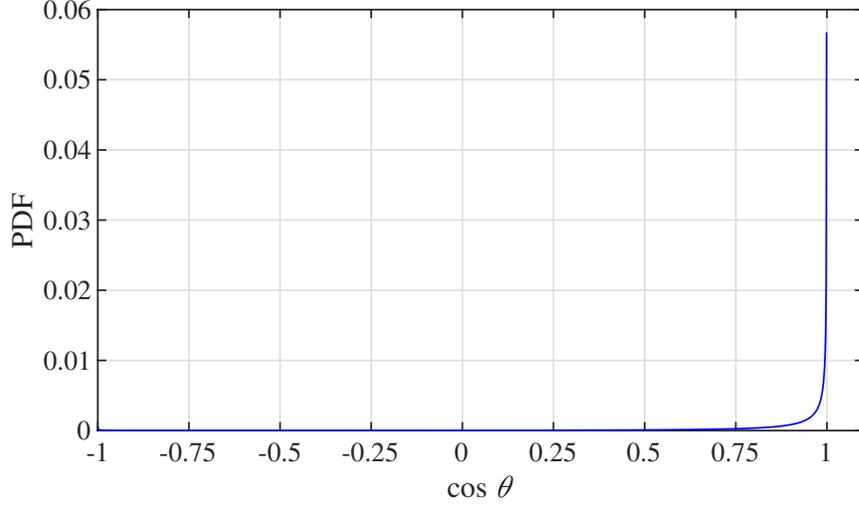


Figure A.1: PDF of $\cos \theta$.

Lemma 8 $\tilde{\Gamma}(k, ck)$ increases monotonically with increasing k if $0 < c \leq 1$. Correspondingly, $\tilde{\gamma}(k, ck)$ decreases monotonically with increasing k if $0 < c \leq 1$.

Proof. $\tilde{\gamma}(k, ck)$ can also be written as [130, Eq. 8.351.4]

$$\tilde{\gamma}(k, ck) = \frac{(ck)^k e^{-ck}}{\Gamma(k+1)} {}_1F_1(1; k+1; ck) \quad (\text{A.5a})$$

$$= \frac{(ck)^k}{\Gamma(k+1)} {}_1F_1(k; k+1; -ck), \quad (\text{A.5b})$$

where ${}_1F_1(a; b; z)$ is Kummer's confluent hypergeometric function. From the definitions of ${}_1F_1(a; b; z)$ and of the Pochhammer symbol $(a)_n = \Gamma(a+n)/\Gamma(a)$, we can rewrite (A.5b) as

$$\begin{aligned} \tilde{\gamma}(k, ck) &= \frac{(ck)^k}{\Gamma(k+1)} \sum_{n=0}^{\infty} \frac{(k)_n (-ck)^n}{(k+1)_n n!} \\ &\stackrel{(a)}{=} \sum_{n=0}^{\infty} \frac{(-1)^n (ck)^{k+n}}{\Gamma(k) (k+n) n!}, \end{aligned} \quad (\text{A.6})$$

where line (a) follows after substituting in the definition of the Pochhammer symbols, using the property $\Gamma(x+1) = x\Gamma(x)$, and cancelling alike terms in the numerator and denominator.

We may then calculate the partial derivative of $\tilde{\gamma}(k, ck)$ with respect to k as

$$\begin{aligned} \frac{\partial}{\partial k} \tilde{\gamma}(k, ck) &= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \left[\frac{(ck)^{k+n}}{\Gamma(k)(k+n)} \left(\frac{k+n}{k} + \ln(ck) \right) \right. \\ &\quad \left. - \frac{(ck)^{k+n}}{(k+n)} [\Gamma(k)]^{-2} [\Gamma(k)\psi(k)] - \frac{(ck)^{k+n}}{\Gamma(k)(k+n)^2} \right], \end{aligned} \quad (\text{A.7})$$

where $\psi(k) \triangleq \frac{\partial}{\partial k} \ln(\Gamma(k)) = [\frac{\partial}{\partial k} \Gamma(k)]/\Gamma(k)$ is the digamma or psi function. Rearranging terms, we get

$$\begin{aligned} \frac{\partial}{\partial k} \tilde{\gamma}(k, ck) &= \frac{(ck)^k}{k \Gamma(k)} \sum_{n=0}^{\infty} \frac{(-ck)^n}{n!} + [\ln(ck) - \psi(k)] \sum_{n=0}^{\infty} \frac{(-1)^n (ck)^{k+n}}{\Gamma(k)(k+n)n!} \\ &\quad - \frac{(ck)^k}{\Gamma(k)} \sum_{n=0}^{\infty} \frac{(-ck)^n}{(k+n)^2 n!}. \end{aligned} \quad (\text{A.8})$$

The sum in the first line of (A.8) is equivalent to e^{-ck} . Comparing the sum in the second line of (A.8) with the second line of (A.6) shows this is simply $\tilde{\gamma}(k, ck)$. For the sum in the third line of (A.8), we multiply the numerator and denominator by $[\Gamma(k+n)\Gamma(k+1)\Gamma(k)]^2$. Then, pairs of Gamma functions in the numerator and denominator can be combined into Pochhammer symbols to yield the following sum:

$$\frac{1}{k^2} \sum_{n=0}^{\infty} \frac{(k)_n (k)_n (-ck)^n}{(k+1)_n (k+1)_n n!}, \quad (\text{A.9})$$

We now note the definition of the generalized hypergeometric function ${}_pF_q$ [130, Eq. 9.14.1]:

$${}_pF_q(a_1, a_2, \dots, a_p; b_1, b_2, \dots, b_q; z) = \sum_{n=0}^{\infty} \frac{(a_1)_n (a_2)_n \dots (a_p)_n (z)^n}{(b_1)_n (b_2)_n \dots (b_q)_n n!}. \quad (\text{A.10})$$

(${}_1F_1$ and ${}_2F_1$ are special cases of ${}_pF_q$.) Comparison of (A.9) and (A.10) shows that the sum in (A.9) can be expressed as ${}_2F_2(k, k; k+1, k+1, -ck)$. Making the substitutions described above into (A.8), we finally arrive at:

$$\frac{\partial}{\partial k} \tilde{\gamma}(k, ck) = [\ln(ck) - \psi(k)] \tilde{\gamma}(k, ck) + \frac{(ck)^k}{\Gamma(k+1)} \left[e^{-ck} - \frac{1}{k} {}_2F_2(k, k; k+1, k+1, -ck) \right]. \quad (\text{A.11})$$

Fig. A.2 plots $\frac{\partial}{\partial k} \tilde{\Gamma}(k, ck)$ (which equals $-\frac{\partial}{\partial k} \tilde{\gamma}(k, ck)$) vs. c and k . As seen, for $0 < c \leq 1$, the surface is always above zero and the distance of the surface from the zero plane decreases

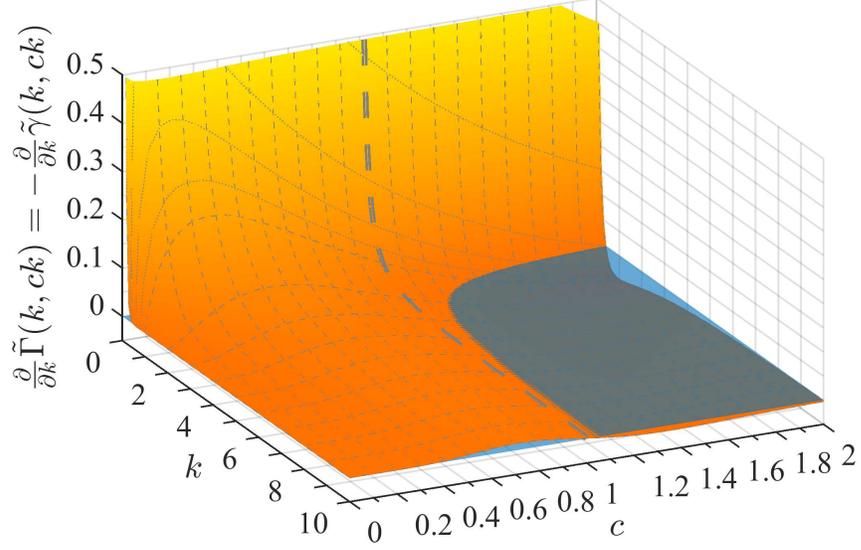


Figure A.2: $\frac{\partial}{\partial k} \tilde{\Gamma}(k, ck)$ (or $-\frac{\partial}{\partial k} \tilde{\gamma}(k, ck)$) vs. c and k . Blue plane indicates where z-axis equals zero; thick dashed line indicates where $c = 1$.

with increasing k . Note that (A.11) applies for any positive real value of k , not just integer values. However, it is rather difficult to formally prove the conjecture of this Lemma analytically instead of pictorially, i.e., that $\frac{\partial}{\partial k} \tilde{\gamma}(k, ck) \leq 0$ if $0 < c \leq 1$. We therefore instead analytically prove the case of $k \in \mathbb{Z}^+$, which suffices for the work in the this paper.

