Data-driven Transfer Learning Methods for Wireless Networks

Rahul Kumar Jaiswal

UiA University of Agder

Doctoral Dissertations at the University of Agder 475

Data-driven Transfer Learning Methods for Wireless Networks

Rahul Kumar Jaiswal

Data-driven Transfer Learning Methods for Wireless Networks

Doctoral Dissertation for the Degree *Philosophiae Doctor (Ph.D.)*
at the Faculty of Engineering and Science, Specialisation in Artificial Intelligence

University of Agder
Faculty of Engineering and Science
2024

Doctoral Dissertations at the University of Agder 475
ISSN: 1504-9272
ISBN: 978-82-8427-192-7

©Rahul Kumar Jaiswal, 2024

Printed by Make!Graphics
Kristiansand

Preface and Acknowledgments

This dissertation incorporates the research work carried out at the Center Intelligent Signal Processing and Wireless Networks (WISENET) Center, Department of Information and Communication Technology (ICT), University of Agder (UiA), Grimstad, Norway.

I would like to express my sincere gratitude to my advisor Prof. Baltasar Beferull Lozano, Director of the WISENET Center, for giving me the opportunity to work in his research group, and for supervising the research work carried out in this dissertation. He offered an uncountable number of suggestions and ideas during the course of my research work. He is an excellent teacher and a wonderful person!

I express my heartfelt thanks to Dr. Siddharth Deshmukh and Dr. Mohamed Elnourani for their continuous suggestions and discussions, and for uplifting my knowledge during the course of my research work.

I am thankful to all my colleagues at the WISENET Centre for providing an excellent research environment and to the administrative and technical staff who helped me throughout with prompt support. I am also grateful to Emma, Kristine, and Julia for providing me with any type of administrative support immediately.

I am pleased to thank my friends Abhishek, Ayan, Diego, Emilio, Hareesh, Joshin, Juan, Jyotirmoy, Kevin, Mohamed, Pooja, Preeti, Raju, Ravi, Rohan, Sarang, Shibarchi, Sreenivasa, Surendra, and Wilson.

I express my eternal gratitude to my parents and each family member, Anu and each family member for the immeasurable love and trust, without whom this journey could not be accomplished.

Finally, I would like to thank the almighty God for bestowing divine blessings on me. I pray to guide me always and keep me on the path of righteousness.

Rahul Kumar Jaiswal
Grimstad, Norway
February 12, 2024

To Maa and Pita Jee!

Sammendrag

Radiokart gir informasjon om signalstyrke og nettverksdekning i et angitt geografisk område. Beregningen av nøyaktige radiokart er nødvendig for å forbedre ytelsen til mange bruksområder for fremtidige trådløse nettverk, for eksempel lokalisering, nettverksplanlegging og ressursallokering. For å utarbeide nøyaktige radiokart kan den nøyaktige posisjonen til sender (`Tx`) og mottaker (`Rx`) brukes. Dette er kjent som den stedsbaserte metoden. Men i praksis har trådløse nettverk en høy grad av multibane, med resultat av at det er vanskelig å finne nøyaktig posisjon av Rx. Alternativt kan man bruke signalets ankomsttidspunkt (ToA), som er lettere å måle. Dette er kjent som den stedsfrie metoden. En av måtene å inkorporere begge metodene er Mixture of Experts (MoE).

På grunn av endringer i utbredelsesegenskapene til trådløse nettverk, kan en radiokartmodell designet under et bestemt miljø (kildemiljø) ikke brukes direkte i et nytt miljø (målmiljø). Utformingen av en ny modell krever en betydelig mengde måleprøver, og ofte store beregningsressurser og datainnsamlingskostnader.

For å adressere disse problemstillingene foreslår vi i denne avhandlingen en serie av Transfer Learning (TL) modeller som bruker hver av de nevnte metodene for å estimere radiokart i nye trådløse miljøer der det er mangel på måleprøver. For dette formålet trener vi først en radiokartmodell i et kildemiljø og overfører den deretter til et annet lignende, men fortsatt annerledes målmiljø. Deretter finjusteres modellen for målmiljøet ved å bruke en liten mengde prøver av det trådløse målmiljøet.

For et slikt system styrer likheten mellom to trådløse miljøer effektiviteten til TL-operasjonen. Derfor, for å kvantifisere likheten, undersøker vi forskjellige klassiske likhetsmål inkludert den mye brukte Wasserstein-avstanden. Numerisk viser vi at disse klassiske målene ikke gir gode resultater i sammenheng med TL for radiokart-estimering. For å overvinne begrensningene til disse klassiske målene, designer vi et datadrevet likhetsmål (DDS), som er i stand til å fange opp alle variasjonene av trådløse miljøer og kan lære de trådløse forplantningsegenskapene direkte fra dataene. I tillegg kan vår DDS forutsi mengden treningsdata som trengs for å estimere radiokart i nye trådløse målmiljøer når TL-operasjonen utføres.

Eksperimenter viser at våre foreslåtte TL-metoder fungerer effektivt med høy modellnøyaktighet og sparer en betydelig mengde sensormålingsdata. Ulike modeller er designet for hvert av tilfellene med stedsbasert, stedsfri og MoE-basert radiokartvurdering. Numeriske eksperimenter viser ytelsen til alle tilfellene.

Til slutt undersøker vi anvendelsen av TL mellom to forskjellige optimaliseringsproblemer med felles ressursallokering (kanaltilordning og kraftallokering) i

underliggende D2D-kommunikasjon. Ressursallokeringsmodellen som er trent på datasettet hentet fra scenarioet med perfekt kanaltilstandsinformasjon (CSI), overføres til det ufullkomne CSI-scenarioet og finjusteres deretter. Eksperimentet viser at TL forbedrer ytelsen til det ufullkomne CSI-scenariet med mindre mengder treningsdata.

Abstract

Radio maps provide information about spatial signal strength and network coverage in a designated geographical area. The estimation of accurate radio maps is necessary to improve the performance of many applications of future wireless networks. For instance, localization, network planning, and resource allocation, to name a few. To obtain accurate radio maps, the exact knowledge of transmitter (`Tx`) and receiver (`Rx`) locations can be used. This is known as the location-based method. However, in practice, wireless networks incur a high degree of multipath. As a result, it is difficult to obtain accurate locations of `Rxs`. Alternatively, time of arrival (ToA) features of radio signals, which are easier to obtain, can be used. This is known as the location-free method. One of the ways to incorporate both methods is the mixture of experts (MoE).

Due to changes in the propagation characteristics of wireless networks, a radio map model designed under a particular wireless environment (source environment) can not be directly used in a new wireless environment (target environment). Moreover, designing a new radio map model for each new wireless environment requires a huge amount of measurement samples and may need substantial computational resources and data acquisition costs.

To address these issues, in this dissertation, we propose a series of transfer learning (TL) schemes using each of the aforementioned methods to estimate radio maps in new wireless environments where there is a scarcity of measurement samples. To this end, we first train a radio map model in a source wireless environment and then transfer it to another similar but still different target wireless environment. It is then fine-tuned using a small amount of samples of the target wireless environment to reduce the data acquisition cost.

For such a scheme, the similarity between two wireless environments controls the effectiveness of the TL operation. Therefore, to quantify the similarity, we investigate different classical similarity measures including the widely used Wasserstein distance. Numerically, we show that these classical measures do not perform well in the context of TL for radio map estimation. To overcome the limitations of these classical measures, we design a data-driven similarity measure (DDS), which is able to capture all the variations of wireless environments and can learn the wireless propagation characteristics directly from the data. Additionally, our DDS can predict the amount of training data needed to estimate radio maps in new target wireless environments when performing the TL operation.

Experiments show that our proposed TL schemes perform efficiently with high

model accuracy and save a substantial amount of sensor measurement data. Different models are designed for each of the cases of location-based, location-free, and MoE-based radio map estimation. Numerical experiments showcase the performance of each case, respectively.

Finally, we investigate the application of TL between two different optimization problems of joint resource allocation (channel assignment and power allocation) in underlay D2D communication. The resource allocation model trained on the dataset obtained from the perfect channel state information (CSI) scenario is transferred to the imperfect CSI scenario and then fine-tuned. The experiment shows that TL improves the performance of the imperfect CSI scenario with less amount of training data.

Publications

The following papers are included in this dissertation and are appended in Appendices A-E at the end of the dissertation.

- Paper A **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano, "Deep Transfer Learning Based Radio Map Estimation for Indoor Wireless Communications," 23rd IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1-5, 2022. **(link)**

- Paper B **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano, "A Data-driven Transfer Learning Method for Indoor Radio Map Estimation," IEEE Transactions on Wireless Communications (TWC), 2023. **(submitted)**

- Paper C **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano, "Location-free Indoor Radio Map Estimation using Transfer learning," 97th IEEE Vehicular Technology Conference (VTC-Spring), pp. 1-7, 2023. **(link)**

- Paper D **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano, "Leveraging Transfer Learning for Radio Map Estimation via Mixture of Experts," IEEE Transactions on Machine Learning in Communications and Networking (TMLCN), 2024. **(submitted)**

- Paper E **Rahul Jaiswal**, Siddharth Deshmukh, Mohamed Elnourani, Baltasar Beferull-Lozano, "Transfer Learning Based Joint Resource Allocation for Underlay D2D Communications," 20th IEEE Wireless Communications and Networking Conference (WCNC), pp. 1479-1484, 2022. **(link)**

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Summerized State of Art 4
1.3 Problem Statements and Main Contributions 5
1.4 Outline of the Dissertation 8

2 Background **11**
2.1 Wireless Signal Propagation 11
2.1.1 Multipath Propagation 13
2.2 Time of Arrival 14
2.3 Trilateration Algorithm 15
2.4 Deep Neural Network 16
2.4.1 Convolutional Neural Network 17
2.5 Transfer Learning 17
2.6 Similarity Measures for Radio Maps 19
2.7 Mixture of Experts 24
2.8 Transfer Learning for Resource Allocation 25
2.8.1 Similarity Measure for Resource Allocation 26

3 Transfer Learning for Location-based Radio Map Estimation **27**
3.1 Motivation 27
3.2 Problem Formulation 28
3.3 Radio Map Estimation Model Using TL 30
3.4 TL System Design 31
3.5 Transfer Learning Approach 31
3.6 Similarity Measure 32
3.6.1 Classical Similarity Measures 32
3.6.2 Data-driven Similarity Measure 34
3.7 Data Generation 36
3.8 Performance Evaluation 38
3.8.1 Model Accuracy 38
3.8.2 Noise Robustness 40
3.9 Visualization of Radio Maps 41
3.10 Performance of the TL Model with Additional Complex Changes in Wireless Environments 41

3.11 Summary 42

4 Transfer Learning for Location-free Radio Map Estimation 45
4.1 Motivation 45
4.2 TL System Design 45
4.3 Baseline Model and Transfer Learning Approach 45
4.4 Similarity Measure 46
4.5 Data Generation 46
4.6 TL Performance 46
4.7 Visualization of Radio Maps 47
4.8 Summary 48

5 Transfer Learning for Radio Map Estimation using Mixture of Experts 51
5.1 Motivation 51
5.2 Problem Formulation 52
5.3 Radio Map Estimation Model Using TL 53
5.4 Transfer Learning Approach 54
5.5 Model Performance 54
5.6 Visualization of Radio Maps 56
5.7 Performance of the TL Model with Additional Complex Changes in Wireless Environments 57
5.8 Summary 58

6 Transfer Learning Based Joint Resource Allocation 59
6.1 Motivation 59
6.2 TL System Model 60
6.2.1 Resource Allocation under Perfect CSI 61
6.2.2 Resource Allocation under Imperfect CSI 62
6.3 Transfer Learning Approach 62
6.4 Similarity Measure 62
6.5 Data Generation 63
6.6 Numerical results 63
6.7 Summary 65

7 Concluding Remarks 67
7.1 Conclusion 67
7.2 Future Work 68

A PAPER A 71
A.1 Introduction 72
A.2 Description of Environments for Transfer learning 74
A.3 Transfer Learning Based Radio Map Estimation 76
A.3.1 Baseline DNN Model and TL Approach 76
A.3.2 Similarity Measure using Wasserstein Distance 77

A.3.3 Data-driven Similarity Measure 77
A.4 Results and Discussions 78
A.5 Conclusion 81

B PAPER B **83**
B.1 Introduction 84
B.2 Related Works and Motivation 86
B.3 Problem Formulation 89
B.4 Proposed Radio Map Estimation Model 90
B.4.1 Baseline DNN Model and Transfer Learning Approach 92
B.4.2 Similarity Measures 94
B.5 Numerical Evaluation 99
B.5.1 Generation of Indoor Wireless Environments 99
B.5.2 Performance Measures 101
B.5.2.1 Model Accuracy 101
B.5.2.2 Noise Robustness 102
B.5.3 System Setup 103
B.5.4 Results and Discussions 103
B.5.4.1 Baseline DNN Model Analysis 104
B.5.4.2 Experimental Analysis of TL 105
B.5.4.3 Model Accuracy 107
B.5.4.4 Noise Robustness 108
B.5.4.5 Time Complexity 109
B.5.4.6 Illustration of Estimated Radio Maps 110
B.5.4.7 Temporal Window Experiment 110
B.5.4.8 Comparison with State-of-the-art 111
B.5.4.9 TL Performance with Cafe Area Indoor Wireless Environment 113
B.6 Conclusion and Future work 116

C PAPER C **117**
C.1 Introduction 118
C.2 Generation of Wireless Environments 120
C.3 Radio Map Estimation Method 122
C.3.1 Baseline Model and TL Operation 122
C.3.2 Similarity Measure based on the Wasserstein Distance 123
C.3.3 Data-driven Similarity Measure 123
C.3.3.1 Case A 124
C.3.3.2 Case B 124
C.4 Results and Discussions 125
C.4.1 Baseline DNN Model 125
C.4.2 Wasserstein Distance as a Similarity Measure 126
C.4.3 Data-driven Similarity Measure 127
C.4.3.1 Method 1 127

C.4.3.2 Method 2 127
C.4.4 Comparison of Reliability 129
C.4.5 Illustration of Radio Maps 130
C.5 Conclusion 131

D PAPER D **133**
D.1 Introduction 134
D.2 Problem Formulation 137
D.3 Proposed Radio Map Estimation Model 138
D.3.1 Baseline MoE Model and Transfer Learning Approach 139
D.3.2 Similarity Measure 140
D.4 Generation of Indoor Wireless Environments 143
D.5 Results and Discussions 146
D.5.1 Numerical Analysis with Noiseless ToAs 146
D.5.2 Numerical Analysis with Noisy ToAs 149
D.5.3 Numerical Analysis with Noisy ToAs and Additional Complex Changes in Wireless Environments 150
D.5.4 Performance Comparison 151
D.5.5 Choice of Thresholds 151
D.5.6 Estimated Radio Maps 152
D.6 Conclusions 152

E PAPER E **155**
E.1 Introduction 156
E.2 System Model 158
E.2.1 Task I: Resource Allocation under Perfect CSI 159
E.2.2 Task II: Resource Allocation under Imperfect CSI 160
E.3 Transfer Learning based Resource Allocation 161
E.3.1 Fully-connected DNN Architecture 161
E.3.2 Output Discretization and Scaling 162
E.3.3 Transfer Learning Strategy 162
E.4 System Setup 163
E.4.1 Data Generation 163
E.4.2 Parameter Selection 163
E.4.3 Training and Testing Stage 164
E.5 Results and Discussions 164
E.6 Conclusion and Future Work 167

Bibliography **169**

List of Figures

1.1 Radio map of: (a) original wireless environment, (b) wireless environment with 4 `Txs` and 3 objects represented by 3 cubes, and (c) difference between the two radio maps in (a) and (b). 2
1.2 Application of radio maps. 2
1.3 Network planning using radio map. 3

2.1 Indoor wireless signal propagation. 12
2.2 Outdoor wireless signal propagation. 12
2.3 Multipath components in an indoor wireless environment. 13
2.4 Illustration of wireless environment showing `Tx`, `Rx` and anchor nodes. 14
2.5 A simple trilateration algorithm in 2D in case of no multipath [1]. . . 15
2.6 A simple trilateration algorithm in 2D in case of multipath [1]. 16
2.7 Baseline DNN model. 16
2.8 TL illustration of floor plans and radio maps of original (source) and target wireless environment. 18
2.9 Illustration of different radio maps corresponding to different indoor wireless environments. 20
2.10 Correlation between inter-image WD and inter-radio map WD. 20
2.11 Illustration of correlation between inter-image and inter-radio map similarities using four different similarity measures. 22
2.12 Illustration of the data-driven similarity measure. 23
2.13 A simple illustration of MoE architecture. 24
2.14 MoE architecture for radio map estimation. 25
2.15 Illustration of the TL system model for resource allocation. 25

3.1 Illustration of radio map for: (a) original (source), and (b) target wireless environment with 4 `Txs` and 3 objects represented by 3 cubes from Paper B [2]. 28
3.2 MapNet: Baseline DNN model. 29
3.3 Data-driven transfer learning based radio map estimation model (training phase). 30
3.4 Data-driven transfer learning based radio map estimation model (execution phase). 31
3.5 Correlation analysis between different types of similarity measures. . . 34
3.6 CNN-based neural structure for the similarity measure. 34

3.7 Data-driven similarity decision between the target $\mathcal{E}_t$ and source $\mathcal{E}_s$ environments 36
3.8 Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes), (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), and (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes) 38
3.9 (a) Radio map for a wireless environment of Type I (noiseless), (b) Radio map of the same environment as in (a) after adding noise to power values, (c) location error map between the true and estimated `Rxs` locations for the same environment as in (a), and (d) plot between test MSE corresponding to TL and the percentage of training data after TL for the noiseless and the noisy case corresponding to the same environment as in (a). 40
3.10 Radio map for (a) original wireless environment with `4 Txs`, (b) a wireless environment from Type III, (c) same wireless environment as in (b) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0038), (d) a wireless environment from Type V, and (e) same wireless environment as in (d) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0048). 42
3.11 Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) same environment as in (b) after performing TL with 35% training data (MSE($\mathcal{E}_t$) = 0.005), (d) another environment with additional wall and changed `Tx` locations, and (e) same environment as in (d) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0002). 42

4.1 DNN model for radio map estimation. 46
4.2 TL Performance of different methods. 47
4.3 Radio map for: (a) original indoor wireless environment, (b) indoor wireless environment where a single object, represented by a cube, is present, (c) the same wireless environment as in (b) after performing TL using both ToA features and locations of `Rxs` (Method 1), where only 40% of training data is used for training the DNN resulting in a test MSE corresponding to TL as 0.0020, and (d) the same wireless environment as in (b) after performing TL using only ToA features (location-free) (Method 2), where only 40% of training data is used for training the DNN resulting in a test MSE corresponding to TL as 0.0036. 48

5.1 Illustration of radio maps: (a) original wireless environment, and (b) target environment with 4 transmitters `Txs` and 3 objects represented by 3 cubes. 52
5.2 MoENet: Baseline MoE model. 52
5.3 Data-driven transfer learning-based radio map estimation model using MoE (execution phase). 53

5.4 Radio map for (a) original wireless environment with `4 Txs`, (b) an environment from Type IV, (c) environment (b) after performing TL using Expert 1 with 40% training data (MSE($\mathcal{E}_t$) = 0.0078), (d) environment (b) after performing TL using Expert 2 with 40% training data (MSE($\mathcal{E}_t$) = 0.006), and (e) environment (b) after performing TL using *MoENet* with 40% training data (MSE($\mathcal{E}_t$) = 0.0054). . . . 56
5.5 Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) environment (b) after performing TL using Expert 1 with 30% training data (MSE($\mathcal{E}_t$) = 0.0096), (d) environment (b) after performing TL using Expert 2 with 30% training data (MSE($\mathcal{E}_t$) = 0.0076), and (e) environment (b) after performing TL using *MoENet* with 30% training data (MSE($\mathcal{E}_t$) = 0.0071). 57

6.1 Illustration of the system model with TL approach. 60
6.2 Baseline DNN with input and output. 62
6.3 Percent of training data vs MSE (Number of epochs = 10). 63
6.4 Percent of training data vs MSE (Number of epochs = 10). 64
6.5 Sum-rate for different values of outage probability ϵ. 64

A.1 Environments for TL: Original and modified environments. 74
A.2 Radio map estimation DNN model. 76
A.3 Network structure for the CNN. 78
A.4 Radio map in (a) original indoor wireless environment, (b) environment where 20% of training data is required when executing the operation of TL, and (c) environment where 30% of training data is required when executing the operation of TL. 82

B.1 Toy example illustrating different radio maps corresponding to different indoor wireless environments. 87
B.2 Correlation between inter-image WD and inter-radio map WD. 88
B.3 Data-driven TL-based radio map estimation model (training phase). . 91
B.4 Data-driven TL-based radio map estimation model (execution phase). 91
B.5 Illustration of radio maps: (a) source (original), and (b) target environment with `4 Txs` and 3 objects represented by 3 cubes. 92
B.6 MapNet: Baseline DNN model. 92
B.7 CNN-based neural structure for the similarity measure. 98
B.8 Data-driven similarity decision between the target $\mathcal{E}_t$ and source $\mathcal{E}_s$ environments 99
B.9 Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes, (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes), and (f) Type VI $\mathcal{E}$. 101
B.10 Correlation analysis between different types of similarity measures. . . 105

B.11 (a) Radio map for a wireless environment from Type I $\mathcal{E}$, and (b) plot between test MSE corresponding to TL and percentage of training data after TL for the noiseless and noisy environment corresponding to the same environment (a). 109

B.12 Reduction in number of training epochs for each new environment. . . 110

B.13 Radio map for (a) original (source) wireless environment with `4 Txs` located at four different fixed positions, (b) a wireless environment from Type V environment, (c) same environment as in (b) after performing TL where only 40% of the training data is required to train the DNN resulting in a test MSE, i.e., MSE($\mathcal{E}_t$) as 0.0048, (d) a wireless environment from Type VI environment (changed `Tx` locations and additional wall), and (e) same environment as in (d) after performing TL where only 35% of the training data is required to train the DNN resulting in a test MSE, i.e., MSE($\mathcal{E}_t$) as 0.005. 111

B.14 Radio map for (a) a wireless environment from Type III environment, (b) same environment as in (a) after performing the TL operation using the proposed method with only 35% training data (MSE($\mathcal{E}_t$) = 0.0038), and (c) same environment as in (a) estimated using RadioUNet (MSE($\mathcal{E}_t$) = 0.4274). 113

B.15 Illustration of the source and the target cafe area wireless environments. 114

B.16 Radio map for (a) original cafe area environment, (b) a target cafe area environment, and (c) same environment as in (b) after performing TL using the proposed method with only 35% training data (MSE($\mathcal{E}_t$) = 0.0036). 116

C.1 TL environments: (a) original (source) environment, and (b) modified (target) environments. 121

C.2 DNN model for radio map estimation. 123

C.3 CNN model for similarity measure. 124

C.4 Similarity decision between two wireless environments. 125

C.5 Reliability comparison of different methods. 130

C.6 Radio map for (a) original indoor wireless environment, (b) a modified indoor wireless environment where the cube is present near the Tx, (c) same environment of (b) after executing the operation of TL using both ToA features and locations of Rxs (Method 1), where only 40% of training data is used for training the DNN resulting in a test MSE after TL as 0.00203, and (d) same environment of (b) after executing the operation of TL using only ToA features (location-free) (Method 2), where only 40% of training data is used for training the DNN resulting in a test MSE after TL as 0.00368. 131

D.1 (a) Location-error map for the original wireless environment (63×106 points with a spacing of 15 cm); Radio map for: (b) the original wireless environment with 4 `Txs`, (c) the same environment as in (b) estimated using Expert 1 (test MSE = 0.0051), (d) the same environment as in (b) estimated using Expert 2 (test MSE = 0.0050), and (e) the same environment as in (b) estimated using the MoE (test MSE = 0.0039). 136
D.2 Illustration of radio maps: (a) original wireless environment, and (b) target wireless environment with `4 Txs` and 3 cube objects represented by 3 black squares. 137
D.3 Data-driven TL-based radio map estimation model using MoE (training phase). 139
D.4 Data-driven TL-based radio map estimation model using MoE (execution phase). 139
D.5 MoENet: Baseline MoE model. 140
D.6 CNN-based neural network structure for the similarity measure. . . . 143
D.7 Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes), (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), and (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes). 146
D.8 Histogram of average location estimation error from all environments. 146
D.9 Radio map for (a) original wireless environment with `4 Txs`, (b) an environment from Type IV, (c) environment (b) after performing TL using Expert 1 with 40% training data (MSE($\mathcal{E}_t$) = 0.0078), (d) environment (b) after performing TL using Expert 2 with 40% training data (MSE($\mathcal{E}_t$) = 0.006), and (e) environment (b) after performing TL using *MoENet* with 40% training data (MSE($\mathcal{E}_t$) = 0.0054). . . . 152
D.10 Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) environment (b) after performing TL using Expert 1 with 30% training data (MSE($\mathcal{E}_t$) = 0.0096), (d) environment (b) after performing TL using Expert 2 with 30% training data (MSE($\mathcal{E}_t$) = 0.0076), and (e) environment (b) after performing TL using *MoENet* with 30% training data (MSE($\mathcal{E}_t$) = 0.0071). 153