Let us first define the following two functions:

$$f(c, k) = \sum_{n=0}^{k-1} \frac{(ck)^n}{n!}; \quad (\text{A.12a})$$

$$g(c, k) = f(c, k+1) = \sum_{n=0}^k \frac{[c(k+1)]^n}{n!}. \quad (\text{A.12b})$$

By inspection, for $k > 0$, f and g are polynomials in c , both with a positive coefficient on all of their c^n terms. Therefore, f and g are both positive and increasing functions of c . Furthermore, for any positive value of c , both f and g must be increasing functions of k , since not only does each individual term in the sum increase with k (other than $n = 0$, which is constant), an additional positive term is added to the sum when k is incremented. As such, it must be true that $g - f > 0$, or alternatively, $g/f > 1$ or $f/g < 1$.

We next consider the ratio f/g in terms of c . We make use of the following theorem:

Theorem 6 Let $A_n(x) = \sum_{i=0}^n a_i x^i$ and $B_n(x) = \sum_{i=0}^n b_i x^i$ be polynomials of order n such that $b_i > 0$ for $0 \leq i \leq n$. If a_i/b_i is increasing (decreasing) for increasing i , $0 \leq i \leq n$, then the function given by $A_n(x)/B_n(x)$ is increasing (decreasing) with increasing x for all $x > 0$. Moreover, if a_i/b_i is strictly increasing/decreasing with increasing i , then $A_n(x)/B_n(x)$ will also be strictly increasing/decreasing with increasing x .

Proof. See [296] and references therein. ■

The n th coefficient of f as a polynomial in c is $f_n = k^n/n!$, $0 \leq n \leq k-1$, whereas the n th coefficient of g is $g_n = (k+1)^n/n!$, $0 \leq n \leq k$. Thus, $\frac{f_n}{g_n} = \left(\frac{k}{k+1}\right)^n$, which is strictly decreasing from $n=0$ up to $n=k-1$. At $n=k$, $f_k = 0$, so $\frac{f_k}{g_k} = 0$, which is still a strict decrease from $n=k-1$. Thus, by Thm. 6, f/g must be a strictly decreasing function of c for $c > 0$, or alternatively, g/f must be strictly increasing with increasing c .

We now proceed to consider $\tilde{\Gamma}(k, ck) = e^{-ck} f(c, k)$ and $\tilde{\Gamma}(k+1, c(k+1)) = e^{-c(k+1)} g(c, k)$. If $\tilde{\Gamma}(k, ck)$ is increasing with k , then $\tilde{\Gamma}(k+1, c(k+1)) - \tilde{\Gamma}(k, ck) \geq 0$ for all values of k , or $\tilde{\Gamma}(k+1, c(k+1))/\tilde{\Gamma}(k, ck) \geq 1$. This then means $e^{-c} g/f \geq 1$, or $g/f \geq e^c$. To test whether this holds, we first consider the limiting case as $c \rightarrow 0$. In this case, the polynomials are dominated by their lowest-order terms, but we must also account for the fact that g is one order higher than f . We can therefore approximate $g/f \approx \frac{1+(k+1)c+(k+1)^2c^2/2}{1+kc}$, except for $k=1$, where g/f is exactly $1+2c$. Meanwhile, from its Taylor series expansion, e^c is closely approximated by $1+c+\frac{c^2}{2}$. For the $k=1$ case, $g/f - e^c \approx (1+2c) - (1+c+\frac{c^2}{2}) = c(1-\frac{c}{2}) > 0$. For larger k , $\frac{1+(k+1)c+(k+1)^2c^2/2}{1+kc} - (1+c+\frac{c^2}{2}) = \frac{k^2c^2(1-c)}{2(1+kc)} > 0$. Therefore, since $g/f > e^c$ for all $k > 0$, $\tilde{\Gamma}(k, ck)$ must increase with k for very small values of c .

At the other end of the scale, as $c \rightarrow \infty$, the polynomials will be dominated by their highest-order terms. Hence, $g/f \approx \frac{(k+1)^k c^k/k!}{[kc]^{k-1}/(k-1)!} = \left[\frac{k+1}{k}\right]^k c = \left[1+\frac{1}{k}\right]^k c$, which grows linearly with c . However, e^c grows exponentially with c , so regardless of the value of k , there must eventually be some large enough value of c such that $g/f \geq e^c$ can no longer be satisfied. There must therefore exist some maximum value of c for which $g/f \geq e^c$ can be satisfied for all values of k .

Let us now consider the case of $c = 1$, and switch our focus back to $g - f$. $g(1, k)$ can be rewritten as

$$\begin{aligned} g(1, k) &= \sum_{m=0}^k \frac{(k+1)^m}{m!} = \sum_{m=0}^k \sum_{n=0}^m \binom{m}{n} \frac{k^n}{m!} \\ &= \sum_{m=0}^k \sum_{n=0}^m \frac{k^n}{n!(m-n)!} = \sum_{n=0}^k \left[\sum_{m=n}^k \frac{1}{(m-n)!} \right] \frac{k^n}{n!}. \end{aligned} \quad (\text{A.13})$$

As such, $g(1, k) - f(1, k)$ is

$$\begin{aligned} (g - f)|_{c=1} &= \sum_{n=0}^k \left[\sum_{m=n}^k \frac{1}{(m-n)!} \right] \frac{k^n}{n!} - \sum_{n=0}^{k-1} \frac{k^n}{n!} \\ &= \frac{k^k}{k!} + \sum_{n=0}^{k-1} \left[\sum_{m=n}^k \frac{1}{(m-n)!} - 1 \right] \frac{k^n}{n!} \\ &= \frac{k^k}{k!} + \sum_{n=0}^{k-1} \left[\sum_{m=n+1}^k \frac{1}{(m-n)!} \right] \frac{k^n}{n!} \end{aligned} \quad (\text{A.14})$$

We now compare the inner sums in the last part of (A.13) and (A.14). Regardless of the value of k , the inner sum in (A.14) has one fewer term than the inner sum in (A.13). Thus, the value of the inner sum in (A.14) must be less than that of the inner sum in (A.13). As such, if k is incremented, $g - f$ must grow less than g when $c = 1$, or in other words, $g - f$ grows slower with increasing k than g for all values of k when $c = 1$. If for $c = 1$, $g - f$ grows slower than g , then $\frac{g-f}{g} = 1 - \frac{f}{g}$ must decrease with increasing k , $\frac{f}{g}$ must increase with increasing k , and $\frac{g}{f}$ must decrease with increasing k .

This being the case, if we want to test whether $g/f \geq e^c$, it suffices to test what happens in the limit as $k \rightarrow \infty$. Referring back to (A.12), we note that $f(c, k)$ will converge to e^{ck} for large k , while $g(c, k)$ will converge to $e^{c(k+1)}$. Hence, g/f will converge to e^c as $k \rightarrow \infty$, which satisfies the test, and for $c = 1$, g/f will be larger for smaller values of k .

To conclude, we have proved that $g/f \geq e^c, \forall k \in \mathbb{Z}^+$ in the limit as c approaches 0 and for $c = 1$, and by the monotonically increasing nature of g/f with c , this property therefore holds for all values of c in between. Because of this property, it thereby analytically proves $\tilde{\Gamma}(c, k)$ increases (and hence $\tilde{\gamma}(c, k)$ decreases) with increasing $k \in \mathbb{Z}^+$ for $0 < c \leq 1$, while Fig. A.2 this same increasing/decreasing nature is true more generally for $k \in \mathbb{R}^+$. ■

Lemma 9

$$\tilde{\gamma}(k, x) \ll 1, \text{ for } x \leq \frac{k}{10}. \quad (\text{A.15a})$$

$$\tilde{\Gamma}(k, x) \ll 1, \text{ for } x \geq 3k. \quad (\text{A.15b})$$

Proof. We wish to find a value of c where $\tilde{\gamma}(k, ck) \ll 1, \forall k$. From Lemma 8, it suffices to find the value of c satisfying this for $k=1$; larger values of k will meet the condition even better. As a general rule of thumb, $a \ll b$ means that a should be at least an order of magnitude less than b . We therefore want to find c such that $\tilde{\gamma}(1, c) < 0.1$. The integral for $\tilde{\gamma}(1, c)$ trivially reduces to $1 - e^{-c}$; solving for $1 - e^{-c} < 0.1$ for c gives $c < -\ln(0.9) \simeq 0.10536$. Since this value of c is within $(0, 1]$, it will also work for larger k ; for example, $\tilde{\gamma}(2, 2c) \simeq 0.0193$. For simplicity, we round down c to 0.1 (or $1/10$), so that $x = ck \leq k/10$ will make $\tilde{\gamma}(k, x) \ll 1$, thereby completing the proof for (A.15a).

By similar arguments, since $\tilde{\Gamma}(k, x) = 1 - \tilde{\gamma}(k, x)$, to have $\tilde{\Gamma}(k, x) \ll 0.1$, we need $1 - (1 - e^{-c}) < 0.1$, or $e^c > 10$, or $c > \ln(10) \simeq 2.3$. Again for simplicity, we round c up to 3. This completes the proof for (A.15b). ■

Next, if for $n_1 \leq k \leq n_2$ all the magnitudes $|a_k| \ll 1$, then

$$\prod_{k=n_1}^{n_2} (1 - a_k) \simeq 1 - \sum_{k=n_1}^{n_2} a_k, \quad (\text{A.16})$$

which is obtained by expanding out the product and using only the first-order terms from the result.

Since the product series in (3.18) consists of infinite terms, for finite x there exists some smallest integer n that satisfies $n \geq 10x$. Thus, for all $k \geq n$, (A.15a) will hold and so we

can apply (A.16). Splitting the series in (3.18) into two parts:

$$\begin{aligned}
1 - \prod_{k=1}^{\infty} (1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)) &= 1 - \prod_{k=1}^{n-1} [1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)] \prod_{k=n}^{\infty} [1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)] \\
&\simeq 1 - \prod_{k=1}^{n-1} [1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)] \left[1 - \beta \sum_{k=n}^{\infty} \tilde{\gamma}(k, c_{\beta} r^2) \right] \\
&\stackrel{(a)}{=} 1 - \prod_{k=1}^{n-1} [1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)] \left[1 - \pi \lambda'_S r^2 + \beta \sum_{k=1}^{n-1} \tilde{\gamma}(k, c_{\beta} r^2) \right],
\end{aligned} \tag{A.17}$$

where $n = \lceil 10c_{\beta} r^2 \rceil$, and line (a) follows from Lemma 1. Note that in the above, we actually need $\beta \tilde{\gamma}(k, c_{\beta} r^2) \ll 1$ rather than just $\tilde{\gamma}(k, c_{\beta} r^2) \ll 1$. However, since $0 < \beta \leq 1$, choosing n as above will more than satisfy the condition.

We can similarly split off the first m terms from the product in (A.17), and rearrange them as follows:

$$\begin{aligned}
\prod_{k=1}^m [1 - \beta \tilde{\gamma}(k, c_{\beta} r^2)] &= \prod_{k=1}^m \left(1 - \beta \left[1 - \tilde{\Gamma}(k, c_{\beta} r^2) \right] \right) \\
&= \prod_{k=1}^m \left([1 - \beta] \left[1 - \frac{\beta}{\beta-1} \tilde{\Gamma}(k, c_{\beta} r^2) \right] \right) \\
&= (1 - \beta)^m \prod_{k=1}^m \left[1 - \frac{\beta}{\beta-1} \tilde{\Gamma}(k, c_{\beta} r^2) \right].
\end{aligned} \tag{A.18}$$

If $\frac{\beta}{\beta-1} \tilde{\Gamma}(k, c_{\beta} r^2) \ll 1$, we can replace the product with a sum much the same as before. However, β may be close to or even equal 1, so that $\frac{\beta}{\beta-1}$ may be extremely large in magnitude. Following from the proof of Lemma 9, we want $\left| \frac{\beta}{\beta-1} \tilde{\Gamma}(1, c) \right| < 0.1$, or $e^c > \frac{10\beta}{1-\beta}$. Thus, if we specify $m = \left\lfloor \frac{c_{\beta}}{\ln(10\beta/(1-\beta))} r^2 \right\rfloor$, we will satisfy the conditions for (A.16) for $k \leq m$. The last line of (A.18) can then be replaced by

$$\simeq (1 - \beta)^m \left[1 + \frac{\beta}{1-\beta} \sum_{k=1}^m \tilde{\Gamma}(k, c_{\beta} r^2) \right]. \tag{A.19}$$

For sufficiently small r and/or sufficiently large β , we will end up $m = 0$. We remark that the above approximation still holds in the event that $m = 0$, since by convention $\prod_{k=1}^0 (\cdot) = 1$; (A.19) also equals 1 for $m = 0$, even in the limit as $\beta \rightarrow 1$.

By replacing the first m terms of the product series in line (a) of (A.17) with the approximation of (A.19), we obtain Thm. 3.

A.3 Proof of Corollary 1 from Chapter 3

Define the three parts of the product in (3.23) as $\Xi_1(r)$, $\Xi_2(r)$, and $\Xi_3(r)$ as denoted in (3.24). Following the chain rule for differentiation, the derivative of (3.23) is $f_{\tilde{R}_{oS}}(r) = -\frac{\partial \Xi_1(r)}{\partial r} \Xi_2(r) \Xi_3(r) - \Xi_1(r) \frac{\partial \Xi_2(r)}{\partial r} \Xi_3(r) - \Xi_1(r) \Xi_2(r) \frac{\partial \Xi_3(r)}{\partial r}$. Note that although m and n are technically functions of r , due to the ceiling and floor functions, they can be treated as constants for differentiation. For compactness, we denote $x = c_\beta r^2$ in the following.

Tackling the derivatives of the three parts separately:

$$\begin{aligned} \frac{\partial \Xi_1(r)}{\partial r} &= \beta(1 - \beta)^{m-1} \left(\sum_{k=1}^m \frac{-e^{-x} x^{k-1}}{\Gamma(k)} \right) (2c_\beta r) \\ &= -2\pi \lambda'_S r (1 - \beta)^{m-1} e^{-x} \sum_{a=0}^{m-1} \frac{x^a}{\Gamma(a+1)} \\ &\stackrel{(a)}{=} -2\pi \lambda'_S r (1 - \beta)^{m-1} \tilde{\Gamma}(m, x), \end{aligned} \quad (\text{A.20})$$

where line (a) follows from the series expression for $\tilde{\Gamma}(m, x)$ (see Footnote 2 on pg. 49).

Similarly,

$$\begin{aligned} \frac{\partial \Xi_3(r)}{\partial r} &= -2\pi \lambda'_S r + \beta \left(\sum_{k=1}^{n-1} \frac{e^{-x} x^{k-1}}{\Gamma(k)} \right) (2c_\beta r) \\ &= -2\pi \lambda'_S r \left(1 - e^{-x} \sum_{a=0}^{n-2} \frac{x^a}{\Gamma(a+1)} \right) \\ &= -2\pi \lambda'_S r \tilde{\gamma}(n-1, x). \end{aligned} \quad (\text{A.21})$$

Finally, for $\Xi_2(r)$, again by the chain rule,

$$\frac{\partial \Xi_2(r)}{\partial r} = \sum_{k=m+1}^{n-1} \left[\left(\frac{\partial}{\partial r} [1 - \beta \tilde{\gamma}(k, x)] \right) \times \prod_{\substack{\ell=m+1, \\ \ell \neq k}}^{n-1} [1 - \beta \tilde{\gamma}(\ell, x)] \right] \quad (\text{A.22})$$

The product above can be rewritten as $[\prod_{\ell=m+1}^{n-1} (\cdot)] \div [1 - \beta \tilde{\gamma}(k, x)]$, which allows the numerator to be pulled out of the sum, since it no longer depends on k . Also note that the

numerator is simply $\Xi_2(r)$ itself. Continuing the differentiation:

$$\begin{aligned} \frac{\partial \Xi_2(r)}{\partial r} &= \Xi_2(r) \sum_{k=m+1}^{n-1} \left[\frac{\frac{\partial}{\partial r} [1 - \beta \tilde{\gamma}(k, x)]}{1 - \beta \tilde{\gamma}(k, x)} \right] = \Xi_2(r) \sum_{k=m+1}^{n-1} \left[\frac{-\beta e^{-x} x^{k-1}}{\Gamma(k)(1 - \beta \tilde{\gamma}(k, x))} (2c_{\beta} r) \right] \\ &= -2\pi \lambda'_S r \Xi_2(r) e^{-x} \sum_{k=m+1}^{n-1} \frac{x^{k-1}}{\Gamma(k)[1 - \beta \tilde{\gamma}(k, x)]}. \end{aligned} \quad (\text{A.23})$$

Substituting the last lines of (A.20), (A.21), and (A.23) into $f_{\tilde{R}_{oS}}(r)$, we find $2\pi \lambda'_S r \Xi_2(r)$ is a common multiplier in all the terms. Pulling this multiplier out in front gives the expression found in (3.24).