E.1 Illustration of the system model with TL approach. 158
E.2 Architecture of fully-connected DNN with input and output. 162
E.3 Percent of training data vs MSE (No. of epochs=10). 165
E.4 Percent of training data vs MSE (No. of epochs=10). 165
E.5 Sum-rate for different values of outage probability ϵ. 166

List of Tables

3.1 Summary of wireless environment $\mathcal{E}$. 38
3.2 Transferability, F1-score, and Accuracy for Type I and all types of environments grouped together. 39

5.1 Transferability, F1-score, and Accuracy 54

7.1 Comparison of chapters 68

A.1 Parameters used in Remcom for data generation. 75
A.2 The Baseline DNN model 79
A.3 The CNN model in Case A 80
A.4 The CNN model in Case B 80
A.5 TL Performance in Case A 80
A.6 TL Performance in Case B 80
A.7 Transferability, F1-score, and Accuracy 81

B.1 Parameters used in Remcom for generating the data. 100
B.2 Summary of wireless environment $\mathcal{E}$. 101
B.3 Parameters and hyper-parameters for $MapNet$ 104
B.4 Parameters and hyper-parameters for $SimNet_A$ and $SimNet_B$ . . . 104
B.5 TL performance for Type I environment and all types of environments grouped together. 106
B.6 Transferability, F1-score, and Accuracy for Type I environment and all types of environments grouped together. 108
B.7 Parameters used in the RadioUNet [3] 112
B.8 Parameters for cafe $SimNet_A$ and $SimNet_B$ 114
B.9 Transferability, F1-score, and Accuracy 115

C.1 Remcom parameters to generate data. 121
C.2 The Baseline model learning 126
C.3 The CNN model under Case A of Method 1. 127
C.4 The CNN model under Case B of Method 1. 127
C.5 The CNN model under Case A of Method 2. 128
C.6 The CNN model under Case B of Method 2. 128
C.7 Performance of proposed TL method. 128

D.1 Parameters used in Remcom for generating the data 145

D.2 Summary of wireless environment $\mathcal{E}$. 145
D.3 Parameters and hyper-parameters for $MoENet$ 147
D.4 Parameters and hyper-parameters for $SimNet_A$ and $SimNet_B$ . . . 148
D.5 TL performance 148
D.6 Transferability, F1-score, and Accuracy 149
D.7 TL performance 149
D.8 Transferability, F1-score, and Accuracy 150
D.9 TL performance 150
D.10 Transferability, F1-score, and Accuracy 151

E.1 Model learning for Perfect CSI 164
E.2 Model learning for Imperfect CSI 165
E.3 Computational performance for perfect and Imperfect CSI 166

List of Abbreviations

AI	Artificial intelligence
Adam	Adaptive moment estimation
BRISK	Binary robust invariant scalable keypoints
CDF	Cumulative distribution function
CNN	Convolutional neural network
CSI	Channel state information
CU	Cellular user
D2D	Device-to-device
DDS	Data-driven similarity measure
DNN	Deep neural network
DoA	Direction of arrival
DoG	Difference-of-Gaussians
FDD	Frequency division duplexing
GN	Gating network
LOS	Line of sight
ML	Machine learning
MoE	Mixture of experts
MSE	Mean square error
ORB	Oriented FAST and rotated BRIEF
PSD	Power spectral density
PSNR	Peak signal-to-noise ratio
QoS	Quality of service
RA	Resource allocation
RBF	Radial basis function
ReLU	Rectified linear unit
`Rx`	Receiver
SC	Spectrum cartography
SIFT	Scale-invariant feature transform
SINR	Signal-to-interference plus noise ratio
SSIM	Structural similarity
TL	Transfer learning
ToA	Time of arrival
`Tx`	Transmitter
WD	Wasserstein distance

Chapter 1

Introduction

1.1 Motivation

A radio map contains meaningful information about the signal propagation environment of a wireless network. In the most general sense, a radio map portrays an estimate of the power spectral density (PSD) over a specified geographical area as a function of location, time, and frequency. Moreover, a radio map also depicts various characteristics of the wireless environment over a certain region, which is governed by various factors, such as reflections, diffractions from large-scale objects and buildings, and scattering from small-scale objects, which result in fluctuations in the received radio signals. For example, Figure 1.1 shows two radio maps corresponding to two wireless environments, one without objects and another one incorporating three objects represented by three cubes in black colour, respectively, showing the power distribution across space with four transmitters (`Txs`), a carrier frequency of 900 MHz with 1 MHz bandwidth. Figure 1.1(c) shows the absolute difference between these two radio maps with the three objects represented by three cubes in black colour. Moreover, it can also be seen that the received power[1] is high near each `Tx` and decreases as one moves away from each `Tx`. Note that the power values are normalized to the scale [0, 250].

There are several applications of radio maps in wireless networks. A wide range of applications of radio maps is shown in Figure 1.2. Specifically, in the case of network planning, an area having inadequate power in some region (weak coverage) can be covered appropriately by adding a new base station in that region [4], as shown in Figure 1.3. Similarly, the `Txs` locations can be inferred by inspecting a radio map [5]. For example, it can be seen in Figure 1.1 that the received power (in dBm) is higher near each `Tx`, and this information can be exploited to identify the locations of `Txs`.

To improve the performance of future wireless networks, as shown in Figure 1.2, accurate estimation of radio maps is essential for efficient network operations. The estimation of radio maps is also known as spectrum cartography (SC) [6]. To estimate radio maps, one needs to first deploy sensors or receivers (`Rxs`) in the considered

[1]Note that the term "power" is used to refer to PSD in the rest of the dissertation.

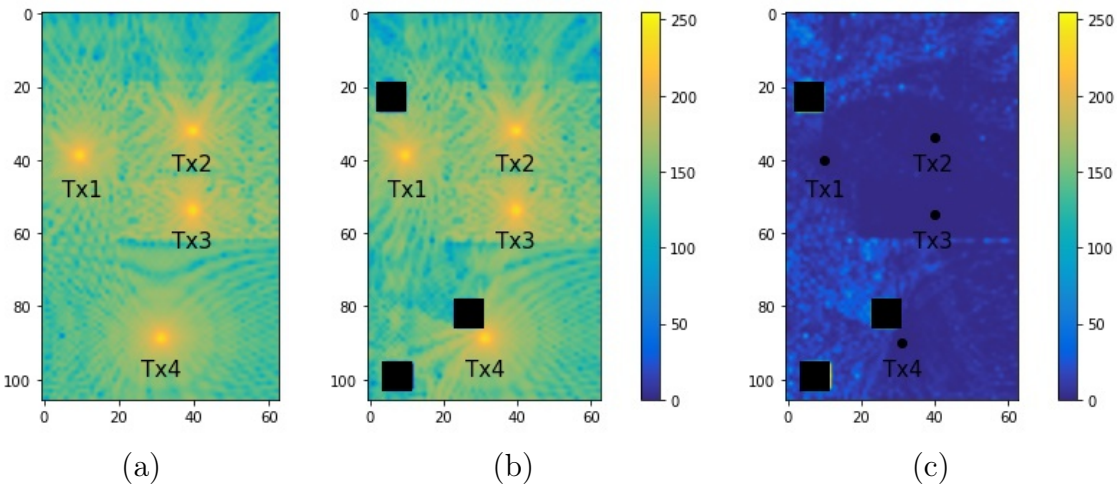


Figure 1.1: Radio map of: (a) original wireless environment, (b) wireless environment with `4 Txs` and 3 objects represented by 3 cubes, and (c) difference between the two radio maps in (a) and (b).

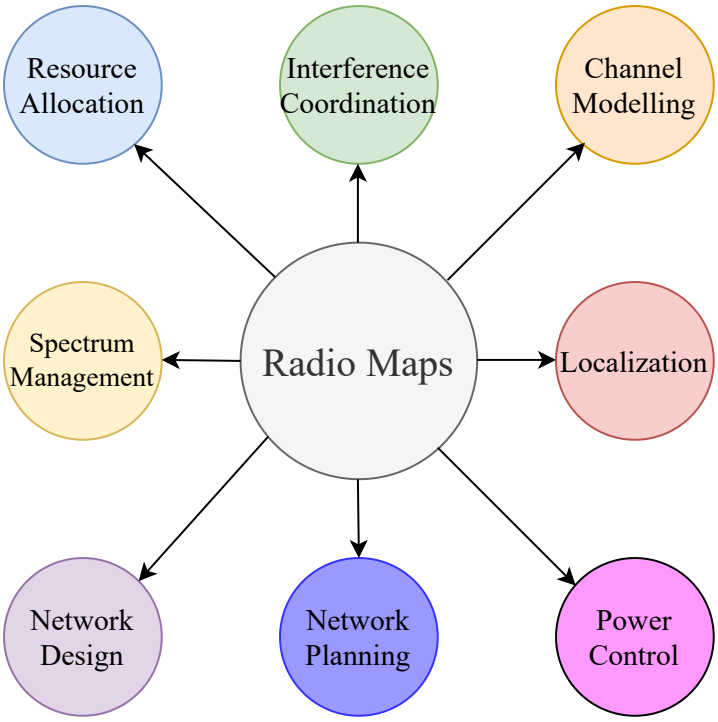


Figure 1.2: Application of radio maps.

area and then construct a map by applying some kind of interpolation or regression techniques. Different methods can be used for this, such as kernel-based methods [7] and deep learning methods, which usually require the knowledge of accurate `Rx` locations. However, in cases where there is a high degree of wireless multipath, locations of `Rxs` can not be estimated accurately. Then, time of arrival (ToA) features of radio signals can be used instead [6, 8] to estimate the radio maps. Moreover, a mixture of experts (MoE) (see **Section 2.7**), which incorporates both location information of `Rxs`, ToAs, and an estimation of the location errors, can also be used to estimate the radio maps.

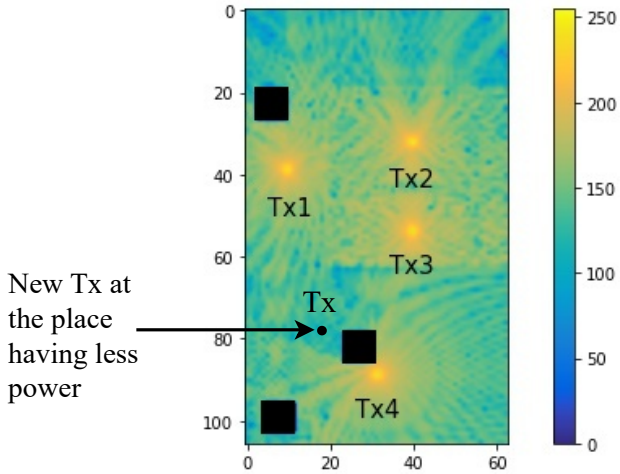


Figure 1.3: Network planning using radio map.

The propagation characteristics of wireless networks vary with the environments, thus, a radio map model learned for one wireless environment (source environment), can not be used directly in the new wireless environment (target environment). Moreover, estimating a radio map for each new wireless environment, in general, requires typically a large amount of sensing data. However, in practice, there is data scarcity, that is, lack of data as it is usually difficult and expensive to have always a sufficiently large number of measurements in each new wireless environment. Collecting a large amount of sensing data for each new wireless environment and learning the corresponding radio map model each time is time-consuming. To address these issues, we propose transfer learning (TL) methods (see **Section 2.5**) to reduce the amount of data samples. The TL approach allows us to exploit the previously learned radio map model in a source environment in order to estimate radio maps in other (new) similar wireless environments (target environment), using only some additional sensor measurement samples instead of a complete set of sensor measurements. TL requires also similarity measures [2] between different environments in order to decide whether the TL operation will be beneficial, reducing the number of sensor measurements. Intuitively, the TL operation will be beneficial when the (source) original environment and the new (target) environment are sufficiently similar. In this sense, we also design a data-driven similarity measure (see **Section 3.6.2**) to measure the similarity between the source and target wireless environments. Our designed similarity measure also predicts the amount of training data needed to perform the TL operation for a new similar wireless environment. For the sake of simplicity, in this dissertation, we focus on data-driven radio map estimation for indoor wireless environments. However, the same methodology can also be applied to outdoor wireless environments for estimating radio maps.

Resource allocation (RA) in wireless networks is a computationally demanding task. In principle, TL can be used for RA if two tasks are sufficiently similar. This will reduce the amount of computational resources needed as well as the training time.

1.2 Summerized State of Art

The estimation of radio maps has gained significant attention in wireless networks, and a number of works [3,9–11] have been done to estimate radio maps using different techniques. One can categorize these techniques into either model-based or model-free methods. Model-based techniques generally assume certain signal propagation models and then combine the PSD of signals received from the active `Txs` [9]. For example, a log distance path loss model for Wi-Fi radio map estimation is proposed in [10], but then it ignores multipath propagation. Model-free techniques explore and make use of neighbourhood information. For example, these models employ radial basis function (RBF) [12] interpolation techniques for estimating radio maps. RBF can use different types of kernels, such as Gaussian, multi-quadrics, or splines. To capture path loss and shadowing, the power map estimation algorithm in [13] uses multiple kernels. Kernel-based regression methods can also be used to estimate power maps from features (e.g., ToA-based features) obtained from the pilot signals in [11]. These aforementioned works can map the power distribution across space efficiently using either location-based or location-free features. However, these techniques are severely impaired by small and large-scale fading while estimating radio maps.

There have been also works that employ deep-learning techniques for estimating radio maps. For example, to estimate radio maps in urban environments, the authors in [3] adopt a RadioUNet model which is a modified UNet architecture [14] (originally designed for biomedical image processing application). Notice that [3] and [14] also use auto-encoders. A deep autoencoder, which learns the spatial structure of shadowing is proposed to estimate power maps in [15]. To estimate outdoor radio maps, a two-phase learning framework integrating the radio propagation model and designing a conditional generative adversarial network is proposed in [16]. A maximum likelihood estimation-based indoor radio map estimation under a Gaussian quantizer, is proposed in [17]. However, these works require training the model from the very beginning for each new wireless environment, resulting in the need for a huge amount of training data, which is typically difficult and expensive to obtain. Moreover, without TL, one also needs to train a new deep learning model from the very beginning when the wireless environment changes, as illustrated in Figure 1.1. Given these difficulties, it is very challenging to learn efficient deep-learning models for estimating radio maps in each new wireless environment.

As mentioned previously, one promising solution to deal with the data scarcity problem is the use of transfer learning (TL) methods [18, 19]. TL has been seen to achieve promising results in several applications of wireless networks and in handling the data scarcity issue, reducing the amount of training samples. For example, for capturing traffic pattern diversity in the cellular data of distinct cities, the work in [20] proposes a spatial-temporal cross-domain neural network (STC-Net). The work in [20] uses model-based TL, where the complete source model is transferred to the target environment, exploiting similarities between distinct types of cellular traffic across distinct cities. For improving the robustness of deep neu-

ral networks (DNN) based spectrum sensing in a cognitive radio scenario, TL is used in [21], where it is assumed that the data collected under distinct characteristics belong to distinct but related distributions. In [22], downlink channel state information (CSI) prediction from uplink CSI using direct model-based TL, is proposed for frequency division duplexing (FDD) in a massive multiple-input multiple-output (MIMO) framework. In [23], wireless fingerprinting localization uses a TL method incorporating the Wasserstein distance [24]. A TL-based method incorporating the Kullback-Leibler (KL) divergence index as a similarity measure, is used for predicting the antenna tilt-dependent radio map in [25]. A two-phase TL generative adversarial network (TPTL-GAN) for estimating power spectrum maps for underlay cognitive radio networks, is proposed in [26], however, this work is tailored for underlay cognitive radio applications. To sum up, to the best of our knowledge, there is no work in the literature that employs transfer learning for estimating radio maps in either indoor or outdoor wireless environments and predicts the amount of training data required for the new target wireless environments.

There have been few works that use TL for optimization problems in different applications. The authors in [27] propose a TL-based dynamic multiobjective evolutionary algorithm (EA) which integrates TL and population-based EAs to solve the dynamic multiobjective optimization problems (DMOPs), in which the optimization functions change over time in varying environments. To maximize the production net present value (NPV) throughout the expected life of the reservoir, a TL-based optimization framework is proposed in [28] for dynamic production optimization problems. A TL-based parallel evolutionary algorithm framework for bi-level optimization problems is proposed in [29], which conducts a parallel lower-level search (LLS) in bi-level optimization. A non-iterative topology optimization method using TL based on a convolutional neural network architecture is proposed in [30] for different 3D design explorations. In [31], a TL-based framework for learning to optimize for resource management (LORM) is proposed, which achieves near-optimal performance for mixed-integer nonlinear programming (MINLP) resource management problems in wireless networks. For tackling an NP-hard mixed-integer nonlinear programming problem of resource allocation, a TL method via self-imitation is proposed in [32]. However, to the best of our knowledge, none of the works present in the literature has used TL between two different optimization problem solvers for two different but similar tasks in wireless networks, in particular, for the resource allocation task.

1.3 Problem Statements and Main Contributions

This dissertation addresses four problem statements which are derived from the literature review:

Q1. TL for Location-based Radio Map Estimation: How can we estimate accurate radio maps in a new target wireless environment where there is data scarcity, that is, an insufficient amount of measurement data to estimate the

radio map?
This problem is presented in papers A and B. The main contributions of these works can be summarized as follows:

- Design of an effective data-driven TL method that transfers and fine-tunes a DNN-based model for a radio map learned from a given source environment to another target environment.
- Design of a data-driven similarity measure that is used to decide whether to perform the TL operation in a new target wireless environment, that outperforms other widely used similarity measures.
- Prediction of the amount of training data needed to estimate the radio map in a different target environment, when performing the TL operation, based on our data-driven similarity measure.
- Analysis of the robustness of the proposed TL scheme under noisy environments when the locations of receivers are estimated and not perfectly known, and the corresponding power values are noisy. Our proposed TL scheme shows satisfactory performance (Paper B).
- Extensive testing of the proposed TL scheme using simulated data from the Remcom simulator [33]. Results demonstrate that our proposed scheme can perform effective TL in most of the tested environments while using a small amount of training data. Moreover, the proposed data-driven similarity measure can accurately recommend TL guaranteeing a satisfactory radio map estimation while reducing the number of necessary training samples.

Q2. TL for Location-Free Radio Map Estimation and its Comparison: Due to a high degree of wireless multipath during the signal propagation, it might be difficult to obtain the exact `Rxs` locations. How can we estimate accurate radio maps in a new wireless environment without the exact knowledge of `Rxs` locations and having insufficient amount of measurement data? Moreover, what are the advantages and disadvantages of this method with respect to a location-based method?
This problem is presented in paper C. The main contributions of this work can be summarized as follows:

- Design of an efficient DNN-based model that learns a radio map for an indoor wireless environment using ToA features and/or `Rxs` locations. In addition, we adapt the previously proposed data-driven TL method to these newly designed models.
- Extensive testing of our algorithms using simulated data from the Remcom simulator [33]. Our simulation results demonstrate that employing only ToA (location-free) feature is better in estimating accurate radio maps when the location information is not accurately known.

- Numerically, we show that the proposed TL method employing only ToA (location-free) feature requires less amount of training data as compared to the location-based TL method, as presented in Paper A. It is also shown experimentally that a similarity measure based on the Wasserstein distance (WD), which is widely used in TL, is also not applicable for the location-free radio map estimation application.

Q3. TL for Radio Map Estimation using Mixture of Experts: In a given wireless environment, there might be both an inexact knowledge of `Rxs` locations and a large number of wireless multipaths. The inaccuracy of `Rxs` locations and the degree of wireless multipath will change across space in the environment. Since there are cases where the location-free is better (e.g. high degree of wireless multipath) than the location-based method, and some cases where the location-based is better (e.g. lower degree of wireless multipath) than the location-free method, how can we leverage both location-based and location-free methods to estimate accurate radio maps in a new wireless environment which has an insufficient amount of measurement data?
This problem is presented in paper D. The main contributions of this paper are:

- Designing a MoE-based radio map model which mixes two experts, i.e., location-based and location-free experts.
- Adapting data-driven TL method for a MoE-based radio map model learned from an original wireless environment that is transferred and fine-tuned for other similar wireless environments.
- Extensive testing of the proposed TL method using simulated data from the Remcom simulator [33]. Results show that the proposed method performs an effective TL for estimating radio maps in most varying wireless environments while using a relatively small amount of training data. Numerically, the proposed method outperforms the individual location-based and location-free experts.

Q4. TL-based Resource Allocation for Underlay D2D Communications: One source of the applications of radio maps is resource allocation (RA) in wireless networks. In this context, our goal is to understand the process of TL between similar resource allocation optimization problems. How can we jointly allocate channels and power values to a given imperfect channel state information (CSI) scenario with less amount of training data, using additional information from the perfect CSI scenario?
This problem is presented in paper E. The main contributions of this paper are:

- We design a baseline DNN model for the data generated from a perfect CSI scenario and then transfer it to an imperfect CSI scenario and

fine-tune it with an additional small amount of data generated from the imperfect CSI scenario.

- We show how the desired outage probability ϵ can be used to measure the similarity between two scenarios. For higher values of ϵ, less amount of training data is required for fine-tuning the transferred baseline DNN model in the imperfect CSI scenario.
- Numerically, we show that the sum-rate obtained by training the imperfect CSI model using TL, is higher than the DNN without TL. At the point when both RA tasks are equivalent ($\epsilon = \frac{1}{e}$), TL achieves approximately the same sum rate as the original algorithm while saving computational resources.

1.4 Outline of the Dissertation

The dissertation is based on five papers which are attached in the Appendix and are organized as the following chapters.

- *Chapter 2* This chapter provides the background theory needed for the readers in order to follow the contents of this dissertation. In this chapter, we introduce wireless signal propagation including multipath propagation, time of arrival (ToA) and estimation of ToA, the trilateration approach, the framework of deep neural networks (DNNs) and convolutional neural networks (CNNs), the concept of transfer learning and the different possible similarity measures, the mixture of experts (MoE), and TL for resource allocation problem, which are used in this dissertation.

- *Chapter 3* This chapter summarizes Paper A [34] and Paper B [2], which answer **Q1**. It discusses the problem of accurate radio map estimation in wireless networks. We propose a transfer learning-based method that transfers and fine-tunes a DNN-based model for a radio map learned from an original (source) wireless environment to other new (target) wireless environments. The DNN-based radio map model is learned with the exact knowledge of `Rxs` locations and we refer to it as the *location-based TL method*. Our method checks the TL recommendation based on our designed data-driven similarity measure (DDS), and then predicts the amount of training data needed to perform the TL operation in new wireless environments to estimate the radio maps with a given accuracy. We show the superiority of our DDS as compared to other standard similarity measures, such as the Wasserstein distance. Additionally, it investigates the noise robustness in both `Rxs` locations and power values of the proposed TL method.

- *Chapter 4* This chapter summarizes Paper C [8], which answers **Q2**. It discusses the problem of radio map estimation in wireless networks when the exact knowledge of `Rxs` locations is not available due to wireless multipath.

However, time of arrival (ToA) based (location-free) features of radio signals can be easily obtained. We propose a TL-based method that transfers and fine-tunes a DNN-based radio map model learned from an original (source) wireless environment to other new (target) wireless environments using location-free features. The DNN-based radio map model is learned using the ToA features and we refer to it as the *location-free TL method*. Similarly, as before, the method checks the TL recommendation based on our designed DDS, and then predicts the amount of training data needed to perform the TL operation in the new wireless environment to estimate the corresponding radio map with a given accuracy. Again, our similarity measure shows a superior performance as compared to other standard similarity measures, such as the Wasserstein distance. Moreover, comparisons are made with the location-based TL method which employs only exact `Rxs` locations as in Paper A [34] and the method which employs both the `Rxs` locations and ToA features, assuming that exact `Rxs` locations are known.

- *Chapter 5* This chapter summarizes Paper D [35], which answers **Q3**. It discusses the problem of radio map estimation in wireless networks when we exploit both inaccurate `Rx` location information and ToA-based information. It introduces a mixture of experts (MoE) method to estimate radio maps in wireless networks. The MoE incorporates both an expert based on estimated `Rxs` locations and another expert based on ToA features, which are combined by a gating network. The MoE-based radio map model is learned for an original (source) wireless environment and then transferred and fine-tuned to other new (target) wireless environments. The method checks the TL recommendation based on our designed DDS and then predicts the amount of training data needed to perform the TL operation in new wireless environments to estimate the radio map with a certain accuracy. Additionally, the amount of computation time required for executing a MoE model, which incorporates three DNNs inside it, will be always higher than the computation time required for executing an individual expert, such as location-based or location-free expert, which incorporates only a single DNN. To this end, we can choose a particular expert to perform the TL operation in new wireless environments by selecting an appropriate threshold value experimentally.

- *Chapter 6* This chapter summarizes Paper E [36], which answers **Q4**. It discusses the problem of joint resource allocation (channel assignment and power allocation) in the context of underlay D2D communications, where TL is designed to transfer solutions between similar resource allocation problems. We propose a TL-based method that transfers and fine-tunes a DNN model, previously trained to provide the solution for the perfect channel state information (CSI) scenario, to provide the solution for the case of imperfect (CSI) scenarios. In this case, the similarity measure between the two types of resource allocation problems is given by the outage probability. We show how a significant amount of training samples are saved using the TL operation.