A.4 Proof of Lemma 2 from Chapter 3

Slivnyak's theorem states that for a PPP, adding or removing a point from the process does not change its distributions [121]; thus, the distributions of $\Phi_1 \setminus k$ are the same as for Φ_1 . Furthermore, Campbell's theorem states that for a stationary point process, $\mathbb{E}_{\Phi} \{S\} = \mathbb{E}_{\Phi} \{ \sum_{x \in \Phi} f(x) \} = \lambda \int_{\mathbb{R}^2} f(x) dx$ [121]. In the specific case of a homogeneous PPP, it also results that $\mathbb{V}(S) = \lambda \int_{\mathbb{R}^2} f^2(x) dx$ [121, Corollary 4.8].

If the typical user associates with an MBS at distance r_1 , this means that any interference from the MBS tier must be from points with distance $\|d_{o,x}\| > r_1$. On the other hand, if the typical user associates with an SBS at distance r_2 , then based on the association policy of (3.2), this means that all MBSs must be at least a distance of $r_{\neq 1} = \left(\frac{(N_M - U + 1)P_1}{UBP_2} \right)^{1/\alpha_1} r_2^{\alpha_2/\alpha_1} = \zeta^{-1/\alpha_2} r_2^{\alpha_2/\alpha_1}$ from the user. This makes the function $f(x)$ for calculating $I_{1,1}$ ($I_{2,1}$) equal to $\frac{P_1}{U} h_{o,x} \|d_{o,x}\|^{-\alpha_1}$ for $\|d_{o,x}\| > r_1$ ($r_{\neq 1}$), and 0 otherwise.

Applying Slivnyak's and Campbell's theorems:

$$\begin{aligned} \mathbb{E}\{I_{1,1}\} &= \mathbb{E}_{\Phi_1, h_{o,x}} \left\{ \sum_{x \in \Phi_1} f(x) \right\} = \mathbb{E}_{h_{o,x}} \left\{ \lambda_M \int_{\mathbb{R}^2} f(x) dx \right\} \\ &= \frac{\lambda_M P_1}{U} \mathbb{E}_{h_{o,x}} \{h_{o,x}\} \int_0^{2\pi} \int_{r_1}^{\infty} r^{-\alpha_1} r dr d\theta \\ &\stackrel{(a)}{=} \frac{2\pi \lambda_M P_1}{\alpha_1 - 2} r_1^{2-\alpha_1}, \end{aligned} \quad (\text{A.24})$$

for $\alpha_1 > 2$ (which we have already assumed in Section 3.2.1). Line (a) substitutes $\mathbb{E}_{h_{o,x}}\{h_{o,x}\} = U$ for an interfering MBS, since $h_{o,x} \sim \Gamma(U, 1)$. Similarly:

$$\begin{aligned} \mathbb{V}(I_{1,1}) &= \mathbb{E}_{h_{o,x}} \left\{ \lambda_M \int_{\mathbb{R}^2} f^2(x) dx \right\} \\ &= \frac{\lambda_M P_1^2}{U^2} \mathbb{E}_{h_{o,x}} \{h_{o,x}^2\} \int_0^{2\pi} \int_{r_1}^{\infty} r^{-2\alpha_1} r dr d\theta \\ &\stackrel{(b)}{=} \frac{\pi \lambda_M P_1^2 (U+1)}{U(\alpha_1-1)} r_1^{2-2\alpha_1}, \end{aligned} \quad (\text{A.25})$$

where line (b) substitutes $\mathbb{E}_{h_{o,x}}\{h_{o,x}^2\} = U^2 + U$ for an interfering MBS. The calculation of $\mathbb{E}\{I_{2,1}\}$ and $\mathbb{V}(I_{1,1})$ is identical except replacing r_1 with $r_{\neq 1}$.

A.5 Proof of Lemma 4 from Chapter 3

Campbell's theorem, in the form used in Appendix A.4, applies to stationary point processes in general, not just a PPP. A consequence of this is that the mean interference when measured over all the points of the process will be the same regardless of the specific stationary process. Hence, $\mathbb{E}\{I_{1,2}\}$ for a β -GPP is the same as for a PPP, i.e., the expression given by (3.45), except we do not substitute $\lambda'_S = 2\lambda_M$ this time. Similarly, we can also use Campbell's theorem to find $\mathbb{E}\{I_{1,2}^2\}$:

$$\begin{aligned} \mathbb{E}\{I_{1,2}^2\} &= \mathbb{E}_{\Phi_2, h_{o,x}} \left\{ \left(\sum_{x \in \Phi_2} f(x) \right)^2 \right\} \\ &\stackrel{(a)}{=} \mathbb{E}_{\Phi_2, h_{o,x}} \left\{ \sum_{x \in \Phi_2} P_2^2 h_{o,x}^2 \|d_{o,x}\|^{-2\alpha_2} \right\} \\ &\stackrel{(b)}{+} \mathbb{E}_{\Phi_2, h_{o,x}, h_{o,y}} \left\{ \sum_{\substack{x, y \in \Phi_2, \\ x \neq y}} P_2^2 h_{o,x} h_{o,y} \|d_{o,x}\|^{-\alpha_2} \|d_{o,y}\|^{-\alpha_2} \right\}. \end{aligned} \quad (\text{A.26})$$

The expectation in line (a) of (A.26) is the same expression seen before when calculating interference variances; it evaluates to $\frac{2\pi \lambda'_S P_2^2}{\alpha_2 - 1} r_{\neq 2}^{2-2\alpha_2}$. (Here $U = 1$, so $\mathbb{E}\{h_{o,x}^2\} = U^2 + U = 2$.) However, the expectation in line (b) now poses more of a challenge due to the $x \neq y$ part of the sum. Unlike a PPP, removing one point from a β -GPP (or from most point

processes, generally) does change its statistics, so Slivnyak's theorem no longer applies as it did in Appendix A.4. Instead, we make use of the second moment density $\varrho^{(2)}(x, y)$ and the second factorial moment measure $\alpha^{(2)}(A, B)$, where A and B are regions [121]. In differential form, the two are related by $\alpha^{(2)}(dx, dy) = \varrho^{(2)}(x, y) dx dy$. From Campbell's theorem, it can be found that [121]:

$$\begin{aligned} \mathbb{E}_{\Phi} \left\{ \sum_{\substack{x, y \in \Phi, \\ x \neq y}} f(x, y) \right\} &= \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} f(x, y) \alpha^{(2)}(dx, dy) \\ &= \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} f(x, y) \varrho^{(2)}(x, y) dx dy \end{aligned} \quad (\text{A.27})$$

$\varrho^{(2)}(u)$ for a β -GPP is given by [107, Eq. 6]:

$$\varrho^{(2)}(u) = \lambda_S^2 [1 - \exp(-c_\beta u^2)], \quad (\text{A.28})$$

where $u = |x - y|$ is the distance between points. If we express the points x and y in polar coordinates as $r_x e^{j\theta_x}$ and $r_y e^{j\theta_y}$, respectively, then (A.28) can be rewritten as

$$\lambda_S^2 \left[1 - \exp(-c_\beta [r_x^2 + r_y^2 - 2r_x r_y \cos(\theta_x - \theta_y)]) \right]. \quad (\text{A.29})$$

Returning to line (b) of (A.26), taking the expectation over $h_{o,x}$ and $h_{o,y}$, and noting $\|d_{o,x}\| = r_x$ and $\|d_{o,y}\| = r_y$, we are left with $P_2^2 \mathbb{E}_{\Phi_2} \left\{ \sum_{x, y \in \Phi_2, x \neq y} r_x^{-\alpha_2} r_y^{-\alpha_2} \right\}$. Just as in the PPP case, if the user associates with the MBS tier, all SBSs must be a distance of at least $r_{\neq 2} = \left(\frac{UBP_2}{(N_M - U + 1)P_1} \right)^{1/\alpha_2} r_1^{\alpha_1/\alpha_2} = \zeta^{1/\alpha_2} r_1^{\alpha_1/\alpha_2}$ away from the user. Thus, $f(x, y) = 0$ for distances shorter than that. Applying (A.27) and substituting in (A.29), we have that line (b) of (A.26) equals:

$$\begin{aligned} &P_2^2 \lambda_S^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_{\neq 2}}^{\infty} \int_{r_{\neq 2}}^{\infty} r_x^{-\alpha_2} r_y^{-\alpha_2} \left[1 - \exp(-c_\beta [r_x^2 + r_y^2 - 2r_x r_y \cos(\theta_x - \theta_y)]) \right] \\ &\quad \times r_x dr_x r_y dr_y d\theta_x d\theta_y \\ &= 4\pi^2 P_2^2 \lambda_S^2 \int_{r_{\neq 2}}^{\infty} \int_{r_{\neq 2}}^{\infty} r_x^{1-\alpha_2} r_y^{1-\alpha_2} dr_x dr_y - \varsigma_{1,2} \\ &= 4\pi^2 P_2^2 \lambda_S^2 \frac{r_{\neq 2}^{4-2\alpha_2}}{(\alpha_2 - 2)^2} - \varsigma_{1,2}, \end{aligned} \quad (\text{A.30})$$

where

$$\varsigma_{1,2} = P_2^2 \lambda_S'^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_{\times 2}}^{\infty} \int_{r_{\times 2}}^{\infty} r_x^{1-\alpha_2} r_y^{1-\alpha_2} \exp\left(-c_\beta [r_x^2 + r_y^2 - 2r_x r_y \cos(\theta_x - \theta_y)]\right) \times dr_x dr_y d\theta_x d\theta_y. \quad (\text{A.31})$$

We note that by inspection, the first term in the last line of (A.30) equals $(\mathbb{E}\{I_{1,2}\})^2$. The second term $\varsigma_{1,2}$ can be evaluated as follows:

$$\begin{aligned} \varsigma_{1,2} &\stackrel{(c)}{=} 4\pi^2 P_2^2 \lambda_S'^2 \int_{r_{\times 2}}^{\infty} \int_{r_{\times 2}}^{\infty} r_x^{1-\alpha_2} r_y^{1-\alpha_2} \exp(-c_\beta [r_x^2 + r_y^2]) I_0(2c_\beta r_x r_y) dr_x dr_y \\ &\stackrel{(d)}{=} 4\pi^2 P_2^2 \lambda_S'^2 \int_{r_{\times 2}}^{\infty} \int_{r_{\times 2}}^{\infty} r_x^{1-\alpha_2} r_y^{1-\alpha_2} \exp(-c_\beta [r_x^2 + r_y^2]) \sum_{k=0}^{\infty} \frac{(c_\beta r_x r_y)^{2k}}{\Gamma^2(k+1)} dr_x dr_y. \end{aligned} \quad (\text{A.32})$$

Line (c) follows from first noting that the starting and ending points for the angular integrations are arbitrary, so long as a range of 2π radians is covered. Thus, the limits for θ_x can be changed to between $-\pi + \theta_y$ and $\pi + \theta_y$. We next use a change of variables $\phi_x = \theta_x - \theta_y$ when integrating over θ_x , then apply [130, Eq. 8.431.3] for the modified Bessel function of first kind $I_\nu(z)$ with $\nu = 0$; note $I_0(z)$ is an even function, so the integral over ϕ_x from $-\pi$ to π is twice the integral from 0 to π . Line (d) follows from the power series expansion of $I_0(z)$ in [130, Eq. 8.447.1], which can be alternatively written as $\lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(z/2)^{2k}}{(k!)^2}$.

Next we apply the monotone convergence theorem (Theorem 2) to interchange the order of the limit and integration. Each term in the sequence $f_1(r_x, r_y), f_2(r_x, r_y), f_3(r_x, r_y), \dots$, where $f_n(r_x, r_y) = \sum_{k=0}^n \frac{(c_\beta r_x r_y)^{2k}}{\Gamma^2(k+1)} \exp(-c_\beta [r_x^2 + r_y^2]) (r_x r_y)^{1-\alpha_2}$, is measurable in $\mathbb{R}_{\geq 0}^2$. The sequence converges pointwise to $f(r_x, r_y) = I_0(2c_\beta r_x r_y) \exp(-c_\beta [r_x^2 + r_y^2]) (r_x r_y)^{1-\alpha_2}$ as $n \rightarrow \infty$. Furthermore, for $r_x, r_y \geq 0$, every term in the summation is positive. Hence, each addition of a term to the summation will increase its value. Thus, the sequence $f_n(r_x, r_y)$ is monotonically increasing, i.e., $0 \leq f_1(r_x, r_y) \leq f_2(r_x, r_y) \leq \dots$. Since $f_n(r_x, r_y)$ converges to function $f(r_x, r_y)$ as $n \rightarrow \infty$, and $0 \leq f_1(r_x, r_y) \leq f_2(r_x, r_y) \leq \dots$, the monotone convergence theorem's conditions are satisfied.

Thus, we can pull the limit outside of the integration, along with the associated sum:

$$\begin{aligned}
\varsigma_{1,2} &= 4\pi^2 P_2^2 \lambda_S^2 \lim_{n \rightarrow \infty} \sum_{k=0}^n \left(\frac{c_\beta^{2k}}{\Gamma^2(k+1)} \int_{r_{\neq 2}} \int_{r_{\neq 2}} (r_x r_y)^{1+2k-\alpha_2} \exp(-c_\beta[r_x^2 + r_y^2]) dr_x dr_y \right) \\
&\stackrel{(e)}{=} 4\pi^2 P_2^2 \lambda_S^2 \sum_{k=0}^{\infty} \frac{c_\beta^{2k}}{\Gamma^2(k+1)} \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_{\neq 2}^2)}{4 c_\beta^{2+2k-\alpha_2}} \\
&= \beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_{\neq 2}^2)}{\Gamma^2(k+1)}, \tag{A.33}
\end{aligned}$$

where line (e) comes from applying [130, Eq. 3.381.9] for each integral. We remark that in the event $\frac{\alpha_2}{2} \in \mathbb{Z}^+$, the first parameter in $\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_{\neq 2}^2)$ will become a negative integer for the first few values of k . However, calculating $\Gamma(-n, x)$ with $n \in \mathbb{N}^+$ does not have the same issue as calculating $\Gamma(-n)$, where singularities cause the value to become ∞ . The integral for $\Gamma(-n, x)$ can still be calculated numerically, or alternatively one of [130, Eq. 8.352.3], [130, Eq. 8.352.5], or [297, Eq. 8.4.15] may be used.

Overall, combining (A.26), (A.30), and (A.33), we have $\mathbb{E}\{I_{1,2}^2\} = \frac{2\pi\lambda_S P_2^2}{\alpha_2-1} r_{\neq 2}^{2-2\alpha_2} + (\mathbb{E}\{I_{1,2}\})^2 - \varsigma_{1,2}$. Since $\mathbb{V}(I_{1,2}) = \mathbb{E}\{I_{1,2}^2\} - (\mathbb{E}\{I_{1,2}\})^2$, we finally obtain $\mathbb{V}(I_{1,2}) = \frac{2\pi\lambda_S P_2^2}{\alpha_2-1} r_{\neq 2}^{2-2\alpha_2} - \varsigma_{1,2}$.

A.6 Proof of Lemma 5 from Chapter 3

To find $\mathbb{E}\{I_{2,2}\}$, we follow a similar methodology as on [121, pg. 173]. Since a β -GPP is motion invariant, the mean interference from the SBS tier without counting the point representing the associated transmitter b^* is found by the reduced Palm expectation, shifting the points so that the transmitter is located at the origin and the user is at the location z . The reduced Palm expectation $\mathbb{E}_o^!$, conditioned on the point located at the origin but not including it, is:

$$\mathbb{E}_o^! \left\{ \sum_{x \in \Phi_2} f(x-z) \right\} = \frac{1}{\lambda_S} \int_{\mathbb{R}^2} f(x-z) \varrho^{(2)}(x) dx, \tag{A.34}$$

where $\varrho^{(2)}(u)$ for a β -GPP is given by (A.28). However, since the associated (closest) SBS transmitter is at a distance r_2 away from the user (i.e., $\|z\| = r_2$), the interference for all

other points $x \in \Phi_2$ must be 0 for distances $\|x - z\| \leq r_2$. So:

$$f(x - z) = \begin{cases} \|x - z\|^{-\alpha_2}, & \text{if } \|x - z\| > r_2, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.35})$$