- *Chapter 7* This chapter discusses the main conclusions of the dissertation. In addition, several possible future directions are also presented in this chapter.

Chapter 2

Background

Every aspect of past and present communication networks have been modelled typically by mathematical (physics-based or stochastic-based) models, that are either derived from theoretical considerations, or empirically from field measurement campaigns. However, we are rapidly reaching the point at which the quality and heterogeneity of the services we demand from communication systems, will exceed the capabilities and applicability of present modelling and design approaches. To provide the end-users with a perceived seamless and limitless connectivity, the reconfiguration of network resources and/or the deployment of additional network nodes in response to new data demands (e.g., lower delay, higher throughput) must be prompt and efficient. The traditional models are difficult to adapt to the changes in the data demands and the propagation characteristics of the wireless networks. However, the data-driven artificial intelligence (AI)-based models are adequate to adapt to these changes in wireless networks. AI-based smart wireless networks [37] have the ability to sense the surrounding wireless environments and then adapt to those wireless environments. In this context, a radio map provides essential information for smart wireless networks as it depicts the various characteristics of wireless signal propagation and the environment over a certain region. Radio signal propagation is generally affected by various factors, such as reflections, diffractions, and scattering, resulting in fluctuations in the received radio signals. In other words, radio maps portray an estimate of the power spectral density (PSD) over a required geographical area as a function of location, time, and frequency. Usually, received signal powers are obtained through sensor measurements at locations of receivers `Rxs`. In this dissertation, we obtain these power values through high-fidelity simulators, such as Remcom [33].

2.1 Wireless Signal Propagation

Ideally, wireless signals travel in a straight line from the transmitter (`Tx`) to the receiver (`Rx`). That is, these signals travel through the direct line of sight (LOS) traversing the shortest distance, resulting in the reception of noiseless signals at the `Rx`. However, in real scenarios, the LOS signals might be blocked by some objects in the environment. Additionally, non-LOS signals might reach the `Rxs`,

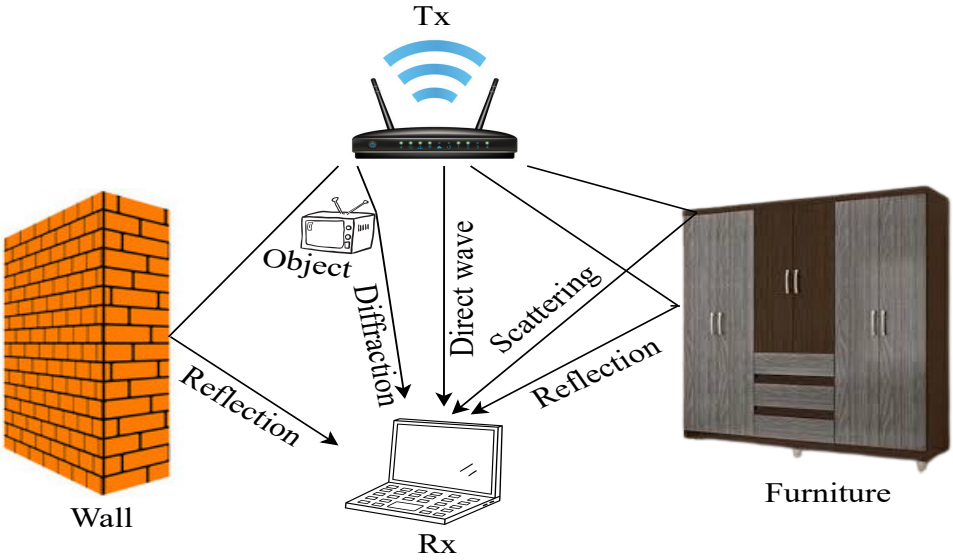


Figure 2.1: Indoor wireless signal propagation.

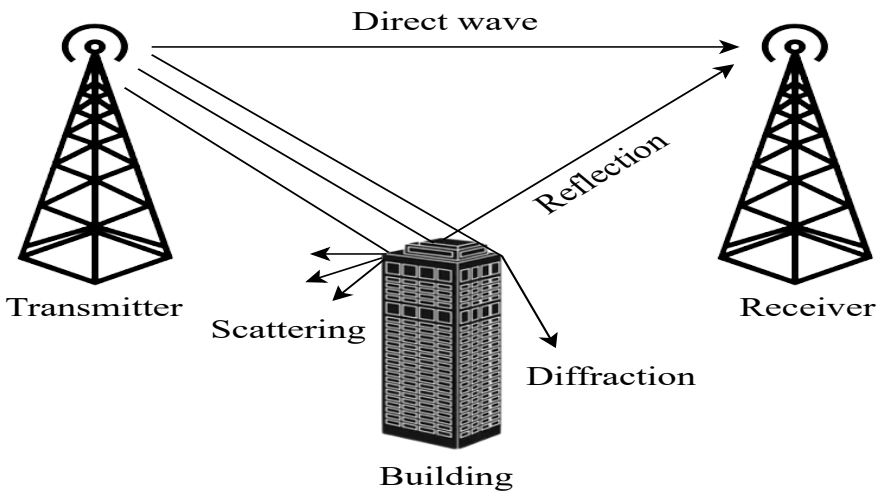


Figure 2.2: Outdoor wireless signal propagation.

after being reflected on walls or objects or passing through various objects in the surroundings. As a result, the received signals will not be only the perfect LOS signals, resulting in several multipath signals. Examples of these objects are: doors, walls, furniture, fire cabinets, metal objects, glass, and concrete in the context of indoor wireless networks; and buildings, mountains, roads, and rain-water molecules of the atmosphere in the context of outdoor wireless networks. These objects impose different effects on the propagation of radio signals, such as reflection, diffraction, and scattering [38]. Figure 2.1 and Figure 2.2 show the indoor and outdoor wireless signal propagation, respectively.

- Reflection: When the propagating signal falls on an object whose size is much larger than the wavelength of the signal, then it is reflected. For example, in an indoor scenario, as shown in Figure 2.1, the wall and furniture reflect the propagating signals. On one side, reflection degrades the signal quality. On the other side, it helps the signal to reach the `Rx` points where the LOS signal

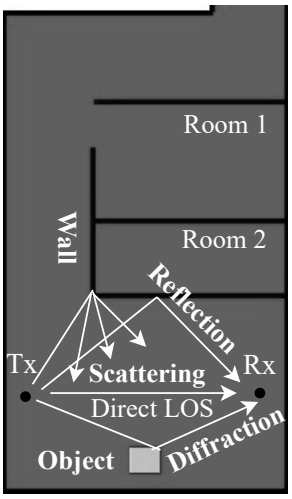


Figure 2.3: Multipath components in an indoor wireless environment.

cannot reach.

- Diffraction: Diffraction of radio waves is caused by sharp edges and corners. The phenomenon causes the bending of the radio waves as they pass through the edge of the object. For example, corners of buildings, furniture in indoor environments, and the back of trees in outdoor environments. Diffraction results in a change in the direction of the wave path from the normal LOS path.

- Scattering: It happens when the dimensions of the objects in the path of the radio waves are very small compared to the wavelength of the waves. The incoming signal is scattered into several weaker outgoing signals. For example, objects, and sharp edges of walls or furniture cause the signal to scatter in indoor scenarios. Similarly, dust, street signs, lamp posts, sharp edges of buildings, and water particles in the air cause the signal to scatter in outdoor scenarios.

2.1.1 Multipath Propagation

As explained before, when radio waves leave the transmitter, due to reflection, diffraction, and scattering, they propagate in many directions. Hence, the multipath propagation implies that the signals reach the `Rx` through multiple paths. In other words, when the signal travels from `Tx` to `Rx` through the radio channel, it is reflected, diffracted, or scattered by the objects in the indoor wireless environment, such as walls, objects as shown in Figure 2.1. As a result, the signal does not only follow the direct LOS path but follows multiple different propagation paths, forming a multipath signal at the receiver. Each multipath component reaches the `Rx` with different amplitude, delay, and phase shifts. For example, it can be seen in Figure 2.3 that the signal reaches the `Rx` through many paths. One path is a direct LOS, and on the other paths, the radio waves encounter reflection, diffraction, and scattering.

This multipath wireless propagation has an influence on the estimation of radio maps. Usually, to estimate accurate radio maps, one needs to have an exact knowledge of the `Rx` locations. One can estimate the `Rx` locations by means of different localization methods, such as the Trilateration [1] method, which uses the estimated time of arrival (ToA) information of radio signals. However, in the presence of wireless multipath, it is difficult to obtain accurate `Rx` locations. Alternatively, one can use the ToA features directly instead of the `Rx` locations for estimating radio maps, as we discuss in **Chapter 4**.

Next, we explain how ToA features are obtained from radio signals.

2.2 Time of Arrival

In practice, we need to have several anchor nodes to estimate the locations of `Txs` and `Rxs` as shown in Figure 2.4. In general, these anchor nodes should transmit pilot signals in different frequency bands or different time slots depending on the protocols in order to not interfere with the `Tx` signals. From these pilot signals, we can use super-resolution algorithms, such as Roy-Kailath's ESPRIT (Estimation of Signal Parameters via Rotational Invariance Technique) [39] and Hua-Sarkar's MPM (Matrix Pencil Method) [40] to estimate the ToA in practice. ToA is the time at which the radio signals from each `Tx` arrive at each of the `Rxs`.

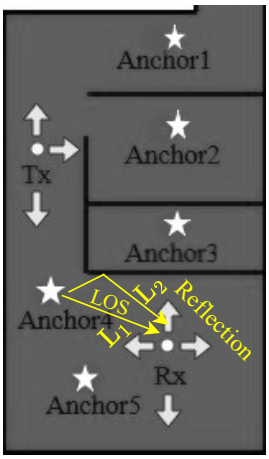


Figure 2.4: Illustration of wireless environment showing `Tx`, `Rx` and anchor nodes.

In the context of our experiment in this dissertation, ToA features are accurately calculated using the high-accuracy ray-tracing X3D ray model [41], using the Remcom [33] software. To this end, a single ray from the transmitter to the receiver point is constructed. The ray path is used to determine the distance, the ToA, and the direction of arrival (DoA). The ToA for each propagation path i is given as:

$$t_i = L_i/c \tag{2.1}$$

where L_i is the total geometrical path length. For example, as shown in Figure 2.4, L_1 and L_2 show two different paths between Anchor4 and receiver `Rx`. c is the speed of light in free space. The unit of ToA is seconds.

As discussed previously, location-based methods require to have exact `Rx` locations to estimate accurate radio maps. However, it is difficult to obtain exact `Rx` locations in the presence of noise, measurement errors, and most importantly, when there is a high degree of wireless multipath. We can estimate approximate `Rx` locations using different localization algorithms, such as the trilateration [1], squared range least squares (SR-LS) [42], squared range difference least squares (SRD-LS) [42], squared range iterative reweighted least squares (SRIR-LS) [43], and iterative re-weighting squared range difference least squares (IRWSRD-LS) [44]. Note that in this PhD thesis, our purpose is not to improve the localization algorithm but to incorporate the location uncertainty information in our TL problem of radio map estimation.

2.3 Trilateration Algorithm

One can estimate the approximate `Rx` locations from the ToA features using the trilateration localization algorithm [1]. The algorithm determines the approximate locations of `Rxs` based on the simultaneous ToAs from multiple anchor nodes at known locations that transmit pilot signals. Note that we can use the same `Txs` as anchor nodes if we have enough of them with known locations. Otherwise, we need to use additional dedicated anchor nodes for localization purposes. Specifically, it finds the intersection of three circles in the 2D case or the intersection of four spheres in the 3D case, where a system of quadratic (non-linear) equations is solved. For example, as shown in Figure 2.5, point S is the intersection point of three circles having centres $A(X_A, Y_A)$, $B(X_B, Y_B)$, and $C(X_C, Y_C)$, respectively, and it is estimated by considering the intersection.

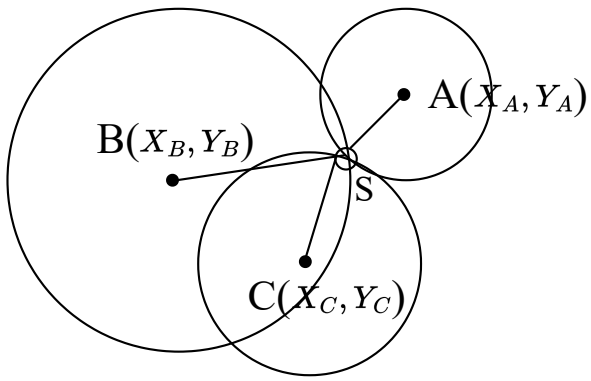


Figure 2.5: A simple trilateration algorithm in 2D in case of no multipath [1].

As shown in Figure 2.5, when there is no wireless multipath then the ToA features are accurate, resulting in the estimated `Rx` location at the intersection point of three circles (see point S). However, in the case of wireless multipath, the ToA features are inaccurate, resulting in these three circles generally not intersecting, as shown in Figure 2.6. Therefore, the localization algorithm will return an inaccurate estimate of the `Rx` location.

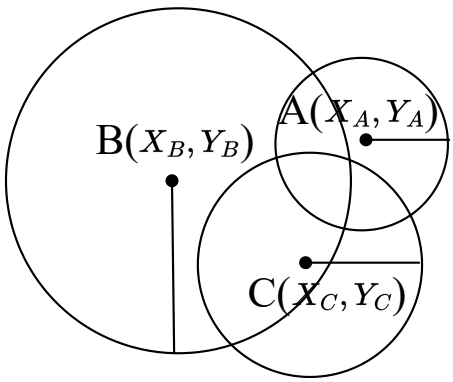


Figure 2.6: A simple trilateration algorithm in 2D in case of multipath [1].

2.4 Deep Neural Network

In addition, we need to map the locations of `Rxs` with the corresponding received signal power values. For this purpose, we use a deep neural network (DNN) as a universal function approximator [45].

A DNN [45] architecture consists of multiple layers between the input and output, each of which consists of a linear operation followed by a point-wise non-linearity, also known as an activation function, as shown in Figure 2.7. Consider a feed-forward DNN with L layers, labelled $l = 1, ..., L$ and each with a corresponding dimension q_l. The layer l is defined by a linear operation $\mathbf{W}_l \in \mathbb{R}^{q_l - 1 \times q_l}$ followed by a non-linear activation function $\sigma_l : R^{q_l} \rightarrow R^{q_l}$. Layer l receives input from the $l-1$ layer denoted as, $\mathbf{Z}_{l-1} \in R^{q_{l-1}}$, the resulting output of the layer l, $\mathbf{Z}_l \in R^{q_l}$, is then computed as $\mathbf{Z}_l := \sigma_l(\mathbf{W}_l \mathbf{Z}_{l-1})$, where $\sigma_l(\cdot)$ is point-wise activation function. The final output $\mathbf{Z}_L$ of the DNN is then related to the input $\mathbf{Z}_0$ by propagating through the various layers of the DNN as $\mathbf{Z}_L = \sigma_L(\mathbf{W}_L(\sigma_{L-1}(\mathbf{W}_{L-1}(....(\sigma_1(\mathbf{W}_1 \mathbf{Z}_0))))))$. The DNN learns the layer-wise weights $\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_L$. The DNN training is performed by going through the training data either one-by-one or batch-by-batch. Going through the whole training dataset is called one training epoch. The whole training is finished by going through multiple training epochs until the weights converge. The number of training epochs can be used to measure the training time of the

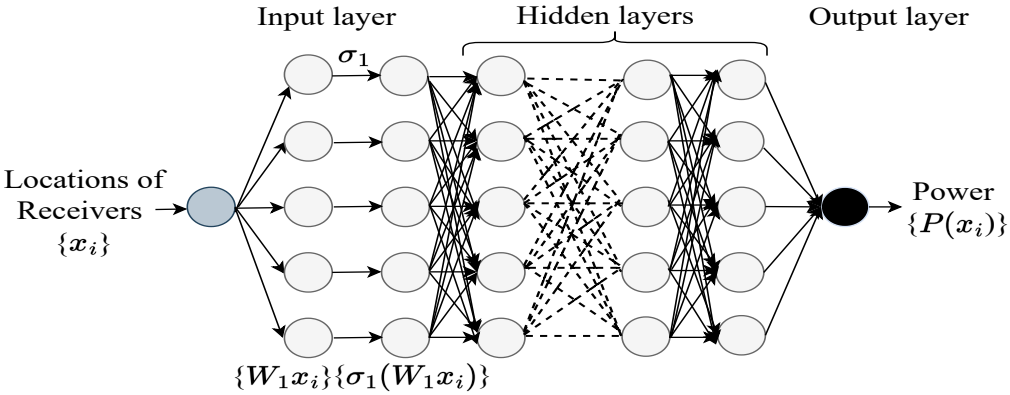


Figure 2.7: Baseline DNN model.

DNN. There are different types of activation functions available in the literature, such as Linear, Sigmoid, Tanh, ReLU, and LeakyReLU [46]. In our experiment, we perform grid search and random search [47] to find the best activation function σ_l, we then obtained a rectified linear unit function (commonly referred to as ReLU) as the best activation function. Similarly, there are different types of optimizers in the literature, such as Gradient Descent, Stochastic Gradient Descent, Mini-Batch Gradient Descent, RMSProp, and Adaptive Moment Estimation (Adam) [48]. In our experiment, we perform grid search and random search [47] to find the best optimizer, we then obtained Adam optimizer as the best optimizer.

Our model should map the location of receivers $\{x_i\}$ to the corresponding power values $\{P(x_i)\}$. We chose a DNN-based model since DNNs are universal function approximators. Moreover, DNN can learn directly high-level features from the data, which do not require domain expertise and manual feature extraction as compared to other machine learning techniques [49]. Figure 2.7 shows a baseline DNN model with its input and output.

A special type of DNN is called convolutional neural network (CNN), which is described in the next section.

2.4.1 Convolutional Neural Network

Convolutional neural network (CNN) [50] is one of the most important networks in the field of deep learning. It is a type of feed-forward neural network which extracts features from data with convolution structures. It assumes and leverages the spatial correlation of the input features, which makes it suitable for image and video processing applications or any similar application where the inputs have a high degree of spatial correlation. (In our case, CNN takes the images of floor plans as input.) One needs typically four components to build a CNN model. Convolution is an important step for extracting features, known as feature maps. One may lose information at the border while selecting a certain size of the convolutional kernel. Hence, padding is also used to enlarge the input with zero value so that the size can be adjusted. Stride is used to control the density of convolving. The larger the stride, the lower the density. After convolution, feature maps comprise a large number of features which may be prone to overfitting. Hence, pooling (down-sampling) is used to remove redundancy. Pooling can be max pooling or average pooling.

Next, effective training of the DNN model or designing any AI-based wireless network requires a sufficiently large amount of sensor measurements, which might not be available. To cope with the data scarcity problem, one can exploit transfer learning (TL).

2.5 Transfer Learning

Transfer learning (TL) [18, 19] is the exploitation of learning in a new task through the transfer of knowledge from a related task. In the context of radio map estimation, the use of knowledge acquired in one wireless environment (source environment) to assist the learning task in the new wireless environment (target environment) is

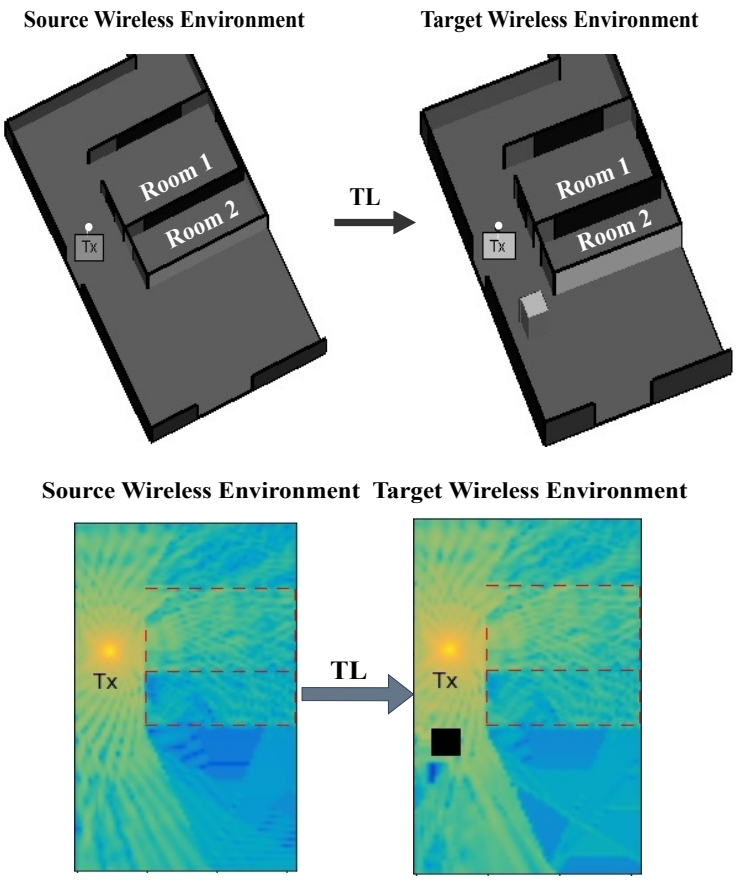


Figure 2.8: TL illustration of floor plans and radio maps of original (source) and target wireless environment.

referred to as TL. Hence, TL can handle the data scarcity issue in the target environment. In this regard, one needs to answer three main questions while performing the operation of TL, namely *"what to transfer"*, *"how to transfer"*, and *"when to transfer"*. Regarding *"what to transfer"*, the specific TL approach is referred to as *transductive transfer* [19] in which the baseline model trained under the source environment is transferred to a similar target environment. Similarly, for *"how to transfer"*, the approach is *parameter transfer* in which the parameters from the trained baseline model under the source wireless environment are transferred to a similar target wireless environment and fine-tuned with a small additional amount of training samples. Similarly, the *"when to transfer"* is associated with the similarity measure between the source and the target wireless environments, that is, the transfer is performed when both wireless environments are sufficiently similar under a certain similarity measurement. In other words, one does not need to train a model from the very beginning for a similar task, instead, the knowledge from the pre-trained model can be utilized and the amount of necessary training samples to estimate an accurate radio map, can be drastically reduced.

Our focus in this dissertation is related to the application of TL for the estimation of radio maps in wireless networks. For example, Figure 2.8 shows a simple illustration of the use of TL to estimate the radio map of the target wireless environment from the source wireless environment.

The different similarity measures used for TL are discussed in the next section.

2.6 Similarity Measures for Radio Maps

A crucial element of TL is that one needs to have a measure of similarity between the source and the target wireless environments because similarity affects the effectiveness of the TL operation. The more similar the source and the target wireless environments are, the more features can be shared between both wireless environments, and the more effective is the TL operation, resulting in less amount of training data needed to perform TL operation in the target wireless environments. To this end, one can consider different classical similarity measures, which are discussed further.

Wasserstein distance: Wasserstein distance (WD) [51] is a popular metric to calculate the distance between two probability distributions. It has been widely used in TL due to its symmetry, smooth gradients, and good numerical results [52]. It is based on the minimal cost that is incurred to transform one distribution (e.g., represented by an image histogram, in our case, the image of environment $\mathcal{E}$) into another one. It is also called earth mover's distance or transportation distance and it is an outcome of the optimal transport theory [53].

In the context of TL, the random variables we want to compare can be either the pixel value distributions of the images or radio maps representing the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environment. For a random variable in the source $\mathcal{E}_s$ and target environments $\mathcal{E}_t$, denoted by $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$, with distribution functions $f_{X_{\mathcal{E}_s}}(x_{\mathcal{E}_s})$ and $f_{X_{\mathcal{E}_t}}(x_{\mathcal{E}_t})$, and joint distribution function $f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(x_{\mathcal{E}_s}, x_{\mathcal{E}_t})$, the WD between $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$ is defined as [51]:

$$\begin{aligned} W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) &= \inf_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}} \in \mathcal{F}} \int_{\mathcal{X}_{\mathcal{E}_s}\mathcal{X}_{\mathcal{E}_t}} |X_{\mathcal{E}_s} - X_{\mathcal{E}_t}| \, df_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(x_{\mathcal{E}_s}, x_{\mathcal{E}_t}) \\ &= \inf_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}} \in \mathcal{F}} \mathbb{E}_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}} |X_{\mathcal{E}_s} - X_{\mathcal{E}_t}| \end{aligned} \tag{2.2}$$

where $\mathcal{F}$ is the collection of all possible joint distributions. $f_{X_{\mathcal{E}_s}}$ and $f_{X_{\mathcal{E}_t}}$ are the marginal distributions. $df_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(X_{\mathcal{E}_s}, X_{\mathcal{E}_t})$ is the derivative of joint distribution of variables $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$. $\mathbb{E}_{f_{X_{\mathcal{E}_s}X_{WE_t}}}$ is the expectation operation with respect to $f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}$. Notice that WD is symmetric, that is, $W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) = W(X_{\mathcal{E}_t}, X_{\mathcal{E}_s})$.