We next perform a change of variables $y = x - z$ in (A.34) (with $dy = dx$), so that when expressing y in polar coordinates as $r_y e^{j\phi_y}$, the condition $\|x - z\| > r_2$ simplifies to $r_y > r_2$. Thus, at location $z = r_2 e^{j\theta_z}$,

$$\begin{aligned} \mathbb{E}\{I_{2,2}\} &= P_2 \lambda'_S \int_{\mathbb{R}^2} \|y\|^{-\alpha_2} [1 - \exp(-c_\beta \|y + z\|^2)] dy \\ &= P_2 \lambda'_S \int_{-\pi}^{\pi} \int_{r_2}^{\infty} r_y^{-\alpha_2} \left[1 - \exp\left(-c_\beta [r_y^2 + r_2^2 + 2r_y r_2 \cos(\phi_y - \theta_z)]\right)\right] r_y dr_y d\phi_y \\ &= P_2 \lambda'_S \int_{-\pi}^{\pi} \int_{r_2}^{\infty} r_y^{1-\alpha_2} dr_y d\phi_y - \varsigma_{2,2} = 2\pi P_2 \lambda'_S \frac{r_2^{2-\alpha_2}}{\alpha_2 - 2} - \varsigma_{2,2}, \end{aligned} \quad (\text{A.36})$$

where

$$\begin{aligned} \varsigma_{2,2} &= P_2 \lambda'_S \int_{-\pi}^{\pi} \int_{r_2}^{\infty} r_y^{1-\alpha_2} \exp\left(-c_\beta [r_y^2 + r_2^2 + 2r_y r_2 \cos(\phi_y - \theta_z)]\right) dr_y d\phi_y \\ &= 2\pi P_2 \lambda'_S \exp(-c_\beta r_2^2) \int_{r_2}^{\infty} r_y^{1-\alpha_2} \exp(-c_\beta r_y^2) I_0(2c_\beta r_2 r_y) dr_y \\ &= \beta P_2 c_\beta^{\frac{\alpha_2}{2}} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1 + k - \frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k + 1)}. \end{aligned} \quad (\text{A.37})$$

For the variance $\mathbb{V}\{I_{2,2}\}$, we again begin by calculating $\mathbb{E}\{I_{2,2}^2\}$. We split the calculation into two parts (respectively denoted ϖ_1 and ϖ_2) much the same as in (A.26).

$$\begin{aligned} \varpi_1 &= 2P_2^2 \mathbb{E}\left\{ \sum_{x \in \Phi_2 \setminus b^*} \|d_{o,x}\|^{-2\alpha_2} \right\} \\ &= 2P_2^2 \mathbb{E}_o^1 \left\{ \sum_{x \in \Phi_2} \|x - z\|^{-2\alpha_2} \right\} \end{aligned} \quad (\text{A.38})$$

this is calculated the same as in (A.36)–(A.37), with α_2 replaced by $2\alpha_2$ and P_2 replaced by $2P_2^2$. (The 2 in front of P_2^2 is $\mathbb{E}\{h_{o,x}^2\} = 2$.) The result is

$$\varpi_1 = 2\pi P_2^2 \lambda'_S \frac{r_2^{2-2\alpha_2}}{\alpha_2 - 1} - 2\beta P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1 + k - \alpha_2, c_\beta r_2^2)}{\Gamma^2(k + 1)}. \quad (\text{A.39})$$

The second part (ϖ_2) involves a sum over pairs of points x and y , where $x \neq y$ and neither x nor y equals b^* . We again use a reduced Palm expectation by shifting all points so b^* is at the origin and the user is at location z . The expectation is now found by means of the third moment density $\varrho^{(3)}$:

$$\mathbb{E}_o^! \left\{ \sum_{\substack{x, y \in \Phi, \\ x \neq y}} f(x - z, y - z) \right\} = \frac{1}{\lambda_S} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} f(x - z, y - z) \varrho^{(3)}(o, x, y) dx dy \quad (\text{A.40})$$

The third moment density for a β -GPP is [107, Eq. 7]:

$$\begin{aligned} \varrho^{(3)}(a, b, c) &= \varrho^{(3)}(o, w_1, w_2) \\ &= \lambda_S^3 \left[1 - e^{-c_\beta |w_1|^2} - e^{-c_\beta |w_2|^2} - e^{-c_\beta |w_1 - w_2|^2} \left(1 - e^{-c_\beta w_1 w_2^*} - e^{-c_\beta w_2 w_1^*} \right) \right] \\ &= \lambda_S^3 \left[\underbrace{1}_{\{1\}} - \underbrace{e^{-c_\beta |w_1|^2}}_{\{2\}} - \underbrace{e^{-c_\beta |w_2|^2}}_{\{3\}} - \underbrace{e^{-c_\beta |w_1 - w_2|^2}}_{\{4\}} \right. \\ &\quad \left. + \underbrace{e^{-c_\beta |w_1 - w_2|^2} \left(e^{-c_\beta w_1 w_2^*} + e^{-c_\beta w_2 w_1^*} \right)}_{\{5\}} \right], \end{aligned} \quad (\text{A.41})$$

where $w_1 = b - a$, $w_2 = c - a$, and point a is specified as the origin o . We split the equation up into five parts as indicated for later use.

Making the change of variables $u = x - z$ and $v = y - z$ in (A.40), the second part of $\mathbb{E}\{I_{2,2}^2\}$ becomes

$$\begin{aligned} \varpi_2 &= \frac{P_2^2}{\lambda_S^2} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} f(u, v) \varrho^{(3)}(o, u + z, v + z) du dv \\ &= P_2^2 \lambda_S^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{-\alpha_2} r_v^{-\alpha_2} \left\{ 1 - e^{-c_\beta [r_u^2 + r_v^2 + 2r_u r_v \cos(\phi_u - \theta_z)]} \right. \\ &\quad \left. - e^{-c_\beta [r_v^2 + r_2^2 + 2r_v r_2 \cos(\phi_v - \theta_z)]} - e^{-c_\beta [r_u^2 + r_v^2 - 2r_u r_v \cos(\phi_u - \phi_v)]} \right. \\ &\quad \times \left[1 - 2 \cos \left(c_\beta [r_u r_v \sin(\phi_u - \phi_v) + r_u r_2 \sin(\phi_u - \theta_z) - r_v r_2 \sin(\phi_v - \theta_z)] \right) \right] \\ &\quad \left. \times e^{-c_\beta [r_2^2 + r_u r_v \cos(\phi_u - \phi_v) + r_u r_2 \cos(\phi_u - \theta_z) + r_v r_2 \cos(\phi_v - \theta_z)]} \right\} r_u dr_u r_v dr_v d\phi_u d\phi_v \quad (\text{A.42}) \end{aligned}$$

We again note that the upper and lower limits on the angle integrals are arbitrary, provided that they cover a range of 2π radians. To solve this quadruple integration, we divide it into

five parts $\varpi_{2,1}$ to $\varpi_{2,5}$ (corresponding to the five parts in (A.41)) and solve them separately.

To begin,

$$\begin{aligned}\varpi_{2,1} &= P_2^2 \lambda_S^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} dr_u dr_v d\phi_u d\phi_v \\ &= 4\pi^2 P_2^2 \lambda_S^2 \frac{r_2^{4-2\alpha_2}}{(2-\alpha_2)^2}.\end{aligned}\quad (\text{A.43})$$

Next:

$$\begin{aligned}\varpi_{2,2} &= -P_2^2 \lambda_S^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} \exp\left(-c_\beta [r_u^2 + r_2^2 + 2r_u r_2 \cos(\phi_u - \theta_z)]\right) \\ &\quad \times dr_u dr_v d\phi_u d\phi_v \\ &= \frac{2\pi P_2^2 \lambda_S^2 r_2^{2-\alpha_2}}{2-\alpha_2} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} r_u^{1-\alpha_2} \exp\left(-c_\beta [r_u^2 + r_2^2 + 2r_u r_2 \cos(\phi_u - \theta_z)]\right) dr_u d\phi_u.\end{aligned}\quad (\text{A.44})$$

We now note that the remaining double integral is mostly the same as the one appearing in (A.37), just with a different factor in front. Thus, we end up with a similar result:

$$\varpi_{2,2} = \frac{2\beta^2 P_2^2}{2-\alpha_2} c_\beta^{1+\frac{\alpha_2}{2}} r_2^{2-\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)}.\quad (\text{A.45})$$

The third term $\varpi_{2,3}$ is nearly identical, just with r_v and ϕ_v swapping places with r_u and ϕ_u , respectively. Since the limits of integration are the same for u and v , the end result is the same as in (A.45):

$$\begin{aligned}\varpi_{2,3} &= -P_2^2 \lambda_S^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} \exp\left(-c_\beta [r_v^2 + r_2^2 + 2r_v r_2 \cos(\phi_v - \theta_z)]\right) \\ &\quad \times dr_u dr_v d\phi_u d\phi_v \\ &= \frac{2\beta^2 P_2^2}{2-\alpha_2} c_\beta^{1+\frac{\alpha_2}{2}} r_2^{2-\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)}.\end{aligned}\quad (\text{A.46})$$

The fourth term $\varpi_{2,4}$ is almost identical to (A.31); the main difference is the lower limit of integration for the distance variables is now r_2 instead of r_{z2} . Hence, we can use the

result from (A.33), replacing r_{z2} with r_2 :

$$\begin{aligned}\varpi_{2,4} &= -P_2^2 \lambda_S'^2 \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} \exp\left(-c_\beta [r_u^2 + r_v^2 - 2r_u r_v \cos(\phi_u - \phi_v)]\right) \\ &\quad \times dr_u dr_v d\phi_u d\phi_v \\ &= -\beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2\left(1+k - \frac{\alpha_2}{2}, c_\beta r_2^2\right)}{\Gamma^2(k+1)}.\end{aligned}\tag{A.47}$$