Equivalently, it is shown in [51] that equation (2.2) can be re-written as:

$$W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) = \int_{[0,1]} |f^{-1}_{X_{\mathcal{E}_s}}(X) - f^{-1}_{X_{\mathcal{E}_t}}(X)| \, dX \tag{2.3}$$

where, $f^{-1}_{X_{\mathcal{E}_s}}(X)$, and $f^{-1}_{X_{\mathcal{E}_t}}(X)$ are the inverse distribution functions in the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments, respectively.

Given a sufficiently large number of realizations (samples) of variables $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$, (2.3) can be numerically estimated.

As shown in Figure 2.8, the WD between the floor images and the corresponding radio maps are 7.1×10^{-5} and 16.3×10^{-5}, respectively.

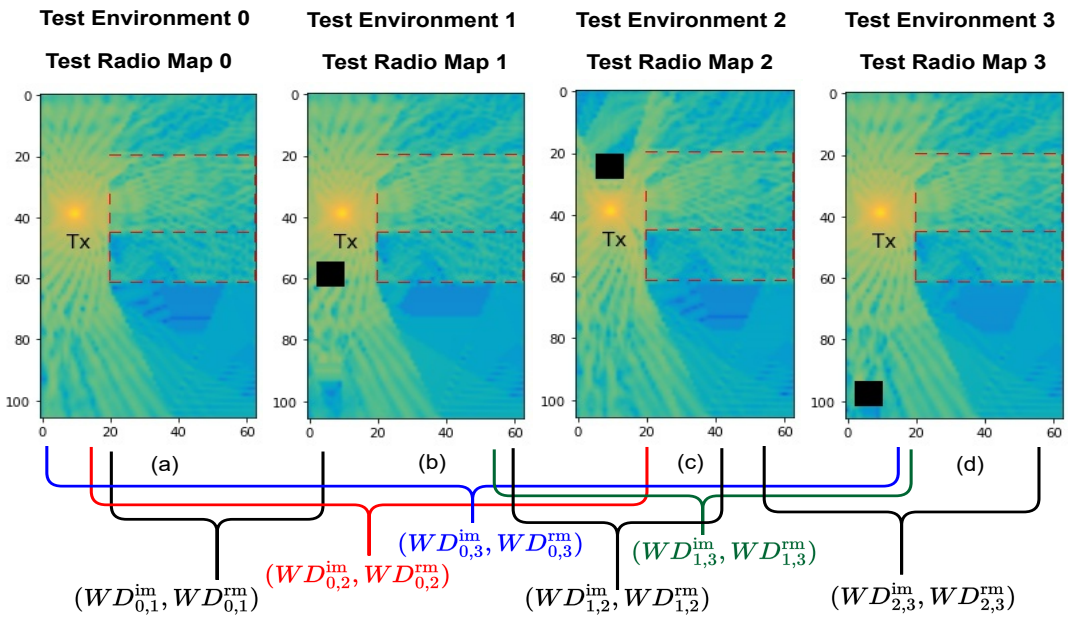


Figure 2.9: Illustration of different radio maps corresponding to different indoor wireless environments.

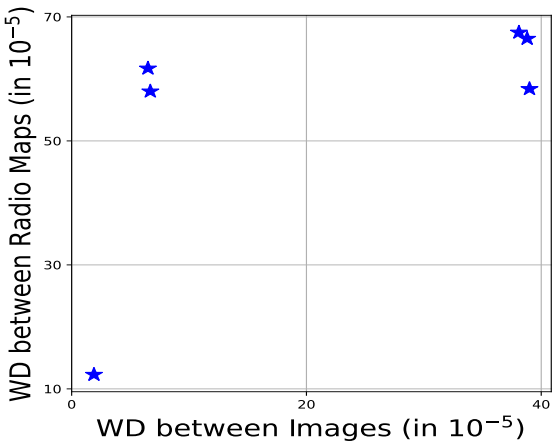


Figure 2.10: Correlation between inter-image WD and inter-radio map WD.

Next, we also consider four different indoor wireless environments as shown in Figure 2.9. Test environment 0 (TE 0) shows the radio map of the original wireless environment. Similarly, Test Environment 1 (TE 1), Test Environment 2 (TE 2), and Test Environment 3 (TE 3), respectively, show the radio maps of three different target environments in which the object (cube) is placed at different locations.

Let $\text{WD}^{\text{im}}_{i,j}$ denotes the WD between the images (im) of TE i and TE j where i, $j \in \{0,1,2,3\}$, and $i \neq j$. For instance, $\text{WD}^{\text{im}}_{0,1}$ denotes the Wasserstein distance between the images of test environment 0 (TE 0) and test environment 1 (TE 1). Similarly, $\text{WD}^{\text{rm}}_{i,j}$ denotes the WD between radio maps (rm) of TE i and TE j where i, $j \in \{0,1,2,3\}$, and $i \neq j$. The WD between images of test environments and their corresponding radio maps are shown in Figure 2.10.

Figure 2.10 shows a mismatch between the inter-image WD and the inter-radio map WD, which gives us 6 points, given that WD is a symmetric distance. The values of WDs (shown by blue stars) show no correlation between the distances in

the image domain (floor plan) and the radio maps domain. Ideally, for the WD to work appropriately as a similarity measure, both WDs should be properly aligned, as the cube location changes. This signifies that the WD is not a suitable similarity measure in our TL problem of radio map estimation.

KAZE: KAZE[1] [54] is a scale, rotation and affine invariant two-dimensional (2D) feature detector algorithm, that uses a designated feature descriptor. The feature detector detects feature points in terms of blobs[2] in an image using scale normalized determinant of the Hessian matrix [55], computed at multiple scale levels. The maxima of the feature detector response are picked up as feature points using a moving window. After feature detection, feature matching is performed using the $L1$-norm for the string-based feature descriptors and setting a threshold.

SIFT: Scale-invariant feature transform (SIFT) [56] is also a scale, rotation and affine invariant 2D feature detector algorithm, that uses a designated feature descriptor. The feature detector in terms of blobs in an image is based on the Difference-of-Gaussians[3] (DoG) operator. Feature points are detected by searching local maxima using DoG at various scales of the subject images. The description method extracts a 16×16 neighbourhood around each detected feature and further segments the region into sub-blocks, rendering a total of 128 bin values. After feature detection, feature matching is performed using the $L1$-norm for string-based feature descriptors and setting a threshold.

ORB: Oriented FAST and rotated BRIEF (ORB) [57] is scale, and rotation invariant but with limited affine changes, having a designated feature descriptor. It is a mixture of modified FAST (features from accelerated segment test) [58] and direction-normalized BRIEF (binary robust independent elementary features) [59] description methods. ORB detects corners in an image. FAST corners are detected in each layer of the scale pyramid and corners of detected points are evaluated using Harris corner score [57, 58] to filter out top-quality points. After feature detection, feature matching is performed using Hamming distance for binary feature descriptors and setting a threshold.

BRISK: Binary robust invariant scalable keypoints (BRISK) [60] is also scale and rotation invariant but with limited affine changes, having a designated feature descriptor. Like ORB, BRISK also detects corners in an image. It detects corners using AGAST (adaptive and generic accelerated segment test) [61] and filters them with FAST corner score while searching for maxima in the scale space pyramid. After feature detection, feature matching is performed using Hamming distance for binary feature descriptors and setting a threshold.

[1]KAZE is a Japanese word which means wind, the authors in [54] decided to use this term for their method.

[2]Blob is a group of connected pixels in an image that share some common property, for example, grayscale value.

[3]DoG is actually a band-pass filter which removes high-frequency components representing noise, and also some low-frequency components representing the homogeneous areas in the image.

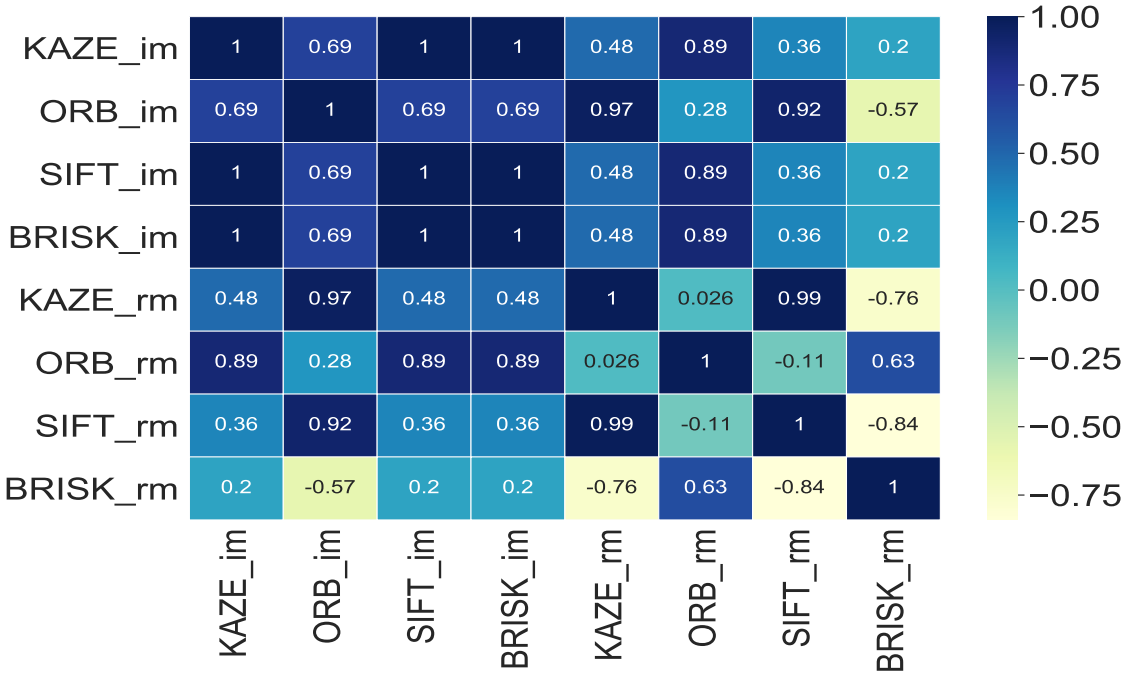


Figure 2.11: Illustration of correlation between inter-image and inter-radio map similarities using four different similarity measures.

As an illustration, we calculate four different similarity measures (KAZE, ORB, SIFT, and BRISK) between the original wireless environment and three different target wireless environments as shown in Figure 2.9. To this end, Figure 2.11 shows Pearson's correlation [20], which is a widely adopted metric for measuring the correlation, between the inter-image and the inter-radio map similarities. It can be seen that the correlation value is very poor for each similarity measure. This again signifies that these similarity measures are not effective for our TL problem of radio map estimation.

PSNR: The objective image quality metric, such as the peak signal-to-noise ratio (PSNR) [62,63] evaluates the quality of the image by exploiting the difference of corresponding pixel values between the original and the reconstructed image (in our case, the image of original and target wireless environment) and provides some measure of closeness between them. The higher the PSNR, the better the image quality. Suppose $f_{i,j}$ and $f'_{i,j}$ are the original and the reconstructed image, respectively. H and W are the height and width of the image, respectively. L is the dynamic range of allowable image pixel intensities (L = 255). The PSNR is defined as [62,63]:

$$\text{PSNR} = 10\log_{10}\left(\frac{L^2}{MSE}\right). \tag{2.4}$$

where the mean square error (MSE) is defined as:

$$\text{MSE}(f, f') = \frac{1}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}(f_{i,j} - f'_{i,j})^2. \tag{2.5}$$

SSIM: Structural similarity (SSIM) [64, 65] is a perception-based metric that considers image degradation as the perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking. The SSIM index is computed locally within a sliding window that moves pixel-by-pixel across the image, resulting in an SSIM map. The SSIM score of the entire image is then computed by pooling the SSIM map, e.g., by simply averaging the SSIM values across the image [65].

Suppose x and y are the local image patches taken from the same location of two images that are being compared (in our case, the image of the original and target wireless environment). SSIM measures the similarities of three elements of image patches: the luminances $l(x, y)$ (brightness values), the contrasts $c(x, y)$, and the structures $s(x, y)$. It is defined as [64]:

$$\begin{aligned} S(x,y) &= l(x,y).c(x,y).s(x,y) \\ &= \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right)\left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right)\left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right) \end{aligned} \tag{2.6}$$

where, μ_x, μ_y and σ_x, σ_y are the local sample means and standard deviations of x and y, respectively. σ_{xy} is the sample cross-correlation of x and y after removing their means. C_1, C_2, and C_3 are small positive constants that stabilize each term so that near-zero sample means, variances, or correlations do not lead to numerical instability.

Since these classical similarity measures can be applied to any type of image representing wireless environments, and they do not consider the radio environment characteristics and propagation properties, intuitively, these similarity measures are not applicable, in general, for TL of radio map models. This has been numerically verified in **Chapter 3**. This motivates us to design a data-driven similarity measure (DDS) (see **Section 3.6.2**) that learns the wireless environment characteristic from the data. For this, we design and then train a CNN regression model.

Data-driven Similarity Measure (DDS): Figure 2.12 shows the DDS based on a CNN with its input and output. The input of the CNN is images of distinct target wireless environments. The output of the CNN is a similarity score. This similarity score can be selected based on the application and it is discussed in more details in **Section 3.6.2** of **Chapter 3**.

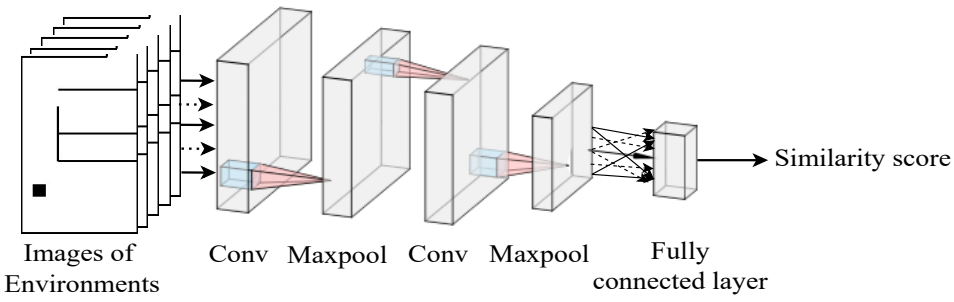


Figure 2.12: Illustration of the data-driven similarity measure.

2.7 Mixture of Experts

As discussed previously, it is possible to estimate the approximate `Rx` locations from the ToA (discussed in **Section 2.2**) features of radio signals using different types of algorithms, such as the trilateration localization [1] (discussed in **Section 2.3**). One can also directly use the ToA features to estimate radio maps. A mixture of experts (MoE) architecture considers these two methods in order to exploit the complementarity between them.

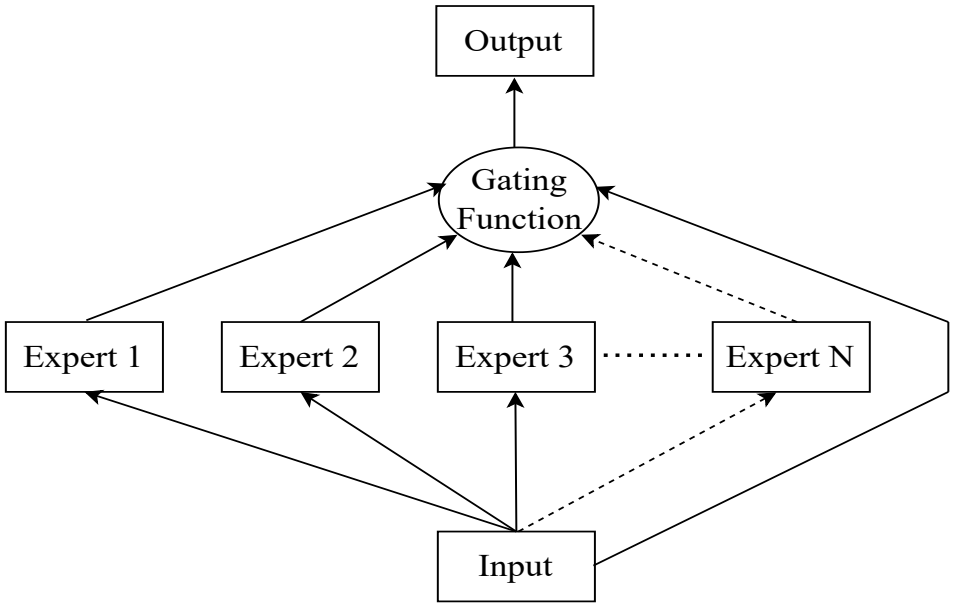


Figure 2.13: A simple illustration of MoE architecture.

The MoE is one of the popular ensemble methods to perform a machine learning (ML) task, such as classification or regression [66]. It mainly works on the divide-and-conquer [67] principle, in which the overall problem (in our case radio map estimation) is divided between different kinds of machine learning experts, such as DNN experts and supervised by a gating function [66]. In other words, different experts designed using different features are used as the input of MoE, as shown in Figure 2.13. The output of each expert is combined linearly or non-linearly using data-dependent weights through a gating function. The gating function can be either a weighted sum of each expert or it can be a DNN incorporating the output of each expert as an input feature. Furthermore, the gating function is trained in order to obtain the final output (in our case radio maps). The key advantages of MoEs are that, instead of assigning a set of fixed combinational weights to each expert, the gating function computes and assigns these weights dynamically from the inputs according to the local efficiency of each expert across space. As a result, the MoE has been shown to perform better in different ML tasks, such as in the classification of cluster structures [68].

In the context of our radio map estimation problem, MoE comprises two DNN-based experts. The first expert is the location-based expert, trained with the estimated locations of `Rxs`, and the second expert is the location-free expert, trained

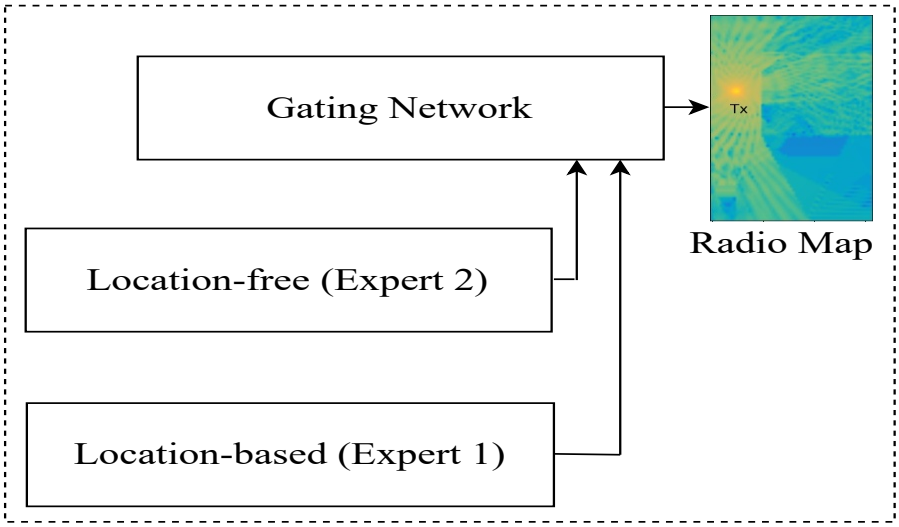


Figure 2.14: MoE architecture for radio map estimation.

with the ToA features, as shown in Figure 2.14. The gating network, which is also a DNN, combines these experts to obtain the final radio maps. It has been discussed in more details in **Chapter 5**.

2.8 Transfer Learning for Resource Allocation

Two optimization problems may have a certain similarity and this can be also exploited using TL. In general, if optimization problem 1 is similar to optimization problem 2 then a DNN trained to solve optimization problem 1 can be transferred and fine-tuned to solve optimization problem 2. For this to work, the two problems should be sufficiently similar.

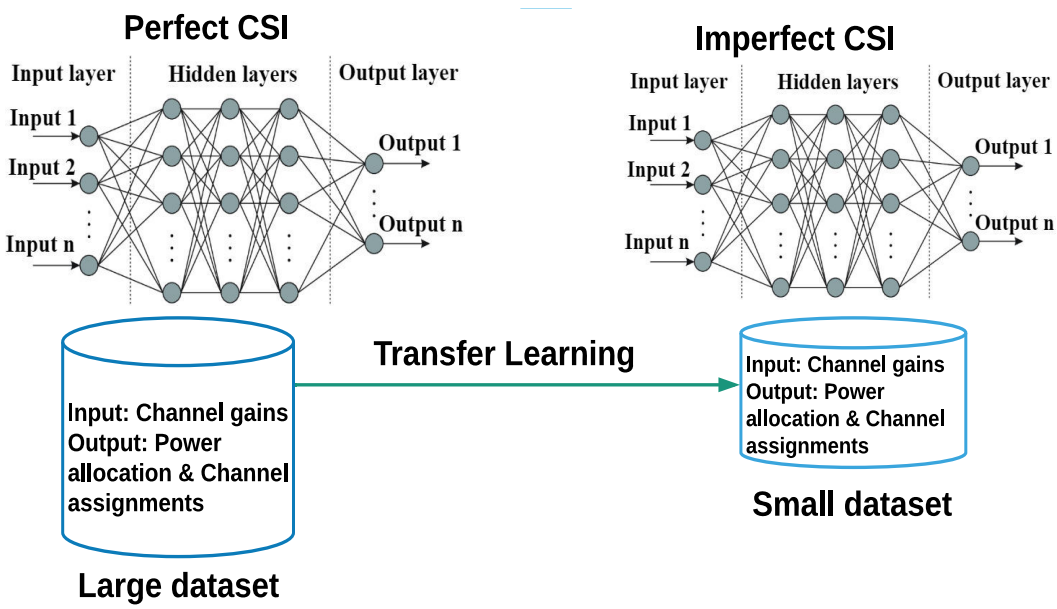


Figure 2.15: Illustration of the TL system model for resource allocation.

For example, in this dissertation, we investigate the TL between similar optimization problems for the resource allocation (RA) (joint channel assignment and power allocation) task among cellular users (CUs) and D2D pairs, as shown in Figure 2.15. In this case, we take the first RA optimization problem of perfect channel state information (CSI) and learn a DNN that imitates/approximates an optimal solver for this problem. We then transfer this trained DNN model to the second RA optimization problem of imperfect CSI which is somehow similar and then fine-tune it with the small amount of data obtained from the solver.

2.8.1 Similarity Measure for Resource Allocation

The similarity between two RA problems of perfect and imperfect CSI can be established using the outage probability. The higher the outage probability, the more similar the two RA problems are, leading to a fewer amount of sensor measurement data for fine-tuning the transferred model in the imperfect CSI scenario. Under specific assumptions, we prove that these two RA problems are equivalent for an outage probability of $\epsilon = \frac{1}{e}$. More details are presented in **Chapter 6**.

Chapter 3

Transfer Learning for Location-based Radio Map Estimation

This chapter summarizes Paper A [34] and Paper B [2]. The results obtained here reply **Q1**.

3.1 Motivation

Estimation of accurate radio maps is necessary for various applications of wireless communications, such as localization, network designing and planning, and resource allocation [36]. For estimating accurate radio maps, most existing methods require knowing the exact locations of receivers (`Rxs`). Moreover, these radio maps change for different wireless environments. As a result, a radio map model learned for a given wireless environment can not be directly deployed and used for other wireless environments. Learning a radio map model in a new wireless environment needs, in general, a sufficiently large amount of training data and is computationally demanding. To address these issues, we propose a transfer learning (TL)-based radio map estimation method that transfers and fine-tunes a deep neural network (DNN)-based radio map model learnt from an original wireless environment to other different (but sufficiently similar) wireless environments. Our proposed method recommends whether to perform TL or not and it also predicts the amount of training data needed to estimate radio maps in new wireless environments, when performing the TL operation, based on our designed data-driven similarity measure (DDS). In Paper B [2], we experimentally tested different classical similarity measures between two different wireless environments in order to investigate their applicabilities in the TL problem of radio map estimation. Moreover, we also investigate the robustness of the proposed scheme under noisy environments, that is, when there are errors in the locations of receivers `Rxs` (inaccurate location estimates), and noise in the power values. These two kinds of inaccuracies are prevalent in wireless networks due to the challenges in estimating accurate receiver locations because of the wireless multipath, and the measurement errors in the device, respectively. The main contributions of papers A and B can be summarized as follows:

- Design of an effective data-driven TL method that transfers and fine-tunes a DNN-based model for a radio map learned from a given source environment to another target environment.

- Design of a data-driven similarity measure that can recommend to perform the TL operation more accurately in a new target wireless environment, as compared to other widely used similarity measures.

- Prediction of the amount of training data needed to estimate the radio map in a different target environment, when performing the TL operation, based on our data-driven similarity measure.

- Analysis of the robustness of the proposed TL scheme under noisy environments when the locations of receivers are estimated and not perfectly known, and the corresponding power values are noisy. Our proposed TL scheme shows satisfactory performance (Paper B).

- Extensive testing of the proposed TL scheme using simulated data from the Remcom simulator [33]. Results demonstrate that our proposed scheme can perform effective TL in several varying environments while using a small amount of training data. Moreover, the proposed data-driven similarity measure can accurately recommend TL guaranteeing a satisfactory radio map estimation while reducing the number of necessary training samples.