Finally, we derive the fifth term $\varpi_{2,5}$ as follows. For later convenience, we set the limits of integration for the angle variables to be from θ_z to $2\pi + \theta_z$.

$$\begin{aligned}\varpi_{2,5} &= 2P_2^2 \lambda_S'^2 \int_{\theta_z}^{2\pi+\theta_z} \int_{\theta_z}^{2\pi+\theta_z} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} e^{-c_\beta [r_u^2 + r_v^2 + r_2^2]} \\ &\quad \times \cos\left(c_\beta [r_u r_v \sin(\phi_u - \phi_v) + r_u r_2 \sin(\phi_u - \theta_z) - r_v r_2 \sin(\phi_v - \theta_z)]\right) \\ &\quad \times e^{-c_\beta [r_u r_2 \cos(\phi_u - \theta_z) + r_v r_2 \cos(\phi_v - \theta_z) - r_u r_v \cos(\phi_u - \phi_v)]} dr_u dr_v d\phi_u d\phi_v.\end{aligned}\tag{A.48}$$

Making the transformation of variables $\theta_u = \phi_u - \theta_z$ and $\theta_v = \phi_v - \theta_z$, and changing the order of the integrals, we obtain

$$\begin{aligned}\varpi_{2,5} &= 2P_2^2 \lambda_S'^2 \left[\int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} e^{-c_\beta [r_u^2 + r_v^2 + r_2^2]} \times \right. \\ &\quad \left. \mathcal{Z} \left\{ \int_0^{2\pi} \int_0^{2\pi} e^{-c_\beta [r_u r_2 \cos \theta_u + r_v r_2 \cos \theta_v - r_u r_v \cos(\theta_u - \theta_v)]} \right. \right. \\ &\quad \left. \left. \times \cos\left(c_\beta [r_u r_v \sin(\theta_u - \theta_v) + r_u r_2 \sin(\theta_u) - r_v r_2 \sin(\theta_v)]\right) d\theta_u d\theta_v dr_u dr_v \right\} \right].\end{aligned}\tag{A.49}$$

We now focus on the inner two integrals, denoted by \mathcal{Z} . Using the properties $\cos(a \pm b) = \cos a \cos b \mp \sin a \sin b$ and $\sin(a \pm b) = \sin a \cos b \pm \cos a \sin b$, we expand the $\cos(\theta_u - \theta_v)$ and $\sin(\theta_u - \theta_v)$ terms to obtain

$$\begin{aligned}\mathcal{Z} &= \int_0^{2\pi} \int_0^{2\pi} e^{-c_\beta [r_u \cos \theta_u (r_2 - r_v \cos \theta_v) - r_u r_v \sin \theta_u \sin \theta_v + r_v r_2 \cos \theta_v]} \\ &\quad \times \cos\left(c_\beta [\sin \theta_u (r_u r_2 + r_u r_v \cos \theta_v) - r_u r_v \cos \theta_u \sin \theta_v - r_v r_2 \sin \theta_v]\right) d\theta_u d\theta_v\end{aligned}$$

(continued next page)

$$\begin{aligned}
&\stackrel{(a)}{=} \int_0^{2\pi} \int_0^{2\pi} e^{\cos \theta_u \underbrace{(c_\beta r_u r_v \cos \theta_v - c_\beta r_u r_2)}_p + \sin \theta_u \underbrace{(c_\beta r_u r_v \sin \theta_v)}_q} e^{-c_\beta r_v r_2 \cos \theta_v} \left(\cos[c_\beta r_v r_2 \sin \theta_v] \right. \\
&\quad \times \cos \left[(\sin \theta_u) \underbrace{(c_\beta r_u r_2 + c_\beta r_u r_v \cos \theta_v)}_g + (\cos \theta_u) (-c_\beta r_u r_v \sin \theta_v) \right] + \sin[c_\beta r_v r_2 \sin \theta_v] \\
&\quad \times \sin \left[(\sin \theta_u) \underbrace{(c_\beta r_u r_2 + c_\beta r_u r_v \cos \theta_v)}_g + (\cos \theta_u) \underbrace{(-c_\beta r_u r_v \sin \theta_v)}_f \right] \Big) d\theta_u d\theta_v \\
&\stackrel{(b)}{=} \int_0^{2\pi} e^{-c_\beta r_v r_2 \cos \theta_v} \cos(c_\beta r_v r_2 \sin \theta_v) \left(\int_0^{2\pi} e^{p \cos \theta_u + q \sin \theta_u} \cos[f \cos \theta_u + g \sin \theta_u] d\theta_u \right) d\theta_v \\
&\stackrel{(c)}{+} \int_0^{2\pi} e^{-c_\beta r_v r_2 \cos \theta_v} \sin(c_\beta r_v r_2 \sin \theta_v) \left(\int_0^{2\pi} e^{p \cos \theta_u + q \sin \theta_u} \sin[f \cos \theta_u + g \sin \theta_u] d\theta_u \right) d\theta_v,
\end{aligned} \tag{A.50}$$

where line (a) comes from applying expanding $\cos(a-b)$ with the expressions in the previous two lines before (a) playing the parts of “a” and “b”, respectively. We also note that the expressions we have denoted p , q , f , and g do not contain θ_u , and so are not involved when integrating over θ_u .

Next, we apply [130, Eq. 3.937.1] to the inner integral of (A.50) line (c) and [130, Eq. 3.937.2] to the inner integral of (A.50) line (b), both with $m=0$. These equations both require $(g-p)^2 + (f+q)^2 > 0$ to be used. In our case, $f+q=0$ and $g-p=2c_\beta r_u r_2$. Since c_β , r_u , and r_2 are all positive real values, the condition is satisfied. With $m=0$, the results of the two integrals simplify considerably. They yield expressions just involving $I_0(\sqrt{C \pm iD})$, where $C = p^2 + q^2 - f^2 - g^2 = -4c_\beta^2 r_u^2 r_v r_2 \cos \theta_v$ and¹ $D = 2(fp + gq) = -4c_\beta^2 r_u^2 r_v r_2 \sin \theta_v$. We further note from [297, Eq. 10.39.9] that $I_0(z) = {}_0F_1(; 1; \frac{z^2}{4})$. Hence,

$$\begin{aligned}
I_0(\sqrt{C \pm iD}) &= I_0\left(\sqrt{-4c_\beta^2 r_u^2 r_v r_2 \cos \theta_v \mp i4c_\beta^2 r_u^2 r_v r_2 \sin \theta_v}\right) \\
&= {}_0F_1\left(; 1; -c_\beta^2 r_u^2 r_v r_2 e^{\mp i\theta_v}\right).
\end{aligned} \tag{A.51}$$

¹Please note that there is a typo in the 8th edition of [130] stating that $D = -2(fp + gq)$. The correct version of D , without the minus sign, is given in the original source, [298, Sec. 337, Eq. 9a] and earlier editions of [130].

Thus, (A.50) becomes

$$\begin{aligned}
\mathcal{Z} &= \int_0^{2\pi} \pi e^{-\overbrace{(c_\beta r_v r_2)^2}_{t_1} \cos \theta_v} \cos(c_\beta r_v r_2 \sin \theta_v) [{}_0F_1(; 1; -c_\beta^2 r_u^2 r_v r_2 e^{i\theta_v}) \\
&\quad + {}_0F_1(; 1; -\overbrace{(c_\beta^2 r_u^2 r_v r_2)^2}_{t_2} e^{-i\theta_v})] d\theta_v \\
&\quad + \int_0^{2\pi} i\pi e^{-c_\beta r_v r_2 \cos \theta_v} \sin(c_\beta r_v r_2 \sin \theta_v) [{}_0F_1(; 1; -c_\beta^2 r_u^2 r_v r_2 e^{i\theta_v}) \\
&\quad - {}_0F_1(; 1; -c_\beta^2 r_u^2 r_v r_2 e^{-i\theta_v})] d\theta_v \\
&= \int_0^{2\pi} \pi e^{-t_1 \cos \theta_v} \left(\cos(t_1 \sin \theta_v) + i \sin(t_1 \sin \theta_v) \right) {}_0F_1(; 1; -t_2 e^{i\theta_v}) \\
&\quad + [\cos(t_1 \sin \theta_v) - i \sin(t_1 \sin \theta_v)] {}_0F_1(; 1; -t_2 e^{-i\theta_v}) d\theta_v \\
&= \int_0^{2\pi} \pi e^{-t_1 \cos \theta_v} \left[{}_0F_1(; 1; -t_2 e^{i\theta_v}) e^{i t_1 \sin \theta_v} + {}_0F_1(; 1; -t_2 e^{-i\theta_v}) e^{-i t_1 \sin \theta_v} \right] d\theta_v \\
&= \int_0^{2\pi} \pi \left[e^{-t_1 \cos \theta_v + i t_1 \sin \theta_v} {}_0F_1(; 1; -t_2 e^{i\theta_v}) + e^{-t_1 \cos \theta_v - i t_1 \sin \theta_v} {}_0F_1(; 1; -t_2 e^{-i\theta_v}) \right] d\theta_v \\
&= \int_0^{2\pi} \pi \left[\exp(-t_1 e^{-i\theta_v}) {}_0F_1(; 1; -t_2 e^{i\theta_v}) + \exp(-t_1 e^{i\theta_v}) {}_0F_1(; 1; -t_2 e^{-i\theta_v}) \right] d\theta_v \\
&\stackrel{(d)}{=} 2\pi \int_0^{2\pi} \exp(-t_1 e^{-i\theta_v}) {}_0F_1(; 1; -t_2 e^{i\theta_v}) d\theta_v \\
&\stackrel{(e)}{=} 2\pi \oint_{\mathcal{C}} \exp(-t_1 z_o^{-1}) {}_0F_1(; 1; -t_2 z_o) \frac{dz_o}{iz_o}. \tag{A.52}
\end{aligned}$$