3.2 Problem Formulation

The main questions we are targeting here are: (a) how to design a baseline model for a specific wireless environment to estimate the radio maps; (b) how to estimate the radio map corresponding to a new wireless environment with limited available sensor data (aggregate power values), by leveraging the information from the previously

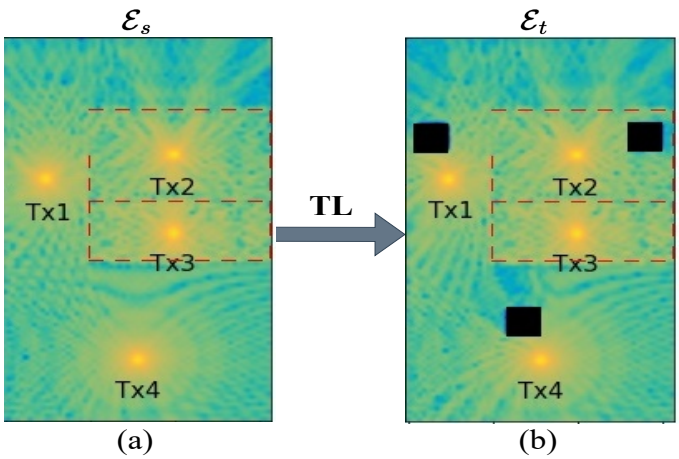


Figure 3.1: Illustration of radio map for: (a) original (source), and (b) target wireless environment with 4 `Txs` and 3 objects represented by 3 cubes from Paper B [2].

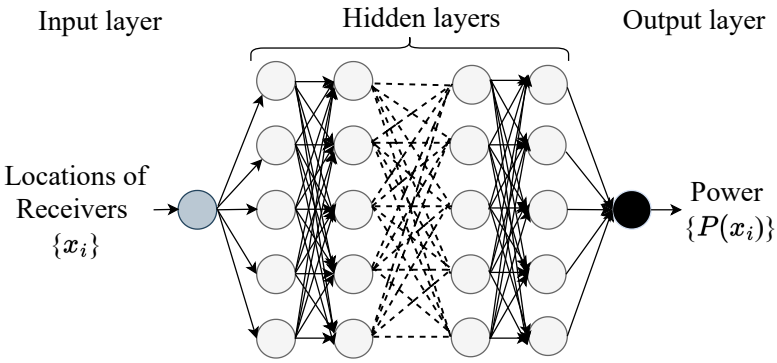


Figure 3.2: MapNet: Baseline DNN model.

learned model (baseline) of source wireless environment that has a certain similarity to the new wireless environment, which requires a certain similarity measure.

Let $\mathcal{E}_s$ and $\mathcal{E}_t$ denote the source and target wireless environment, respectively, as shown in Figure 3.1. Let us assume that only a small amount of aggregate power values are available in the target wireless environment $\mathcal{E}_t$, as compared to the number of samples in the source environment $\mathcal{E}_s$. The radio map consists of aggregate power measurements $P(x_i)$ corresponding to a set of N locations $\{x_i\}_{i=1}^{N}$, where i is the index of the receiver. To obtain the aggregate power at each `Rx` location, multiple signals from different transmitters `Txs` are added with each different phase.

Using a DNN (referred to as *MapNet*), as shown in Figure 3.2, we create a predictive function that predicts the power values at any locations in a radio map for a given wireless environment. More specifically, for the source environment $\mathcal{E}_s$, we first learn a predictive function $f_s(\cdot)$ that is approximated by a DNN with weight parameters $\theta_s^\star$. Note that $\theta_s^\star$ is the optimal value of the weight parameters in the source environment $\mathcal{E}_s$. Similarly, for the target environment $\mathcal{E}_t$, one can also learn the predictive function $f_t(\cdot)$ that is approximated by a DNN with weight parameters $\theta_t^\star$. Let us also assume that $\mathcal{E}_s$ and $\mathcal{E}_t$ have some similarity in terms of wireless propagation characteristics.

Instead of learning $f_t(\cdot)$ directly without exploiting any previous knowledge, we perform first TL by exploiting the already learned function $f_s(\cdot)$. This allows for improving the learning of the target predictive function $f_t(\cdot)$ by using a smaller amount of additional measurement samples, as compared to not exploiting the previous information.

In the source environment $\mathcal{E}_s$, we learn the optimal weight parameters $\theta_s^\star$ for the power value prediction using a DNN by minimizing a loss function:

$$\theta_s^\star = \arg\min_{\theta \in \Theta} [\text{Loss}(P_s(x_i), \hat{P}_s^\theta(x_i))] \tag{3.1}$$

where Θ is the space of parameters of θ, $P_s(x_i)$ and $\hat{P}_s^\theta(x_i) = f_s(x_i, \theta)$ are the actual and the predicted power values at i^{th} location, respectively, in the source environment $\mathcal{E}_s$. We denote the loss function as $\text{Loss}(P_s(x_i), f_s(x_i, \theta))$, which is chosen in this work to be the Mean Square Error (MSE), defined as, $\text{MSE} = \frac{\sum_{i=1}^{N_s} [P(x_i) - \hat{P}(x_i)]^2}{N_s}$,

where N_s is number of receivers `Rxs`. Note that the number of receivers may also be different in each environment.

Then, we can learn a DNN model for the target environment $\mathcal{E}_t$ by first transferring and initializing (TL operation) the DNN model parameters as $\theta_t^{[0]} = \theta_s^\star$, and then solving with the usual training iterations to minimise the same loss function in the target environment $\mathcal{E}_t$. After performing TL, the training will result in a "fine-tuning", which should require a small number of additional data samples if the environments $\mathcal{E}_s$ and $\mathcal{E}_t$ are sufficiently similar.

$$\theta_t^\star = \arg\min_{\theta \in \Theta} [\text{Loss}(P_t(x_i), \hat{P}_t^\theta(x_i))] \tag{3.2}$$

where $P_t(x_i)$ and $\hat{P}_t^\theta(x_i) = f_t(x_i, \theta)$ are the actual and the predicted power values at i^{th} location of the receiver, respectively, at the target environment $\mathcal{E}_t$. If the TL is effective, it is expected that the number of samples $\{P_t(x_i)\}$ to estimate an accurate radio map will be smaller as compared to the number of samples that are used to train the source environment $\mathcal{E}_s$, and smaller than the number of samples that would need to be used at the target environment $\mathcal{E}_t$ without TL. Note that the loss function $\text{Loss}(\cdot, \cdot)$ is typically chosen to be the same for both source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments, namely MSE.

3.3 Radio Map Estimation Model Using TL

The overall architecture of the radio map estimation model is shown in Figure 3.3 and Figure 3.4. It mainly consists of two phases of operations: (i) training shown in Figure 3.3, and (ii) execution shown in Figure 3.4. The training itself has two stages: (i) the development of the baseline DNN model in the source wireless environment $\mathcal{E}_s$, followed by the possibility of transferring it to the target wireless environment $\mathcal{E}_t$ by first transferring the baseline model and then fine-tuning, and (ii) the establishment of a data-driven similarity measure between the source and the target wireless

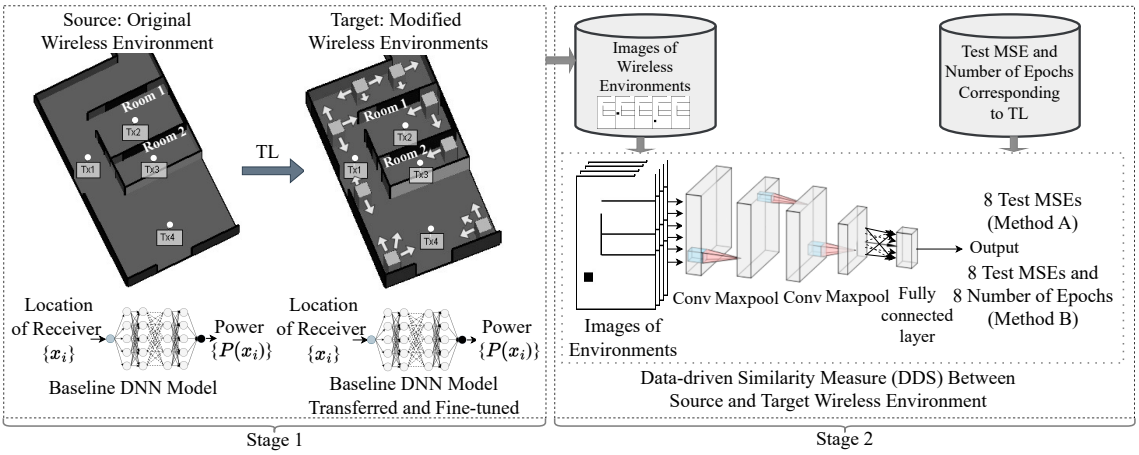


Figure 3.3: Data-driven transfer learning based radio map estimation model (training phase).

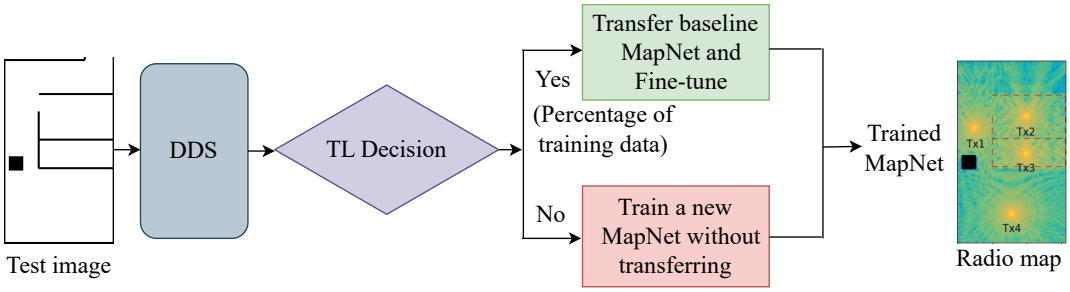


Figure 3.4: Data-driven transfer learning based radio map estimation model (execution phase).

environment. During execution, the image of a test wireless environment is given as an input to the DDS. The DDS predicts different MSE values and number of training epochs for the TL operation at different data splits (number of training samples). Next, in the TL decision block, these values are compared with the set MSE and epochs thresholds to decide further whether to perform TL operation with a specific data split. If the thresholds are satisfied, then the TL can be performed with that specific data split, else, TL will not be performed and *MapNet* needs to be trained from the very beginning.

3.4 TL System Design

The TL system design proposed for the radio map estimation from the original wireless environment to the target environment is shown in Stage 1 of Figure 3.3. The original indoor wireless environment is a single floor comprising two rooms. The target indoor wireless environments are created by incorporating several objects, represented by cubes, inside the original wireless environment. Each cube can represent any kind of object, such as furniture, or TV. These cubes can cause different wireless propagation phenomena, such as reflection, diffraction, and scattering similar to the effects caused by the real-world objects.

3.5 Transfer Learning Approach

The baseline DNN *MapNet* is learnt from the power values $\{P(x_i)\}$ observed at multiple locations of `Rxs` $\{x_i\}$ under the source wireless environment $\mathcal{E}_s$ (see Figure 3.2).

A high-level description of the flow of the baseline DNN *MapNet* in the form of pseudo-code is presented in **Algorithm 1**.

Next, given the baseline model *MapNet* shown in Figure 3.2, we first decide whether to perform the TL operation based on the similarity measure. We, then, transfer the *MapNet* to the target environment $\mathcal{E}_t$, followed up by fine-tuning using some additional amount of training data only 5% to 40% from the target environment. In our work, we consider a set of possible target environments. After training and fine-tuning the transferred baseline model *MapNet* for each new target environ-

Algorithm 1: Baseline DNN Model *MapNet*

Input: $D := \{x_i, P(x_i)\}$: dataset of n locations of `Rxs` x_i, and power values $P(x_i)$
B: Batch size, α: Learning rate, N_{EP}: Number of epochs
Output: θ: Trained DNN model parameters, MSE: MSE of trained DNN based on test data
Data-Splitting: Splitting data into training D_{train} and test D_{test}

1: `Training stage`
2: Randomly initialize the network parameters
3: **for** k = 1,...,N_{EP} **do**
4: **for** j = 1,...,N_B = ceil($|D_{\text{train}}|/B$) **do**
5: Randomly select B training samples from training data D_{train} as the training batch
6: Update DNN parameters θ with learning rate α to minimize $Loss = \text{MSE} = \frac{\sum [P(x_i) - \hat{P}^{\theta}(x_i)]^2}{|B|}$
7: **end for**
8: **end for**
9: `Testing stage`
10: Initialize MSE: MSE $\leftarrow 0$
11: Use test data D_{test}
12: Predict power values $\hat{P}(x_i)$ on given locations of `Rxs` $\{x_i\}$
13: Calculate MSE as MSE $= \frac{\sum [P(x_i) - \hat{P}^{\theta}(x_i)]^2}{|D_{\text{test}}|}$
14: Save trained DNN model parameters θ.

ment $\mathcal{E}_t$, we store the test MSE and the number of training epochs, which are used further to establish our data-driven similarity between the source $\mathcal{E}_s$ and the target $\mathcal{E}_t$ wireless environments, as we explain in **Section 3.6**.

A high-level description of the flow of the TL method in the form of pseudo-code is presented in **Algorithm 2**.

3.6 Similarity Measure

A fundamental element in TL is the notion of similarity between the source $\mathcal{E}_s$ and the target $\mathcal{E}_t$ wireless environments because similarity affects directly the effectiveness of the TL operation. The more similar the source $\mathcal{E}_s$ and the target $\mathcal{E}_t$ wireless environments are, the less amount of training data is expected to be required for the TL operation in the corresponding target $\mathcal{E}_t$ environment.

3.6.1 Classical Similarity Measures

To establish the similarity between the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ wireless environments, we first consider and analyse one of the widely used similarity measures used in TL, such as the Wasserstein distance (WD) [23]. WD computes the distance between

Algorithm 2: TL Method

Input: $\mathcal{E}_s$: Source environment, $\mathcal{E}_t$: Target environment, $MapNet$: baseline DNN model $\theta_s^\star$, α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}: Number of epochs

Output: $MapNet$: DNN model $\theta_t^\star$ for target environment $\mathcal{E}_t$, MSE($\mathcal{E}_t$): MSE of transferred model based on the test data, $N_{EP}(\mathcal{E}_t)$: Number of training epochs for the transferred model

Data-Splitting: Splitting data into training D^t_{train} and test D^t_{test} with splitting ratio r

1: `Fine-Tuning and Testing stage`
2: Initialize MSE($\mathcal{E}_t$): MSE($\mathcal{E}_t$) $\leftarrow 0$
3: Initialize $N_{EP}(\mathcal{E}_t)$: $N_{EP}(\mathcal{E}_t) \leftarrow 0$
4: `Fine-Tuning stage`
5: **for** k = 1,...,N_{EP} **do**
6: **for** j = 1,...,N_B = ceil($|D^t_{\text{train}}|/B$) **do**
7: Randomly select B training samples from training data D^t_{train} as the training batch
8: Update DNN parameters θ with learning rate α to minimize $Loss = \text{MSE} = \frac{\sum[P(x_i)-\hat{P}^\theta(x_i)]^2}{|B|}$
9: **end for**
10: Check early stopping criterion and stop when satisfied
11: **end for**
12: Save number of training epochs N_{EP}
13: `Testing stage`
14: Predict power values $\hat{P}(x_i)$ on given locations of `Rxs` $\{x_i\}$
15: Calculate MSE($\mathcal{E}_t$) $= \frac{\sum[P(x_i)-\hat{P}^\theta(x_i)]^2}{|D^t_{\text{test}}|}$

two distributions, which in our case, correspond to the two images of the two floor plans of these environments. In our case, we compute WD between two images of different wireless environments. The mathematical formulation of the WD is presented in **Section 2.6** of **Chapter 2**.

There exist different similarity measures, such as KAZE [54], SIFT [56], ORB [57], BRISK [60], PSNR [64], SSIM [64], in addition to the Wasserstein distance, which we also consider. A brief description of all these similarity measures is presented in **Section 2.6** of **Chapter 2**.

Notice that since these classical similarity measures can be applied to any type of image representing source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments, and they do not consider the radio environment characteristics and propagation properties, it is expected that, these similarity measures are not suitable, in general, for TL of radio map models. In fact, we show in both Paper A [34] and Paper B [2], respectively, that these similarity measures, including WD, do not perform well in the context of radio map estimation, as shown in Figure 3.5. This motivates us to design our data-driven similarity measure (DDS) (see Figure 3.6) which incorporates and learns the wireless

Figure 3.5: Correlation analysis between different types of similarity measures.

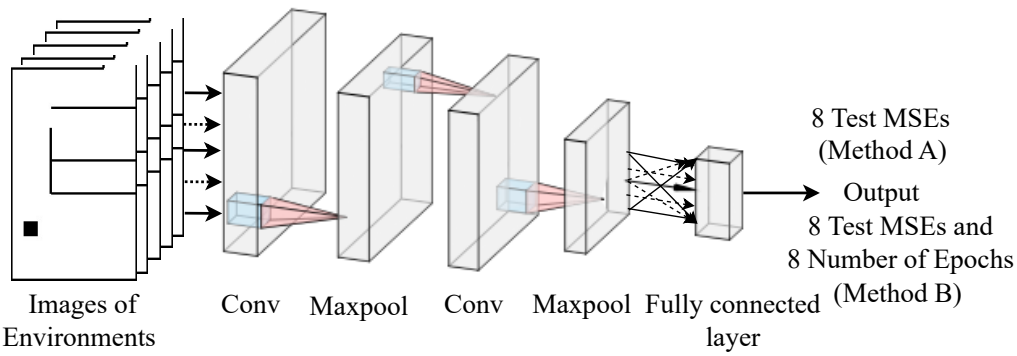


Figure 3.6: CNN-based neural structure for the similarity measure.

environment characteristics from the data. This is done by incorporating in the loss function, that is optimized in the DDS, the MSE (or number of training epochs) corresponding to the radio map obtained after using TL from the baseline *MapNet*. Thus, our DDS is coupled with the baseline *MapNet*.

3.6.2 Data-driven Similarity Measure

Our DDS incorporates explicitly the TL-operation effect in terms of the MSE and/or the number of training epochs that are required to achieve a certain accuracy (MSE) of the radio map in the target environment $\mathcal{E}_t$, along with the floor plans images of the environments. To this end, we train a convolutional neural network (CNN) [50] regression model under two different loss functions. Loss function A, given by $L_A = |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t|^2$, is the squared difference between the actual test MSE obtained previously after performing TL, denoted by $\text{MSE}(\mathcal{E}_t)$, and the predicted test MSE, denoted by $\widehat{\text{MSE}}_t$, which is obtained when performing TL. Loss function B, given by

$L_B = ||[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t, N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]||^2$, contains both the test MSE and the number $N_{EP}(\mathcal{E})$ of training epochs, the latter one representing the training time to perform TL. Similarly, $N_{EP}(\mathcal{E}_t)$, and $\widehat{N_{EP_t}}$ are the actual and predicted TL number of training epochs, respectively. Loss function B measures the quality of the image in terms of MSE as well the training time in terms of the number of training epochs needed to perform TL.

Method A: This method comprises of a CNN (see Figure 3.6) having the images of distinct target environments $\mathcal{E}_t$ as input, and the test MSE, i.e., $MSE(\mathcal{E}_t)$ obtained after performing TL, as output. It uses the loss function L_A, and the corresponding learned CNN model is referred to as $SimNet_A$. Each image is assumed to be a three-level image in which free space is white (gray value 255), walls are assumed 128, and objects (e.g., cubes) are black (gray value 0). These differences in intensity levels are useful to differentiate different elements.

Method B: This method comprises of a CNN with the same input, however, the output is now the combination of the test MSE, i.e., $MSE(\mathcal{E}_t)$ obtained after performing TL and the number $N_{EP}(\mathcal{E}_t)$ of training epochs required for each target environment $\mathcal{E}_t$ when *MapNet* is transferred and re-trained. It uses the loss function L_B and the corresponding learned CNN model is referred to as $SimNet_B$.

For any similarity measure, to obtain a similarity decision on whether TL is beneficial (in our case, small $\text{MSE}(\mathcal{E}_t)$, and small $N_{EP}(\mathcal{E}_t)$) or not between two environments, we need to set a threshold. In the case of our DDS, under Method A, we can set a threshold for the test MSE, which we denote as MSE_{thr} and compare it with the predicted test MSE, i.e., $\widehat{MSE}\mathcal{E}_t$, obtained after TL. Similarly, under Method B, we can also set a threshold for both the predicted test MSE obtained after TL, i.e., MSE_{thr} and the number of training epochs, i.e, $N_{EP_{thr}}$ and compare these thresholds with the corresponding predicted values after TL. If the predicted values are less than these thresholds, then two radio maps are recognized as similar and it is expected that TL will perform effectively, thus, it is decided to perform TL, otherwise, we assume that TL is not effective. Notice that the choices of threshold values (MSE_{thr} and $N_{EP_{thr}}$) provide additional degrees of freedom on the proposed model. For Method A, a very high threshold MSE_{thr} will result in a less accurate radio map estimation. However, if we choose a very low threshold, then it will result in a better radio map estimation, but a lower recommendation rate[1]. Similarly, for Method B, higher threshold values MSE_{thr} and $N_{EP_{thr}}$ will result in both estimating a poorer radio map and having a longer convergence time. However, if we choose very low threshold values, then it will result in a better radio map estimation and a shorter convergence time, but the recommendation rate will be lower. Moreover, it should be noticed that the choice of these thresholds will also depend on the application scenario, that is, the complexity of the environment patterns. To select the value of these thresholds, we have to trade-off between the radio map quality, convergence time (Method B), and the recommendation rate of the TL. Figure 3.7 illustrates the similarity decision taken for a new test wireless environment under

[1]The recommendation rate of TL is calculated as the number of environments for which the TL operation is recommended divided by the total number of environments.

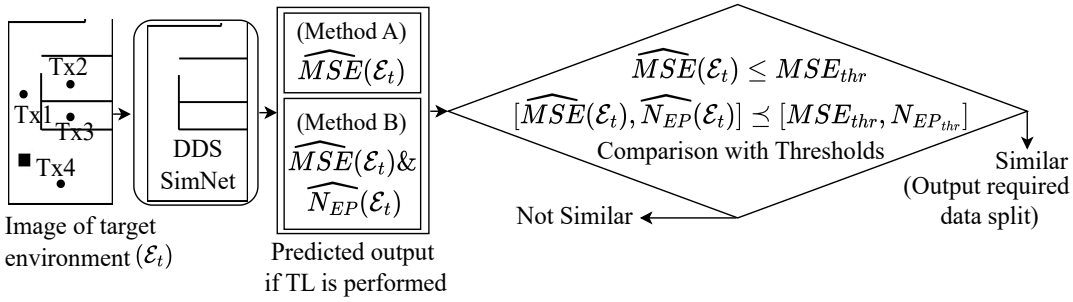


Figure 3.7: Data-driven similarity decision between the target $\mathcal{E}_t$ and source $\mathcal{E}_s$ environments.[2]

each method. The DDS takes an image of a test environment as an input. It then predicts different MSE values and the number of training epochs for the TL operation at different data splits (number of training samples). These values are compared with the set MSE and epochs thresholds. If the thresholds are satisfied, then the given test environment is considered to be similar and it outputs the specific data split.

A high-level description of DDS in the form of pseudo-code is presented in **Algorithm 3**.

3.7 Data Generation

We use the standard high-accuracy ray-tracing X3D ray model [41], computed using the software Remcom [33] to obtain the power measurements that are sufficiently representative of the real indoor wireless environments.

In Paper A [34], we place one `Tx` at one fixed location x^t in the original environment, and then multiple `Rxs` at uniformly spaced locations (x_i^r) with a 15 cm spacing, horizontally and vertically. This results in a total of 6678 `Rx` locations and their corresponding power values $\{P(x_i^r)\}$, where i is the index of the receiver. Next, to create different indoor wireless environments, we incorporate one object, represented by a solid cube (made of metal), within the original wireless environment.

Similarly, in Paper B [2], we place four `Txs` in the original indoor wireless environment at four different fixed locations and then multiple `Rxs` at uniformly spaced locations x_i^r with a 15 cm spacing, horizontally and vertically. Next, to create different indoor wireless environments, we place and denote the inclusion of one cube within the original wireless environment as *Type I* $\mathcal{E}$, two cubes as *Type II* $\mathcal{E}$, three cubes as *Type III* $\mathcal{E}$, four cubes as *Type IV* $\mathcal{E}$, and eight cubes as *Type V* $\mathcal{E}$. The inclusion of an increasing number of cubes makes the target wireless environments $\mathcal{E}_t$ more complex, providing more complex power radio maps to investigate the suitability of our proposed TL method in estimating the respective radio maps. The summary of different types of indoor wireless environments with the respective number of environments is presented in Table 3.1 and shown in Figure 3.8.

[2]The symbol $\preceq$ indicates component-wise less than or equal operation.

Algorithm 3: Data-driven Similarity Measure *SimNet*

Input: $D := \{\text{Im}_t, \text{MSE}_t, N_{EP_t}(\text{only for Method B})\}$: dataset comprising three-level images of target environments, MSE after TL for 8 different data splits (5% to 40%) (**Algorithm 2**), Number of training epochs required after TL for 8 different data splits (5% to 40%) (**Algorithm 2**), for each environment, respectively. MSE_{thr}: MSE threshold, $N_{EP_{thr}}$: Threshold for number of epochs, α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}^{cnn}: Training epochs of *SimNet*

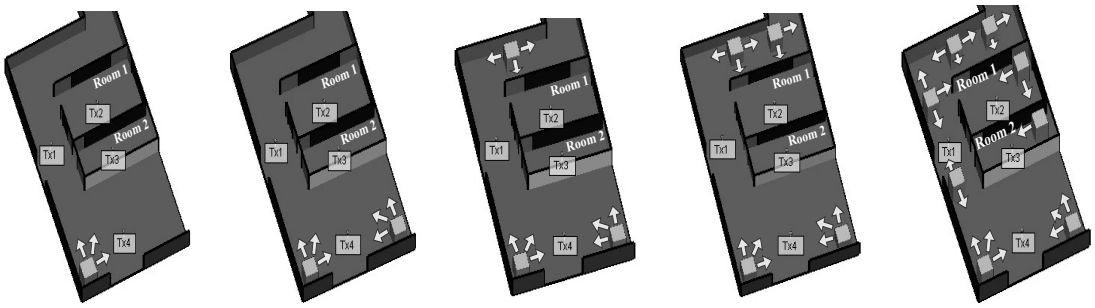
Output: $SimNet(\theta_{sim})$, $\widehat{\text{MSE}_t}$, $\widehat{N_{EPt}}$ are the predicted MSE and number of training epochs, respectively, for 8 different data splits (5% to 40%), TL decision: Yes/No, the required data split

Data-Splitting: Splitting data into training D_{train} and test D_{test} with splitting ratio r

1: `Training stage`
2: Use data from D_{train}
3: Normalize image values Im_t, MSE_t and N_{EP_t} to $[0, 1]$
4: Randomly initialize θ_{sim}
5: **for** k = 1,...,N_{EP}^{cnn} **do**
6: **for** j = 1,...,$N_B = \text{ceil}(|D_{\text{train}}|/B)$ **do**
7: Randomly select B training samples from D_{train} as the training batch
8: Update θ_{sim} with learning rate α to minimize *loss* given by the

$$\text{MSE}_A^{cnn} = L_A = |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}_t}|^2 \quad \text{(for Method A)}$$
$$\text{MSE}_B^{cnn} = L_B = |[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}_t}, N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]|^2 \quad \text{(for Method B)} \tag{3.3}$$

9: **end for**
10: **end for**
11: `Testing stage`
12: Initialize MSE^{cnn}: $\text{MSE}^{cnn} \leftarrow 0$
13: Use test data from D_{test}
14: Predict $\widehat{\text{MSE}_t}$ and $\widehat{N_{EPt}}$ for 8 different data splits (5% to 40%) on a given image Im_t
15: `TL decision`
16: **for** q = 1,...,$|D|$ **do**
17: **if** $\widehat{\text{MSE}_t} \leq \text{MSE}_{thr}$(for Method A) **then**
18: TL decision $\leftarrow$ Yes and output the corresponding data split, **else**, TL decision $\leftarrow$ No
19: **if** $\widehat{\text{MSE}_t} \leq \text{MSE}_{thr}$ & $\widehat{N}_{EP_t} \leq N_{EP_{thr}}$(for Method B) **then**
20: TL decision $\leftarrow$ Yes and output the corresponding data split, **else**, TL decision $\leftarrow$ No
21: **end if**
22: **end if**
23: **end for**



(a) (b) (c) (d) (e)

Figure 3.8: Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes), (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), and (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes)

Table 3.1: Summary of wireless environment $\mathcal{E}$.

Type of environment $\mathcal{E}$	Description	Number of environment $\mathcal{E}$
Type I	`4 Txs`, 1 Cube	250
Type II	`4 Txs`, 2 Cubes	252
Type III	`4 Txs`, 3 Cubes	248
Type IV	`4 Txs`, 4 Cubes	254
Type V	`4 Txs`, 8 Cubes	42
Total number of environments		1046

Furthermore, for each change in the location of the object, a new indoor wireless environment is created and simulated to obtain the power values for that particular environment. A total of 250 (with one `Tx`) and 1046 (with 4 `Txs`) different indoor wireless environments are created in Paper A [34], and Paper B [2], respectively. The image of each environment is saved as a 160 x 275-pixel image which is used to investigate the suitability of different similarity measures as discussed in **Section 3.6**, including our designed similarity measure. The different parameters used in Remcom for data generation are provided in both Paper A [34] and Paper B [2], respectively.

3.8 Performance Evaluation

We evaluate the performance of the proposed scheme based on the overall model accuracy (for Paper A [34] and Paper B [2]) and noise robustness (for only Paper B [2]).

3.8.1 Model Accuracy

We calculate the transferability, F1-score, and accuracy [69]. The transferability represents how good the considered model is in recommending a correct TL for a new target environment $\mathcal{E}_t$ (recommending a correct TL means that the actual TL operation is successful). A higher value of transferability means a higher chance of

Table 3.2: Transferability, F1-score, and Accuracy for Type I and all types of environments grouped together.

	(a)		(b)	
	Type I		All Types Grouped Together	
Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	237	226	828	779
TN	0	8	75	96
FP	13	10	103	132
FN	0	6	40	39
Transferability	1.0	0.974	0.953	0.952
F1-score	0.973	0.965	0.920	0.901
Accuracy	0.948	0.936	0.863	0.836

correctly recommending TL when TL is possible. The F1-score is a measure of the test accuracy of the model, normalized to be between 0 and 1. The closer it is to 1, the better the model is. Accuracy measures the degree of veracity of the model. All these measures are defined in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) as [69]:

$$\text{Transferability} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \ \text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{3.4}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{3.5}$$

where TP means that the source $\mathcal{E}_s$ and the target $\mathcal{E}_t$ environment are sufficiently similar $\text{MSE}(\mathcal{E}_t) \leq MSE_{thr}$ (Method A) or $[\text{MSE}(\mathcal{E}_t), \text{N}_{EP}(\mathcal{E}_t)] \preceq [MSE_{thr}, N_{EP_{thr}}]$ (Method B) and the decision to transfer is taken $\widehat{\text{MSE}}_t \leq \text{MSE}_{thr}$ (Method A) or $[\widehat{\text{MSE}}_t, \widehat{N}_{EP_t}] \preceq [\text{MSE}_{thr}, N_{EP_{thr}}]$ (Method B). TN means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are not sufficiently similar and the decision of not to transfer is taken. FP means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are not sufficiently similar but the decision to transfer is taken. FN means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are sufficiently similar but the decision of not to transfer is taken. For better accuracy, FP and FN should be small.

Table 3.2 presents the transferability, F1-score, and accuracy of the proposed model (for Paper B [2]). It can be noticed from Table 3.2 (a) that all performance measures for the Type I environments under Method A are slightly better than in Method B. The model accuracy under Method A is 94.8%, indicating a high accuracy. On the other hand, the model accuracy under Method B is 93.6%. Similarly, it can be noticed from Table 3.2 (b) that when all types of environments are grouped together, then all performance measures under Method A are also better than in Method B. The overall accuracy under Method A is 86.3% and the overall accuracy under Method B is 83.6%. This suggests that our proposed scheme can be used effectively for estimating radio maps in new wireless environments.

3.8.2 Noise Robustness

To investigate the robustness of the proposed scheme (for Paper B [2]) towards the presence of noise, we investigate two cases of noise: (a) error in the `Rxs` locations (inaccurate location estimations), and (b) noise in the power values. To this end, for case (a), we consider a practical scenario where the `Rxs` locations are estimated from the time of arrival (ToA) features of the radio signals using the *trilateration localization algorithm*[3] [1] (see **Section 2.3**). ToA is the time at which the radio signals from each `Tx` arrive at each of the `Rxs`. For case (b), in order to reflect the inexact power measurements, we add Gaussian noise of zero mean and a variance of 0.5 dBm to the power values in each environment. This procedure is followed for each target environment $\mathcal{E}_t$ of each type. We then transfer the baseline DNN model *MapNet* to each target environment $\mathcal{E}_t$ and fine-tune it to obtain the test MSE, i.e.,

[3]Notice that the particular choice of the algorithm is not important for the evaluation of our TL performance with respect to error locations.

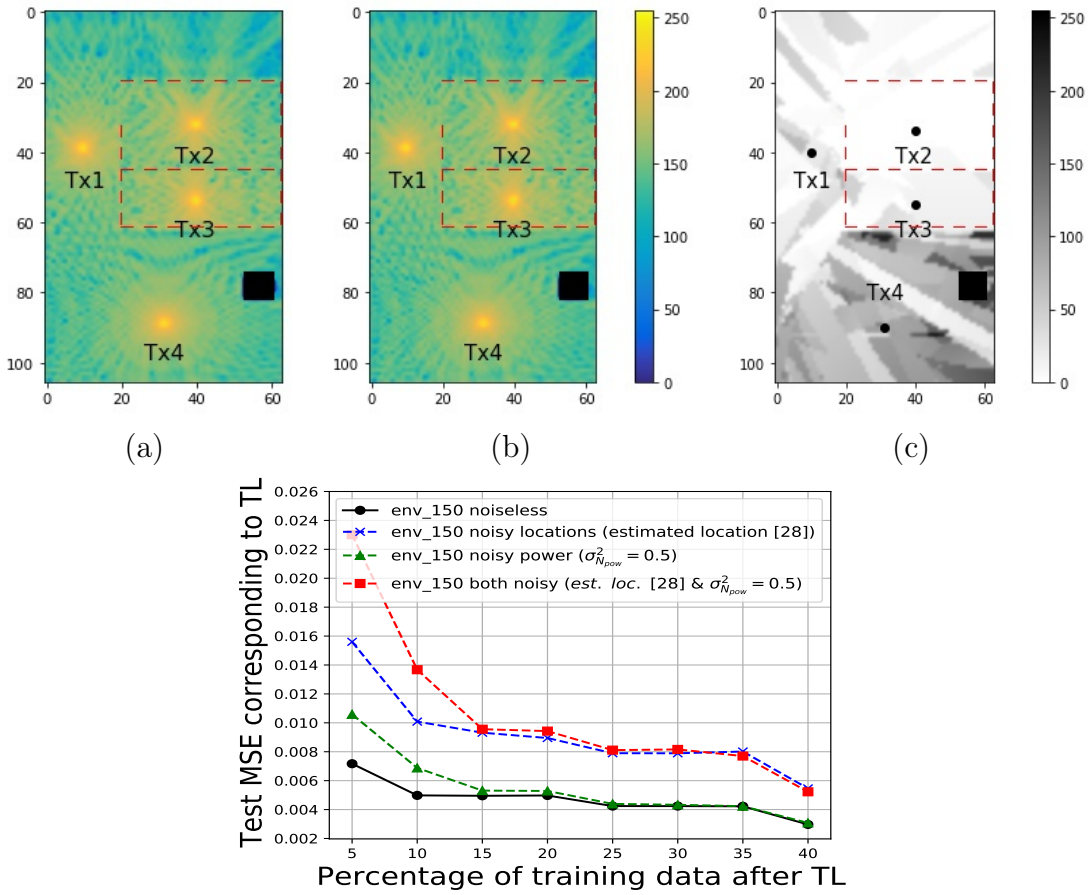


Figure 3.9: (a) Radio map for a wireless environment of Type I (noiseless), (b) Radio map of the same environment as in (a) after adding noise to power values, (c) location error map between the true and estimated `Rxs` locations for the same environment as in (a), and (d) plot between test MSE corresponding to TL and the percentage of training data after TL for the noiseless and the noisy case corresponding to the same environment as in (a).

MSE($\mathcal{E}_t$), and the number $N_{EP}(\mathcal{E}_t)$ of training epochs corresponding to TL for each target environment $\mathcal{E}_t$.

Figure 3.9(a) shows one of the wireless environments from the Type I environment (noiseless), Figure 3.9(b) shows noisy radio map after adding noise to the power values, Figure 3.9(c) shows the location error map between the true and estimated `Rxs` locations, and Figure 3.9(d) shows the plot between the test MSE corresponding to TL and the necessary percentage of training data after the TL operation, for the noiseless and the noisy environments.

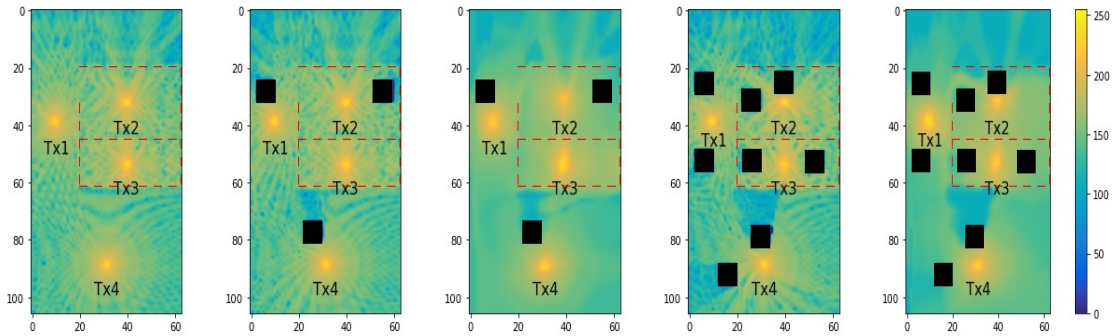
It can be seen from Figure 3.9(d) that as the percentage of training data increases, the test MSE corresponding to TL, i.e., MSE($\mathcal{E}_t$), decreases. For the noiseless environment, the MSE($\mathcal{E}_t$) is smaller (shown by the solid black line). However, when we have noisy power values, then there is a significant increase in the MSE($\mathcal{E}_t$) (shown by the dashed green line) when using a smaller amount of training data (upto 25%) and thereafter, it overlaps with the noiseless test MSE. This signifies that our proposed scheme shows some robustness against noisy power values. On the contrary, when estimated `Rx` inexact locations are used, then the MSE($\mathcal{E}_t$) increases significantly (shown by the dashed blue line) and the increment is slightly higher, as compared to the case of having noise in the power values. This means that the errors in the estimated `Rx` locations affect more than the noise in the power values. This signifies that our proposed scheme is less robust against the errors in location estimations as compared to the noisy power values. Similarly, when we have both estimated `Rx` locations and noisy power values, then the overall increment in the MSE($\mathcal{E}_t$) is comparatively higher (shown by the dashed red line) than in the noiseless environment. This means that our proposed scheme is less robust against having a combination of errors in location estimations and noisy power. In addition, it can also be seen that for noisy scenarios, more data is needed to compensate for this noise, as expected. For instance, to achieve an MSE($\mathcal{E}_t$) = 0.005, in the noiseless case, it requires only 10% of the training data. However, to achieve the same amount of MSE($\mathcal{E}_t$), our model trained with estimated `Rx` locations and no noise in power, requires 40% of the training data. This motivates the design of TL with location-free methods, as we explain in **Chapter 4**.

3.9 Visualization of Radio Maps

We visualize different radio maps (for Paper B [2]) which are obtained before and after applying TL operation, as shown in Figure 3.10. The cubes and the walls of the rooms are represented in black and dashed red colours, respectively. The differences in the radio maps can be easily visualized.

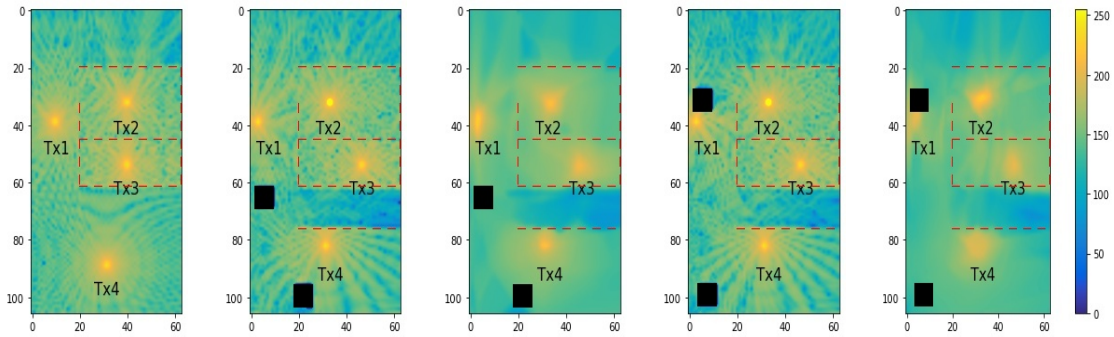
3.10 Performance of the TL Model with Additional Complex Changes in Wireless Environments

Additionally (for Paper B [2]), we generated 130 different wireless environments by making the following changes: (a) changing the carrier frequency of waveform



(a) (b) (c) (d) (e)

Figure 3.10: Radio map for (a) original wireless environment with `4 Txs`, (b) a wireless environment from Type III, (c) same wireless environment as in (b) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0038), (d) a wireless environment from Type V, and (e) same wireless environment as in (d) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0048).



(a) (b) (c) (d) (e)

Figure 3.11: Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) same environment as in (b) after performing TL with 35% training data (MSE($\mathcal{E}_t$) = 0.005), (d) another environment with additional wall and changed `Tx` locations, and (e) same environment as in (d) after performing TL with 40% training data (MSE($\mathcal{E}_t$) = 0.0002).

from 900 to 950 MHz, (b) changing the location of each `Tx` by 1 meter, (c) adding a wall below to second room, and (d) adding two objects in the floor plan and changing its locations horizontally, and vertically with a spacing of 15 cm. This experiment was successful with a TL-recommendation rate of 100% and an overall model accuracy of 84.4%. The radio maps of two of these wireless environments are shown in Figure 3.11. The differences in these radio maps can be easily visualized.

3.11 Summary

- We design an effective data-driven TL method that transfers and fine-tunes a DNN-based radio map model learned from an original indoor wireless environment to other different indoor wireless environments.

- Our data-driven similarity measure performs efficiently in finding the similarity between the source and the target wireless environments. It also predicts the amount of training data needed to estimate radio maps in different target wireless environments when performing the TL operation.

- The traditional similarity measures, including the widely used Wasserstein distance, are not able to provide accurate similarity between the source and the target wireless environment that considers the wireless propagation characteristics. As a result, these similarity measures are not suitable for our TL problem in the context of radio map estimation.

- With different types of wireless environments tested, our proposed TL scheme performs satisfactorily with high model accuracy and saves a substantial amount of sensor measurement data, even in the presence of complex changes in the wireless environments, such as the change in `Tx` locations, carrier frequency, additional wall, etc.

- The errors in the estimated `Rx` locations affect the model accuracy more than the noise in the power values. This signifies that our proposed TL scheme is less robust against errors in location estimations as compared to errors in power values. In addition, for noisy scenarios, more training data is needed to compensate for this noise. This motivates the investigation of methods that are location-free.

Chapter 4

Transfer Learning for Location-free Radio Map Estimation

This chapter summarizes Paper C [8]. The results obtained here reply **Q2**.

4.1 Motivation

Obtaining accurate locations, especially, in scenarios having a high degree of wireless multi-path, is difficult. Alternatively, one can employ location-free features of radio signals, such as time of arrival (ToA), time difference of arrival (TDoA), and direction of arrival (DoA), to estimate radio maps. ToA, which is the focus of our work, is the time at which the radio signals arrive at the receiver `Rx` from the transmitter `Tx`, and it can be estimated through different methods (see **Section 2.2** in **Chapter 2**). To this end, we propose a transfer learning (TL) method using ToA features to estimate radio maps for indoor wireless communications. Moreover, we also compare the proposed TL method with (a) the TL method of **Chapter 3**, where the exact locations of `Rxs` are used, and (b) the TL method where both the ToA features and the exact locations of `Rxs` are used.

4.2 TL System Design

Similarly to **Chapter 3**, we propose a TL system for estimating radio maps from the source (original) indoor wireless environment to the target indoor wireless environments.

4.3 Baseline Model and Transfer Learning Approach

To estimate radio maps, we first develop a fully-connected deep neural network (DNN)-based radio map model in the original wireless environment, as shown in Figure 4.1. In this case, the DNN is designed to map the ToA $\{\tau_i^r\}$, corresponding to each `Rx` location $\{x_i^r\}$, to the power values $\{P(\tau_i^r)\}$ obtained at each `Rx` location $\{x_i^r\}$ under the original wireless environment (referred to as Method 2). We

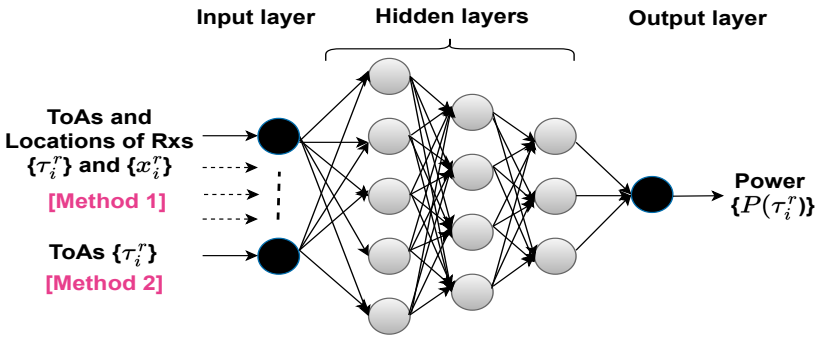


Figure 4.1: DNN model for radio map estimation.

use the first two ToA values $\{\tau_i^r\}$. The target wireless environment then exploits this baseline radio map model and fine-tunes it with additional data available from the target wireless environment, as shown in Figure 3.3 of **Chapter 3**. Since a sufficient amount of similarity between the source and target wireless environments is necessary to perform effectively the TL operation, it is also necessary in this case to consider a similarity measure, as described previously in **Chapter 3**.

We consider two methods (see Figure 4.1). Method 1 uses both exact ToAs and exact locations of `Rxs`. Method 2 uses only exact ToAs. These methods are compared with Method 3 which uses only exact locations of `Rxs` [34].

4.4 Similarity Measure

Generally, the amount of training data needed to perform the TL operation relies on its similarity measure. For instance, less amount of training data is needed in the new target wireless environment, if the source and the target wireless environment are sufficiently similar, and vice-versa. Thus, a similarity measure is needed to determine whether the TL operation is effective (see **Section 2.5** in **Chapter 2**). We design a data-driven similarity measure (DDS), as described previously in **Chapter 3**.

4.5 Data Generation

Similar to **Chapter 3**, we also use the high-accuracy ray-tracing X3D ray model [41], computed using the software Remcom [33] to obtain the power measurements that are sufficiently representative of the real indoor wireless environments. The detailed description to generate data for TL is presented under **Section 3.7** of **Chapter 3**.

4.6 TL Performance

To understand the performance of the proposed TL method, transferability, F1-score and accuracy (see **Section 3.8.1**) are calculated for both cases under each Method. Higher values of these performance measures imply a better performance of the model.

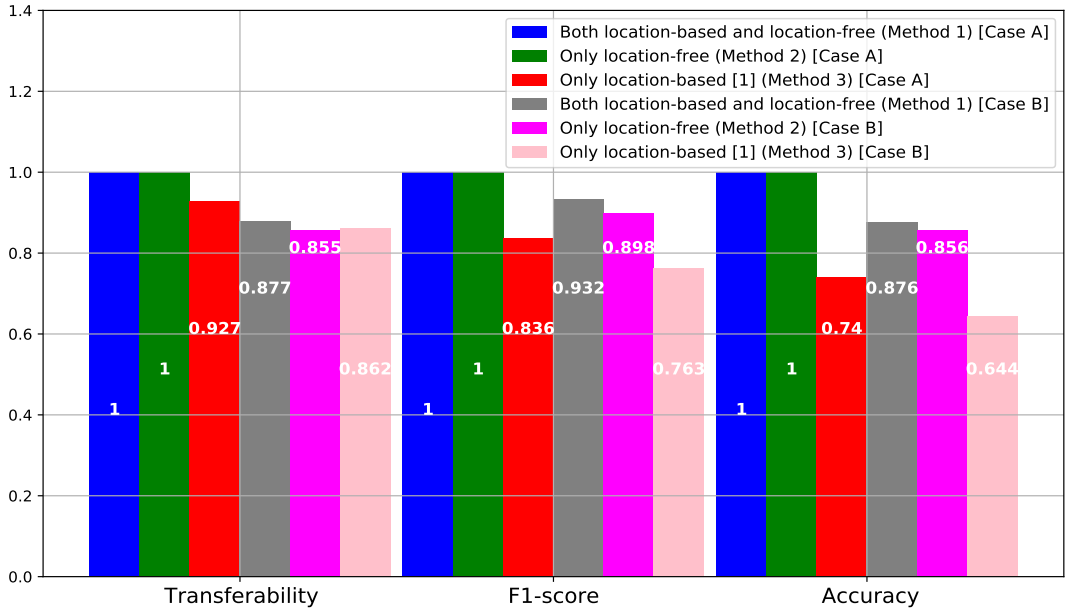


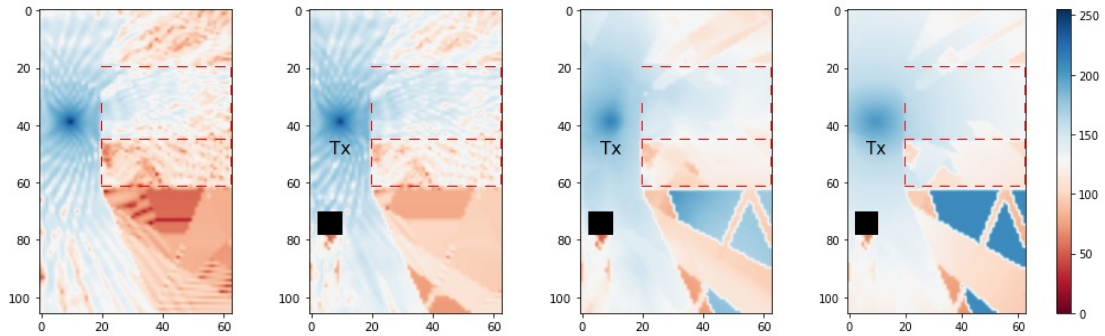
Figure 4.2: TL Performance of different methods.