Line (d) follows from a change of variable $\theta_s = -\theta_v$ in the second half of the line preceding (d), then noting it results in the same integral as the first half, just from -2π to 0 ; it is equivalent to shift those limits to be from 0 to 2π . Line (e) follows from the change of variable $z_c = e^{i\theta_v}$, which transforms the equation into a contour integral, where the contour \mathcal{C} is the unit circle.

e^{z_c} has only one (essential) singular point at $z_c = \infty$. Hence, $e^{-t_1 z_c^{-1}}$, with $t_1 \in \mathbb{R}^+$, has one singular point at $z_c = 0$. Meanwhile, ${}_0F_1(; 1; -t_2 z_c)$ with $t_2 \in \mathbb{R}^+$ has no poles or singularities inside the unit circle. Therefore, by the residue theorem for contour integration,

when integrating over function $f(z_c)$, the value of the integral will be $2\pi i$ times the sum of residues of $f(z_c)$ at $z_c = 0$. Furthermore, if $f(z_c)$ is written as a power series, i.e., $f(z_c) = \sum_{n=-\infty}^{\infty} a_n z_c^n$, the sum of residues in this case will be the coefficient on the z_c^0 term of $z_c \cdot f(z_c)$. So,

$$\begin{aligned}
z_c \cdot f(z_c) &= \frac{2\pi}{i} \exp(-t_1 z_c^{-1}) {}_0F_1(; 1; -t_2 z_c) \\
&= \frac{2\pi}{i} \left(\sum_{k=0}^{\infty} \frac{(-t_1 z_c^{-1})^k}{k!} \right) \left(\sum_{\ell=0}^{\infty} \frac{(-t_2 z_c)^\ell}{(1)_\ell \ell!} \right) = \frac{2\pi}{i} \sum_{\ell=0}^{\infty} \sum_{k=0}^{\infty} \frac{(-1)^{k+\ell} t_1^k t_2^\ell z_c^{\ell-k}}{(1)_\ell \ell! k!} \\
&\stackrel{(f)}{=} \frac{2\pi}{i} \sum_{\ell=0}^{\infty} \sum_{n=-\infty}^{\ell} \frac{(-1)^{\ell-n+\ell} t_1^{\ell-n} t_2^\ell z_c^n}{(1)_\ell \ell! (\ell-n)!} \stackrel{(g)}{=} \frac{2\pi}{i} \sum_{n=-\infty}^{\infty} \left(\sum_{\ell=n}^{\infty} \frac{(-1)^n t_1^{\ell-n} t_2^\ell}{(1)_\ell \ell! (\ell-n)!} \right) z_c^n.
\end{aligned} \tag{A.53}$$

Line (f) follows from the change of variables $n = \ell - k$, and line (g) follows from reversing the order of the summations.

Thus, the coefficient on the z_c^0 term is

$$\frac{2\pi}{i} \sum_{\ell=0}^{\infty} \frac{t_1^\ell t_2^\ell}{(1)_\ell \ell! \ell!} = \frac{2\pi}{i} \sum_{\ell=0}^{\infty} \frac{(t_1 t_2)^\ell}{(1)_\ell (1)_\ell \ell!} = \frac{2\pi}{i} {}_0F_2(; 1, 1; t_1 t_2). \tag{A.54}$$

So, we arrive at

$$\begin{aligned}
\mathcal{Z} &= 2\pi i \times \frac{2\pi}{i} {}_0F_2(; 1, 1; t_1 t_2) = 4\pi^2 \cdot {}_0F_2(; 1, 1; t_1 t_2) \\
&= 4\pi^2 \cdot {}_0F_2(; 1, 1; c_\beta^3 r_u^2 r_v^2 r_2^2)
\end{aligned} \tag{A.55}$$

and

$$\varpi_{2,5} = 8\pi^2 P_2^2 \lambda_S'^2 \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} e^{-c_\beta [r_u^2 + r_v^2 + r_2^2]} {}_0F_2(; 1, 1; c_\beta^3 r_u^2 r_v^2 r_2^2) dr_u dr_v. \tag{A.56}$$

Converting ${}_0F_2$ back to its power series form, we get

$$\varpi_{2,5} = 8\pi^2 P_2^2 \lambda_S'^2 e^{-c_\beta r_2^2} \int_{r_2}^{\infty} \int_{r_2}^{\infty} r_u^{1-\alpha_2} r_v^{1-\alpha_2} e^{-c_\beta [r_u^2 + r_v^2]} \sum_{k=0}^{\infty} \frac{(c_\beta^3 r_2^2)^k (r_u r_v)^{2k}}{\Gamma^3(k+1)} dr_u dr_v, \tag{A.57}$$

which we note is very similar in structure to the double integral in (A.32). As such, it can

be solved in much the same manner. The result is

$$\begin{aligned}
\varpi_{2,5} &= 2\pi^2 P_2^2 \lambda_S'^2 e^{-c_\beta r_2^2} c_\beta^{\alpha_2 - 2} \sum_{k=0}^{\infty} \frac{(c_\beta r_2^2)^k \Gamma^2(1 + k - \frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^3(k + 1)} \\
&= 2\beta^2 P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma^2(1 + k - \frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^3(k + 1)}. \tag{A.58}
\end{aligned}$$

Collecting everything together, we have $\mathbb{V}(I_{2,2}) = \varpi_1 + \varpi_2 - (\mathbb{E}\{I_{2,2}\})^2 = \varpi_1 + \sum_{n=1}^5 \varpi_{2,n} - (\mathbb{E}\{I_{2,2}\})^2$. The last term, found by squaring (A.36)–(A.37), is

$$\begin{aligned}
(\mathbb{E}\{I_{2,2}\})^2 &= 4\pi^2 P_2^2 \lambda_S'^2 \frac{r_2^{4-2\alpha_2}}{(\alpha_2 - 2)^2} + \frac{4\beta^2 P_2^2}{2 - \alpha_2} c_\beta^{1+\frac{\alpha_2}{2}} r_2^{2-\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \\
&\quad + \beta^2 P_2^2 c_\beta^{\alpha_2} e^{-2c_\beta r_2^2} \left(\sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \right)^2 \tag{A.59}
\end{aligned}$$

We note by inspection that the first line of (A.59) equals $\varpi_{2,1}$ in (A.43), and the second line of (A.59) equals $\varpi_{2,2} + \varpi_{2,3}$ from (A.45) and (A.46). Hence, those terms will cancel each other in $\mathbb{V}(I_{2,2})$. We therefore finally obtain

$$\begin{aligned}
\mathbb{V}(I_{2,2}) &= 2\pi P_2^2 \lambda_S' \frac{r_2^{2-2\alpha_2}}{\alpha_2 - 1} - 2\beta P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\alpha_2, c_\beta r_2^2)}{\Gamma^2(k+1)} \\
&\quad - \beta^2 P_2^2 c_\beta^{\alpha_2} \sum_{k=0}^{\infty} \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \\
&\quad + 2\beta^2 P_2^2 c_\beta^{\alpha_2} e^{-c_\beta r_2^2} \sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma^2(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^3(k+1)} \\
&\quad - \beta^2 P_2^2 c_\beta^{\alpha_2} e^{-2c_\beta r_2^2} \left(\sum_{k=0}^{\infty} (c_\beta r_2^2)^k \frac{\Gamma(1+k-\frac{\alpha_2}{2}, c_\beta r_2^2)}{\Gamma^2(k+1)} \right)^2. \tag{A.60}
\end{aligned}$$