Figure 4.2 shows the transferability, F1-score and accuracy for each case under each Method along with Method 3 [34] for comparison. An empirical threshold for the test MSE corresponding to TL, and the number of training epochs are set as 0.01 and 20, respectively, for each case of each Method. Two environments are recognized as sufficiently similar, if both obtained values are less than these thresholds, else, the TL operation is not recommended. We can see that all these measures have higher values under case A for both Method 1 and Method 2 than for case B. In addition, 100% accuracy is achieved by Method 1 and Method 2 under case A, which is higher than the corresponding value under case B. On the other hand, all measures under case B corresponding to Method 2 are slightly lower than the corresponding values under case B corresponding to Method 1.

Moreover, while comparing all these measures, we notice that both case A and case B are performing better under Method 1 and Method 2 than the corresponding cases under Method 3. In particular, there is an improvement of around 26% in accuracy under Method 1, for case A and case B, respectively, as compared to Method 3. Similarly, there is an improvement of around 24.5% in accuracy under Method 2 (only location-free) for case A and case B, as compared to Method 3 (only location-based). In addition, the values of all the performance measures under Method 2 (only location-free) are almost the same as Method 1, for both case A and case B. Generally, Method 1 (both location-based and location-free) performs the best as it has both features, as expected. Numerically, Method 2 (only location-free) performs better than Method 3 (only location-based). This can be caused by the fact that the received power is easier to relate to the ToA than the `Rxs` locations.

4.7 Visualization of Radio Maps

For the purpose of visualization, we present different radio maps which are estimated using the various methods, as shown in Figure 4.3. The cubes and the walls of the



(a) (b) (c) (d)

Figure 4.3: Radio map for: (a) original indoor wireless environment, (b) indoor wireless environment where a single object, represented by a cube, is present, (c) the same wireless environment as in (b) after performing TL using both ToA features and locations of `Rxs` (Method 1), where only 40% of training data is used for training the DNN resulting in a test MSE corresponding to TL as 0.0020, and (d) the same wireless environment as in (b) after performing TL using only ToA features (location-free) (Method 2), where only 40% of training data is used for training the DNN resulting in a test MSE corresponding to TL as 0.0036.

rooms are represented in black and dashed red colours, respectively. The differences in the radio maps can be easily visualized.

4.8 Summary

- We address the problem of transfer learning in radio map estimation for indoor wireless communications in scenarios having wireless multi-path, where it may be difficult to have accurate location estimations.

- We consider the use of ToA features of the radio signals instead of using directly the locations of `Rxs`. Additionally, we consider the combination of ToAs and locations of `Rxs`.

- We design an effective data-driven TL method that transfers and fine-tunes a DNN-based radio map model, learned from an original indoor wireless environment, to other different and similar indoor wireless environments. We consider three methods: Method 1 is the combination of location-based and location-fee; Method 2 is only location-free; and Method 3 is only location-based.

- We design a data-driven similarity measure to investigate the similarity between the source and the target wireless environments. Our DDS can predict the amount of training data needed in new target environments when performing the TL operation.

- We also investigate the suitability of the widely used similarity measure, that

is, Wasserstein distance in estimating radio maps, and find that it is not applicable to our radio map estimation problem.

- Method 1 (combination of location-based and location-free), and Method 2 (only location-free) require less amount of training data, as compared to Method 3 (only location-based) [34], in order to perform TL operation in new target wireless environments for estimating radio maps.
- Both Method 1 and Method 2 have better accuracy as compared to Method 3 [34].
- Use of only location-free ToA features is satisfactory for estimating accurate radio maps in the new indoor wireless environments in scenarios of having wireless multi-path, where it may be difficult to have accurate location estimations.
- The radio map estimated using only location-free ToA features (Method 2) is almost as good as Method 1 (combination of location-based and location-free).

Chapter 5

Transfer Learning for Radio Map Estimation using Mixture of Experts

This chapter summarizes Paper D [35]. The results obtained here reply **Q3**.

5.1 Motivation

As shown in **Chapter 4**, with the incorporation of methods that use both estimated sensor locations and methods that use the time of arrival (ToA) features, one can estimate accurate radio maps. Along these lines, we develop a mixture of experts (MoE)-based radio map model in the original wireless environment which comprises two deep neural network (DNN)-based experts. Expert 1 is location-based, trained with the estimated locations of receivers (`Rxs`) and corresponding power values, and Expert 2 is location-free, trained with the ToA features and corresponding power values. In addition, a gating network, which is also a DNN, combines the output of both experts to estimate the final radio map. This MoE-based radio map model is then transferred to new wireless environments and fine-tuned to estimate the corresponding radio maps in those environments. Moreover, our designed data-driven similarity measure coordinates with the transfer learning (TL) operation, similar to **Chapter 3**, and predicts the amount of training data needed to perform TL operation in the new wireless environments. In general, location-free estimation has more input parameters than the location-based. Thus, it requires more training data to sufficiently train its bigger network. Note that this will be the case only if the location estimation is sufficiently accurate for the location-based method. However, when the location estimation is not sufficiently accurate, then the location-based method will require more training data as compared to the location-free method that uses accurate ToA features. In the case that both the locations of `Rxs` and ToAs are inaccurate, the location-free estimation will require more training data, but it will converge to a more accurate estimation since it has more features. Thus, to take advantage of both methods, MoE is expected to be beneficial.

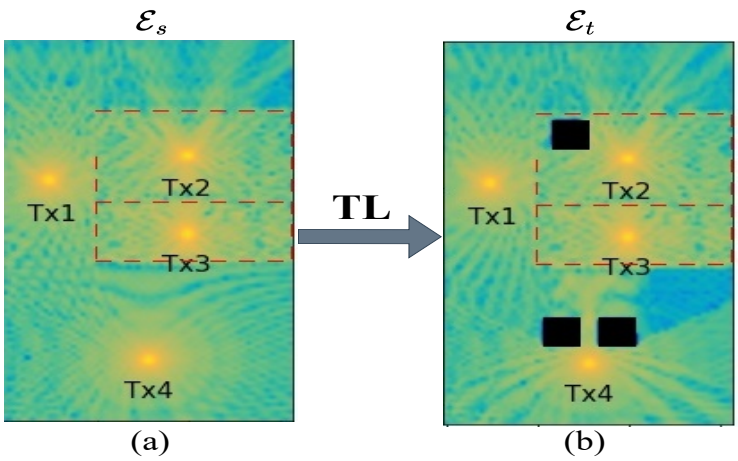


Figure 5.1: Illustration of radio maps: (a) original wireless environment, and (b) target environment with 4 transmitters `Txs` and 3 objects represented by 3 cubes.

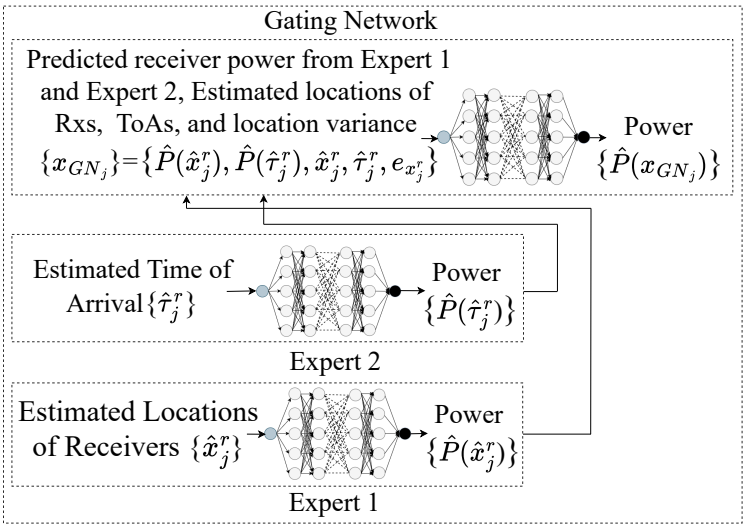


Figure 5.2: MoENet: Baseline MoE model.

5.2 Problem Formulation

To design a TL-based radio map estimation model using MoE, we consider three parts, namely, the design of a MoE, the TL operation, and the TL similarity measure. Firstly, we consider two separate experts. Expert 1 is location-based having the estimated locations of `Rxs` $\{\hat{x}_j^r\}$ as input (where j is the index of the `Rx`), and $F_1(\hat{x}_j^r) = \hat{P}(\hat{x}_j^r)$ as output, the corresponding aggregate power. Similarly, Expert 2 is location-free having the estimated ToAs $\{\hat{\tau}_j^r\}$ as input, and $F_2(\hat{\tau}_j^r) = \hat{P}(\hat{\tau}_j^r)$ as output, the corresponding aggregate power. Then, considering the MoE principle [66], we can combine both experts using a gating network $G(x_{GN_j})$. The input to the gating network is $\{x_{GN_j}\} = \{\hat{P}(\hat{x}_j^r), \hat{P}(\hat{\tau}_j^r), \hat{\tau}_j^r, \hat{x}_j^r, e_{x_j^r}\}$, where $\hat{P}(\hat{x}_j^r)$, and $\hat{P}(\hat{\tau}_j^r)$ are the predicted receiver powers from Expert 1 and Expert 2, respectively, and $e_{x_j^r}$ is the location estimation error, assumed to be known. In practice, $e_{x_j^r}$ can be either estimated by performing location estimation measurements in advance, e.g., varying the wireless environment and estimating (offline) $\tilde{e}_{x_j^r}$ at each position of the potential receiver and calculating the average of these errors across environments (envs)

$e_{x_j^r} = \mathbb{E}_{\text{envs}}\tilde{e}_{x_j^r}$, or through a localization algorithm that is able to estimate this error. This error generally defines an uncertainty region around the estimated location where the actual location resides. Its output is $G(x_{GN_j}) = \hat{P}(x_{GN_j})$, that is, the aggregate received power at $\texttt{Rx}_j$. Our problem is how to learn functions $F_1(\cdot)$, $F_2(\cdot)$, and $G(\cdot)$.

Let $\mathcal{E}_s$, and $\mathcal{E}_t$ denote the source (original) and target wireless environments, respectively, as shown in Figure 5.1. Next, let us assume that only a small amount of power values are available in $\mathcal{E}_t$, as compared to the number of samples in $\mathcal{E}_s$.

For the source environment $\mathcal{E}_s$, we first learn the DNN-based MoE $G_s(\cdot)$, as shown in Figure 5.2, with weight parameters $\theta_s^\star$ through minimizing the loss function mean square error (MSE), MSE $= \frac{\sum_{i=1}^{N_s}[P(x_i)-G_s(\cdot)]^2}{N_s}$, where N_s is number of $\texttt{Rxs}$. Let us also assume that $\mathcal{E}_s$ and $\mathcal{E}_t$ have some similarity in terms of wireless propagation characteristics. We perform TL by transferring the baseline MoE model learnt in $\mathcal{E}_s$ to the target environment $\mathcal{E}_t$, initializing its parameters by the values obtained from the baseline MoE model of $\mathcal{E}_s$. Then, we fine-tune this model to minimise the same loss function but in the target environment $\mathcal{E}_t$, by using some additional measurement samples. If the TL operation is successful, this additional number of samples will be small. The detailed mathematical formulation of TL can be seen in [2]. The similarity measure is defined in the same way as in **Chapter 3**, where the baseline model is now given by *MoENet*, instead of *MapNet*.

5.3 Radio Map Estimation Model Using TL

Similarly to **Chapter 3**, the proposed model has two phases, namely; the training phase and the execution phase. The training phase, as shown in **Chapter 3**, consists of two stages: (i) the design of the baseline MoE model in the source environment $\mathcal{E}_s$, followed by the possibility of transferring it to the target environment $\mathcal{E}_t$, and (ii) the establishment of a similarity measure between $\mathcal{E}_s$ and $\mathcal{E}_t$. The execution phase, shown in Figure 5.3, takes a floor plan image of a new $\mathcal{E}_t$ as input. It first checks if TL can be performed based on our data-driven similarity measure (DDS). If TL can be performed then we obtain the percentage of training data needed to perform TL and the corresponding radio map. On the other hand, if TL can not be performed, then a new baseline model is trained from the very beginning.

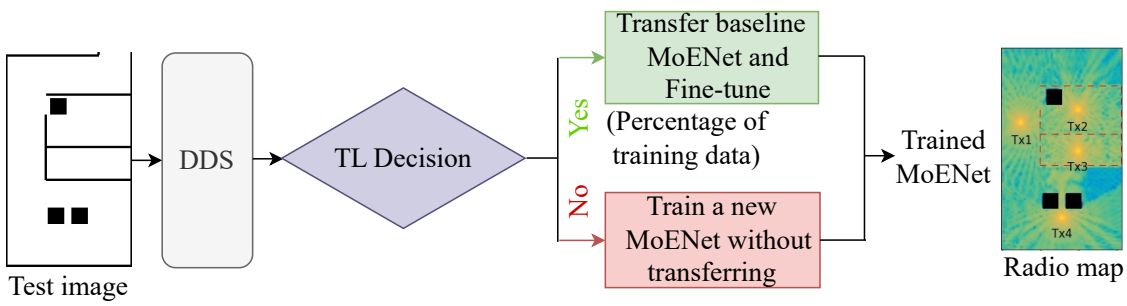


Figure 5.3: Data-driven transfer learning-based radio map estimation model using MoE (execution phase).

5.4 Transfer Learning Approach

As shown in Figure 5.2, Expert 1 is location-based having the power values obtained at multiple estimated locations of `Rxs` in the original wireless environment (see Figure 5.1(a)). The estimated locations of `Rxs` are obtained using the trilateration localization algorithm [1] (see **Section 2.3**). Expert 2 is location-free mapping of the power values to the estimated ToA features for the same original wireless environment. We use 2 ToA features for every `Tx`. The gating network takes the outputs of each expert, the inputs of each expert, and the location estimation error $e_{x_j^r}$ to obtain the final output (power values) for estimating the radio maps. Both experts and the gating network are trained jointly as feed-forward DNNs for the original wireless environment. The trained baseline MoE model, as shown in Figure 5.2, is referred to as *MoENet*.

Then, given the baseline MoE model *MoENet*, we first decide whether to perform TL operation based on the similarity measure (DDS). However, since our similarity measure is data-driven, a sufficient amount of data obtained from performing TL operation is also necessary to learn this similarity measure. To this end, we transfer the *MoENet* to each individual target wireless environment $\mathcal{E}_t$, followed up by fine-tuning using a small amount of additional training data from $\mathcal{E}_t$.

After training and fine-tuning the transferred *MoENet* for each new target wireless environment, we store the test MSE, i.e., $MSE(\mathcal{E}_t)$ and the number $N_{EP}(\mathcal{E}_t)$ of training epochs, which are used further to investigate similarity measures between the source $\mathcal{E}_s$ and the target $\mathcal{E}_t$ wireless environments.

Next, we design a data-driven similarity measure (DDS), similar to the one in **Chapter 3**. We use the same Method A and Method B for our DDS, where the test MSE, i.e., $MSE(\mathcal{E}_t)$ and number $N_{EP}(\mathcal{E}_t)$ of training epochs are obtained from *MoENet* instead of *MapNet*.

A high-level description of DDS in the form of pseudo-code is presented in **Algorithm 4**.

5.5 Model Performance

We use Remcom [33] to obtain 1046 different target wireless environments as before. Additionally, we use the trilateration algorithm [1] to obtain estimated `Rx` locations for each wireless environment. Moreover, we add Gaussian noise with zero mean and a variance of 1 dB to the ToA features of each wireless environment.

Table 5.1: Transferability, F1-score, and Accuracy

Performance measures	Method A MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
Transferability	0.957	0.970
F1-score	0.859	0.851
Accuracy	0.765	0.747

Algorithm 4: Data-driven Similarity Measure *SimNet*

Input: $D := \{\text{Im}_t, \text{MSE1}_t, \text{MSE2}_t, \text{MSE3}_t, N_{EP_t}(\text{only for Method B})\}$: dataset comprising three-level images of target environments, 8 sets of MSEs (MSE1, MSE2, MSE3) of the transferred model after TL for different data splits $r_t = \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%\}$, Training epochs of the transferred model after TL for all 8 sets of data splits, for each environment, respectively

MSE_{thr}: MSE threshold, $N_{EP_{thr}}$: Epoch threshold, α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}^{cnn}: Training epochs of *SimNet*

Output: $SimNet(\theta_{sim})$, MSE^{cnn}: MSE of trained CNN model, TL decision: Yes/No

Data-Splitting: Split data into training D_{train} & test D_{test} with ratio r

1: `Training stage`
2: Use data from D_{train}
3: Normalize image values Im_t, MSE1_t, MSE2_t, MSE3_t and N_{EP_t} to $[0, 1]$
4: Randomly initialize θ_{sim}
5: **for** k = 1,...,N_{EP}^{cnn} **do**
6: **for** j = 1,...,N_B = ceil($|D_{\text{train}}|/B$) **do**
7: Randomly select B training samples from D_{train} as the training batch
8: Update θ_{sim} with learning rate α to minimize *loss* given by the

$$\begin{aligned} \text{MSE}^{cnn} &= L_A = |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t|^2 \quad \text{(Method A)} \\ &= L_B = |[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t, \\ &\qquad N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]|^2 \quad \text{(Method B)} \end{aligned} \tag{5.1}$$

9: **end for**
10: **end for**
11: `Testing stage`
12: Initialize MSE^{cnn}: $\text{MSE}^{cnn} \leftarrow 0$
13: Use test data from D_{test}
14: Predict MSE1_t, MSE2_t, MSE3_t and N_{EP_t} on a given image of Im_t
15: Calculate MSE^{cnn} using (5.1)
16: `TL decision`
17: **for** q = 1,...,$|D|$ **do**
18: **for** each r_t **do**
19: **if** max $(\widehat{\text{MSE1}}_t, \widehat{\text{MSE2}}_t, \widehat{\text{MSE3}}_t) \leq \text{MSE}_{thr}$ **then**
20: TL decision $\leftarrow$ Yes and output r_t, **else**, TL decision $\leftarrow$ No
21: **if** max $(\widehat{\text{MSE1}}_t, \widehat{\text{MSE2}}_t, \widehat{\text{MSE3}}_t) \leq \text{MSE}_{thr}$ & $\widehat{N}_{EP_t} \leq N_{EP_{thr}}$(Method B) **then**
22: TL decision $\leftarrow$ Yes and output r_t, **else**, TL decision $\leftarrow$ No
23: **end if**
24: **end if**
25: **end for**
26: **end for**

To investigate the performance of the proposed model, Table 5.1 presents the transferability, F1-score, and accuracy [2]. It can be observed that the F1-score under each method is the same, while the transferability is slightly better in Method B. The accuracy under Method A is 76.5% and the accuracy under Method B is 74.7%. The reason for observing a smaller accuracy under Method B is because Method B is more strict towards satisfying both the test MSE and the number of training epochs threshold criteria, as compared to Method A, where only the test MSE criterion needs to be satisfied. This suggests that our proposed model can be used effectively for estimating radio maps in new wireless environments with fewer sensor measurements, as compared to the case of training a raw model without TL.

5.6 Visualization of Radio Maps

To visualize the accuracy of the estimated maps, we present different radio maps that are estimated using each expert, as shown in Figure 5.4. The cubes and the walls of the rooms are represented in black and dashed red colours, respectively. The differences in the radio maps can be easily seen.

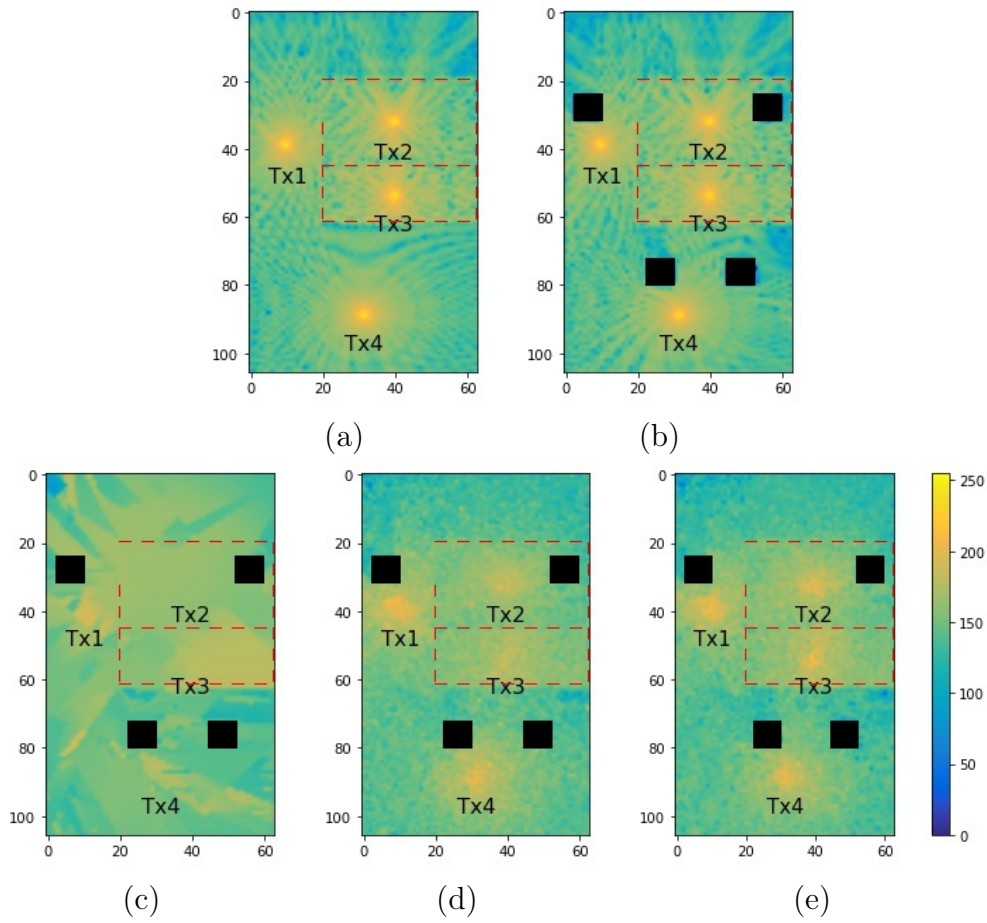


Figure 5.4: Radio map for (a) original wireless environment with `4 Txs`, (b) an environment from Type IV, (c) environment (b) after performing TL using Expert 1 with 40% training data (MSE($\mathcal{E}_t$) = 0.0078), (d) environment (b) after performing TL using Expert 2 with 40% training data (MSE($\mathcal{E}_t$) = 0.006), and (e) environment (b) after performing TL using *MoENet* with 40% training data (MSE($\mathcal{E}_t$) = 0.0054).

5.7 Performance of the TL Model with Additional Complex Changes in Wireless Environments

Additionally, we generated 130 different wireless environments by making the following changes: (a) changing the carrier frequency of waveform from 900 to 950 MHz, (b) changing the location of each `Tx` by 1 meter, (c) adding a wall below to second room, and (d) adding two objects in the floor plan and changing its locations horizontally, and vertically with a spacing of 15 cm. This experiment was successful with a TL-recommendation rate of 99.78% and an overall model accuracy of 72.5%. The radio map of one of these wireless environments is shown in Figure 5.5. The differences in the radio maps can be easily visualized.

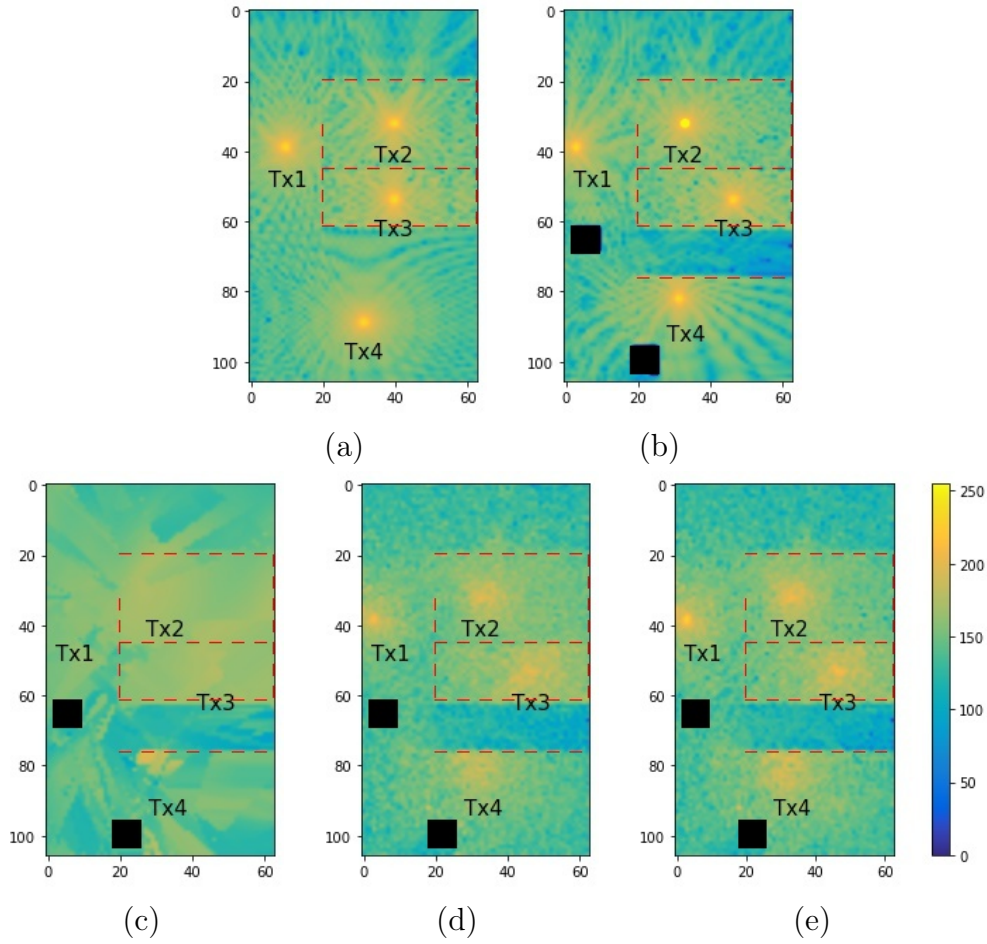


Figure 5.5: Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) environment (b) after performing TL using Expert 1 with 30% training data (MSE($\mathcal{E}_t$) = 0.0096), (d) environment (b) after performing TL using Expert 2 with 30% training data (MSE($\mathcal{E}_t$) = 0.0076), and (e) environment (b) after performing TL using *MoENet* with 30% training data (MSE($\mathcal{E}_t$) = 0.0071).

5.8 Summary

- We design an effective data-driven transfer learning method that transfers and fine-tunes a MoE-based radio map model, learned from an original wireless environment, to other different but similar wireless environments.
- MoE comprises two DNN-based experts, namely; a location-based expert (Expert 1), and a location-free expert (Expert 2), where the output of each expert is connected to a gating network to estimate the final radio maps.
- With different types of wireless environments tested, our proposed MoE-based TL method performs effectively in estimating accurate radio maps, while saving a substantial amount of sensor measurement data, even in the presence of complex changes in the wireless environments, such as the change in `Tx` locations, carrier frequency, additional wall, etc.
- The resulting radio map estimated using MoE is more accurate than the one estimated using either Expert 1 or Expert 2, for the same amount of training data.

Chapter 6

Transfer Learning Based Joint Resource Allocation

This chapter summarizes Paper E [36]. The results obtained here reply **Q4**.

6.1 Motivation

Resource allocation (RA) algorithms are necessary to achieve high performance in both cellular and device-to-device (D2D) networks. Proper/Optimal resource allocation can limit interference and maximize performance metrics, such as total sum rate, energy efficiency, and spectral efficiency, to name a few. However, RA formulations result usually in non-convex optimization and are difficult to solve efficiently. Alternatively, given an optimal high complexity solver for a certain non-convex problem, we can train first offline a deep neural network (DNN) that learns the mapping between instances to the problem and the corresponding optimal (or close-to-optimal) solutions provided by this solver. Then, when a similar (but still different) non-convex optimization problem has to be solved, we can perform first transfer learning (TL) of the DNN-based solver that imitates the optimal (or close-to-optimal) solver for the first problem, and then fine-tune it with a small additional training dataset to learn efficiently DNN-based solver for the other similar problem. As usual, DNNs need a large amount of data for training to learn this mapping. The idea is to use TL to reduce the amount of necessary training data. To this end, we investigate the application of TL for the joint RA (channel assignments and power allocation) in underlay D2D communications. More specifically, we address the problem of learning joint RA (channel assignments and power allocation) to cellular users (CUs) and D2D pairs in an imperfect channel state information (CSI) scenario, by performing TL from a perfect CSI scenario. This is also motivated in this case by the fact that it is easier to obtain a large training dataset for simpler scenarios with perfect CSI as compared to more challenging scenarios with imperfect CSI. Even though we focus on this particular application of RA, the same technique can be applied to other applications that can benefit from performing TL between similar optimization problems.

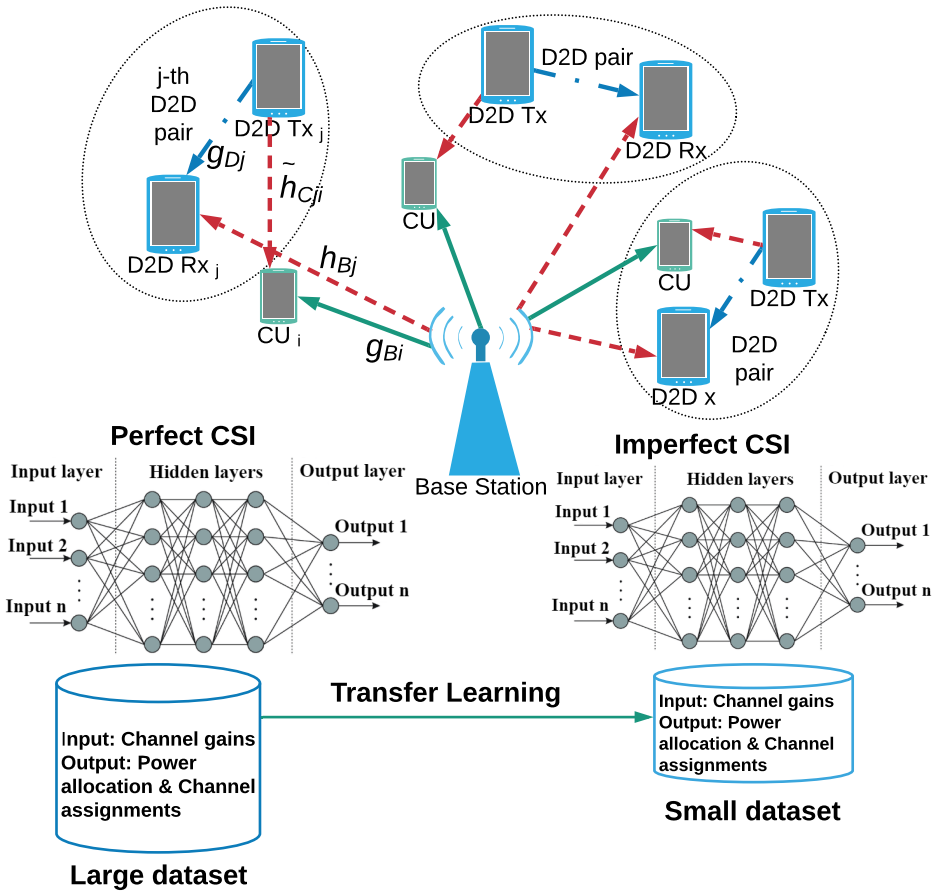


Figure 6.1: Illustration of the system model with TL approach.

6.2 TL System Model

The system model with the TL approach is shown in Figure 6.1. We investigate the transfer of a DNN model trained for resource allocation (channel assignment and power allocation) to cellular users and D2D pairs, from the case of perfect CSI conditions to the case of imperfect CSI conditions.

We consider a cell served by a single base station (BS), which communicates with N_C cellular users (CUs) through N_C downlink channels. The cell is assumed to operate in, without loss of generality, a fully loaded mode, where all channels are assigned to cellular users; thus, the CUs can be indexed by $\mathcal{C} = \{1, ..., N_C\}$. Next, we consider N_D D2D pairs (indexed by $\mathcal{D} = \{1, ..., N_D\}$) wishing to communicate in underlay[1] mode using the aforementioned N_C downlink channels. The notations for the channels are as follows: g_{B_i} and g_{D_j} denote respective direct channel gains between the BS to i-th CU and transmitter and receiver of the j-th D2D pair; h_{B_j} and $h_{C_{j,i}}$ denote, respectively, interference channel gain between BS to the receiver of j-th D2D pair and the transmitter of j-th D2D pair to i-th CU. Furthermore, we denote the total noise power in any channel by N_0.

[1]In underlay D2D communication, channels (e.g. frequency bands or time slots) can be simultaneously used by both D2D users and traditional CUs.

6.2.1 Resource Allocation under Perfect CSI

In this problem, we follow the same formulation as in [70]. The objective of this problem is the sum rate maximization of both the cellular and the D2D network along with a fairness measure in channel assignment to the D2D pairs. Here, all channel gains g_{B_i}, g_{D_j}, h_{B_j} and $h_{C_{j,i}}$; $1 \leq i \leq N_C$, $1 \leq j \leq N_D$ are perfectly known at the BS. Let $\Gamma(z) := \log_2(1+z)$; the sum rate over i-th channel is defined as: $R_i := \sum_{j\in\mathcal{D}} \beta_{i,j}[R_{C_{i,j}} + R_{D_{j,i}}] + (1 - \sum_{j\in\mathcal{D}} \beta_{i,j})R_{C_{i,0}}$, where $R_{C_{i,j}} = \Gamma(P_{B_i} g_{B_i}/(N_0 + P_{D_{ji}} h_{C_{j,i}}))$ denotes the rate of the i-th CU when sharing the channel with the j-th D2D pair ($\beta_{ij} = 1$); $R_{D_{j,i}} = \Gamma(P_{D_{ji}} g_{D_j}/(N_0 + P_{B_i} h_{B_j}))$ the rate of the j-th D2D pair when sharing the channel with the i-th CU ($\beta_{ij} = 1$); and $R_{C_{i,0}} = \Gamma(P_{B_{max}} g_{B_i}/N_0)$ the rate of the i-th CU when it shares its channel with no D2D pair ($\beta_{ij} = 0\ \forall j$). The overall network rate of both cellular and D2D networks can be expressed as: $R := \sum_{i\in\mathcal{C}} R_i$. The overall resource allocation optimization problem under perfect CSI can be formulated as:

$$\max_{B,P_B,P_D} \quad R(B, P_B, P_D) - \gamma\delta^2(B) \tag{6.1a}$$

$$\text{subject to} \quad \beta_{i,j} \in \{0,1\}, \ \sum_{j=1}^{N_D} \beta_{i,j} \leq 1 \quad \forall i \tag{6.1b}$$

$$0 \leq P_{B_i} \leq P_{B_{max}} \quad \forall i \qquad 0 \leq P_{D_{ji}} \leq P_{D_{max}} \ \forall j, i \tag{6.1c}$$

$$\frac{P_{B_i} g_{Bi}}{N_0 + P_{D_{ji}} h_{C_{j,i}}} \geq \eta^{C_{min}} \quad \forall i, j, \quad \text{if } \beta_{ij} = 1 \tag{6.1d}$$

$$\frac{P_{D_{ji}} g_{Dj}}{N_0 + P_{B_i} h_{B_j}} \geq \eta^{D_{min}} \quad \forall i, j, \quad \text{if } \beta_{ij} = 1. \tag{6.1e}$$

where $\eta^{C_{min}}$ and $\eta^{D_{min}}$ are the respective minimum SINR requirements for CUs and D2D pairs. The binary variable $\beta_{i,j}$ denotes the channel assignment to the j-th D2D pair; $\beta_{i,j} = 1$, if the i-th CU shares the channel with the j-th D2D pair, and $\beta_{i,j} = 0$ otherwise. It is assumed that each D2D pair can access multiple channels at the same time, but no channel can be used by multiple D2D pairs simultaneously, which implies that $\sum_{j=1}^{N_D} \beta_{i,j} \leq 1$, $\forall i$. P_{B_i} and $P_{D_{j,i}}$ denote, respectively, the transmit power allocated to the BS over the i-th channel and the j-th D2D pair when accessing the i-th channel. The unfairness measure $\delta^2(B)$ (from [71]) is defined as $\delta^2(B) := 1/(N_D x_0^2) \sum_{j=1}^{N_D} (x_j(B) - x_0)^2$, where $x_j := \sum_{i=1}^{N_C} \beta_{i,j}$ is the number of channels assigned to the j-th D2D pair, $x_0 := N_C/N_D$ and B is channel assignment.

$\delta(B)$ can be interpreted as standard deviation of x_j from their fairest value x_0 and thus, assigning more uneven channels among D2D pairs, will lead to a larger $\delta(B)$. Note that the expression for R has a sum of the log functions of ratios, which makes the expression non-convex. Additionally, the variable $\beta_{i,j}$ is binary. These two issues make the optimization problem a mixed-integer non-convex problem.

This resource allocation optimization problem can be solved efficiently [70] to find close-to-optimal solutions, allowing to obtain efficiently a large dataset to learn a DNN-based solver model.

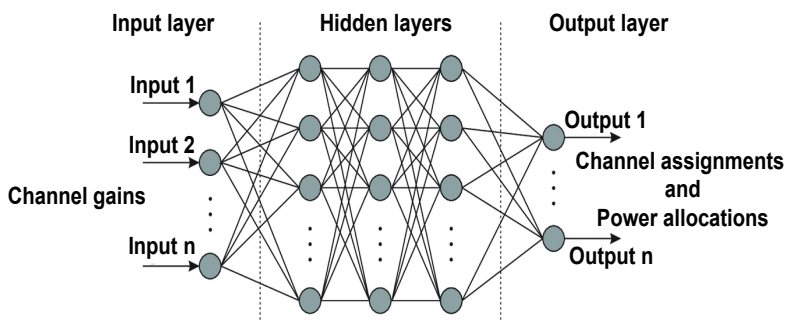


Figure 6.2: Baseline DNN with input and output.

6.2.2 Resource Allocation under Imperfect CSI

In this problem, we follow the same formulation as in [72]. Here, the interference channel gain is assumed to be exponentially distributed (Rayleigh fading[2] channel response) and is denoted by $\tilde{h}_{C_{j,i}}$, $1 \leq i \leq N_C$, $1 \leq j \leq N_D$. In the case of imperfect knowledge of CSI, the stochastic SINR constraint (6.1d) is replaced with a probabilistic constraint to guarantee a minimum outage probability ϵ, and is expressed as:

$$\frac{P_{B_i} g_{B_i}}{N_0 + P_{D_{ji}} F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon)} \geq \eta^C_{min} \tag{6.2}$$

where $F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon)$ is the inverse cumulative distribution function (CDF) for $\tilde{h}_{C_{j,i}}$ evaluated at $(1-\epsilon)$. The solution to this non-convex mixed-integer resource allocation problem in [72] is much more computationally demanding as compared to [70]. Thus, generating large datasets to train and learn a DNN-based solver is difficult.

6.3 Transfer Learning Approach

We first learn a baseline DNN model under the perfect CSI scenario with the dataset generated from the algorithm presented in [70] where the input of the DNN comprises channel gains, and the output of the DNN comprises both channel assignments and power allocations. The baseline model used for TL is shown in Figure 6.2.

The baseline DNN model is then transferred to the imperfect CSI scenario and fine-tuned with a small dataset generated from the algorithm presented in [72].

6.4 Similarity Measure

In this work, we have not designed a data-driven similarity measure (DDS) as before. However, we show that the outage probability can indicate how similar the imperfect CSI scenario is to the perfect CSI scenario. The smaller it is, the more different the

[2]Rayleigh fading is suitable for indoor scenarios where there is high multipath and no line of sight (LOS).

scenarios are. This measures the similarity between the optimization problems, which in turn measures the similarity between the solvers.

Moreover, we prove that under the assumption that the interference channel gains $\tilde{h}_{C_{j,i}}$ follow an exponential distribution (Rayleigh fading channel response) with a mean equal to the true channel gain value in the perfect CSI case, the two resource allocation problems for perfect CSI and imperfect CSI are equivalent for an outage probability $\epsilon = \frac{1}{e}$.

Note that equivalent optimization problems have the same set of solutions. However, a solution provided by one solver for a problem might be different from another solver of an equivalent problem, since there might not be a unique solution.

6.5 Data Generation

The simulation setup comprises a circular cell of radius 500m in which the CUs and D2D transmitters are placed uniformly at random. Each D2D receiver is placed uniformly at random inside a circle of radius 5m centred at the corresponding transmitter. The channel gains are calculated using a path-loss model with exponent 2 and gain -5 dB at a reference distance of 1m. We assume $\tilde{h}_C$ to be exponentially distributed with the mean value obtained from the mentioned path-loss model. Averages over 100,000 independent realizations of the user locations with parameters $BW = 15$ kHz, $\gamma = 50 \times BW$, $N_D = 5$, $N_C = 5$, $N_0 = -70$dBW (γ is scaled with BW to ensure that the unfairness and the achieved rate are of comparable values) are performed. Thus, the input to the DNN is the set of channel gains, which accounts for 40 inputs. The output of the DNN is a joint set of 50 power allocations and 25 channel assignment variables. For training the baseline DNN model with perfect CSI, we consider 100,000 input-output pair samples.

6.6 Numerical results

We show how the amount of training data required to fine-tune the baseline model depends on the degree of mismatch between the tasks of solving the two optimization

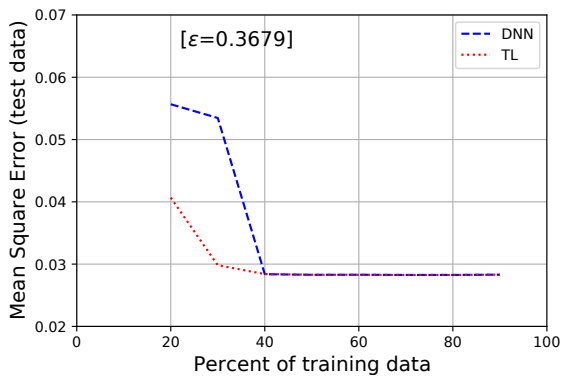


Figure 6.3: Percent of training data vs MSE (Number of epochs = 10).

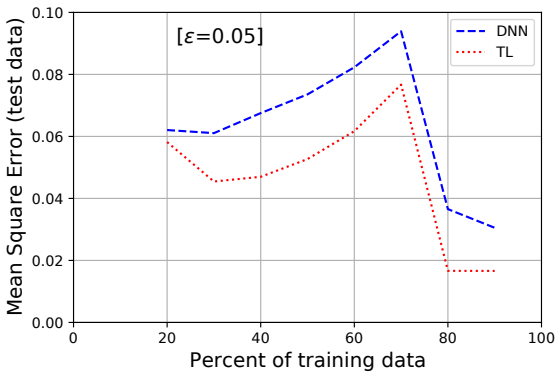


Figure 6.4: Percent of training data vs MSE (Number of epochs = 10).

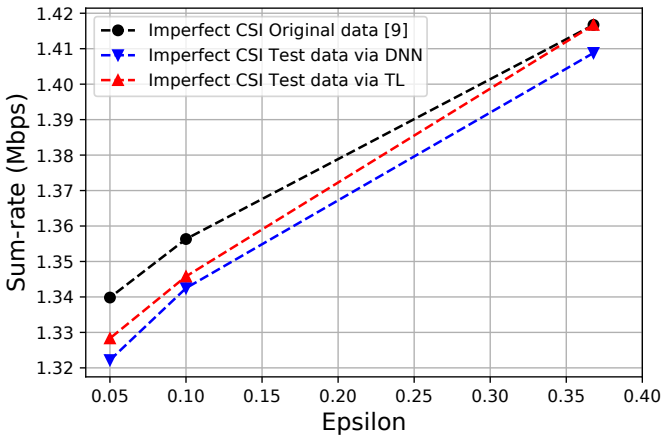


Figure 6.5: Sum-rate for different values of outage probability ϵ.

problems. For example, Figure 6.3 shows that when two problems are equivalent, i.e., outage probability $\epsilon = 0.3679$, then with only 30% of training data, the transferred baseline model is completely trained. However, Figure 6.4 shows that when there is a large mismatch between the two problems, i.e., $\epsilon = 0.05$, then almost 80% of training data is required for the complete training of the transferred baseline model.

Next, Figure 6.5 shows the plot of sum-rate obtained for different values of outage probability ϵ, that is, $\epsilon = 0.05$, $\epsilon = 0.1$ and $\epsilon = 0.3679$, for the case of imperfect CSI [72]. It can be noticed that the resulting sum-rate achieved after performing TL is higher than the one obtained from a DNN-based solver model trained without TL. Specifically, by performing TL, we achieve the same sum-rate as the original solver [72] for $\epsilon = 0.3679$, a condition in which the two resource allocation problems for perfect CSI and imperfect CSI (under an outage probability $\epsilon = 0.3679$) are equivalent. This signifies that even with a small number of training samples, by performing TL, we achieve nearly the same performance as the original solver [72], and superior performance as compared to using a DNN-based solver without TL for the same amount of training data. Hence, it results in a substantial reduction of the required computational resources.

6.7 Summary

- We investigate the application of transfer learning for joint resource allocation (channel assignment and power allocation) in underlay D2D communications.
- We design a baseline DNN model for the data generated from a perfect CSI scenario and then transfer it to an imperfect CSI scenario and fine-tune it with an additional small amount of data generated from the imperfect CSI scenario.
- The baseline DNN model transferred from the perfect CSI scenario improves the learning for the imperfect CSI scenario while requiring a (relatively) small amount of additional training data.
- We show how the similarity between two RA problems can be measured by the desired outage probability ϵ. At $\epsilon = \frac{1}{e} = 0.3679$, both RA problems are equivalent. As a result, less amount of training data is required for fine-tuning the transferred baseline DNN model in the imperfect CSI scenario.
- For other smaller values of ϵ, there is a larger mismatch between the two RA problems, and a larger amount of training data is required for fine-tuning the transferred baseline DNN model in the imperfect CSI scenario.
- The amount of data required for fine-tuning the transferred baseline DNN model in the imperfect CSI scenario is significantly smaller as compared to the training of a new DNN model without performing TL.
- The sum-rate obtained by training the imperfect CSI model using TL, is higher than the DNN without TL for the same amount of training data. At the point when both RA problems are equivalent ($\epsilon = \frac{1}{e}$), TL achieves the original sum rate while saving computational resources.

Chapter 7

Concluding Remarks

7.1 Conclusion

This dissertation proposes transfer learning (TL) methods for: (a) estimating radio maps in wireless communications, and (b) resource allocation. The four major areas covered in this dissertation are; *(i)* TL methods to combat the data-scarcity challenge (location-based and location-free), *(ii)* similarity measure between wireless environments for machine learning algorithms in the context of wireless networks, *(iii)* TL-based non-linear mixing of two non-linear experts (location-based and location-free), and *(iv)* TL between similar resource allocation problems with a focus on the TL from perfect channel state information (CSI) to imperfect CSI. A detailed description of the proposed methods can be found in **Chapters 3 - 6**, while motivation and background can be found in **Chapters 1 and 2**, respectively.

In **Chapter 3**, we assume that we have an exact knowledge of receivers `Rxs` locations and corresponding power values for the wireless environment, which we refer to as the *location-based TL method*. Based on this, we propose a data-driven TL method to estimate radio maps in new and sufficiently similar wireless environments. We design a baseline radio map model in the original (source) wireless environment and transfer and fine-tune it to the new wireless environments. We assume that a relatively small amount of training data is available in new wireless environments. We also design a data-driven similarity measure (DDS) in order to measure, in terms of wireless propagation characteristics, the similarity between the original and new wireless environments. Our DDS also allows us to predict the accuracy, in terms of mean square error (MSE), and the amount of training data needed to perform TL operations in new wireless environments for estimating radio maps.

In **Chapter 4**, we assume that there is a high degree of wireless multipath in the wireless environments and hence it is difficult to obtain accurate `Rxs` locations. Then, we employ time of arrival (ToA) features of radio signals and learn a map between the ToAs and corresponding power values for the wireless environment, and we refer to this method as *location-free TL method*. More specifically, we design a baseline radio map model employing ToA features in the original (source) wireless environment and then we transfer and fine-tune it to the new wireless environments. Similar to **Chapter 3**, our DDS can also predict the accuracy, in terms of MSE,

Table 7.1: Comparison of chapters

	Location-based	Location-free	Mixture of location-based and location-free	Resource allocation
Chapter 3	✔	✖	✖	✖
Chapter 4	✖	✔	✖	✖
Chapter 5	✔	✔	✔	✖
Chapter 6	✖	✖	✖	✔

and the amount of training data required to perform TL operations in new wireless environments for estimating radio maps.

In **Chapter 5**, we estimate radio maps of new wireless environments using a mixture of experts (MoE)-based method. We design a MoE-based radio map model in the original wireless environment which comprises two deep neural network (DNN)-based experts. Expert 1 is the estimated location-based model, trained with the estimated locations of `Rxs` and the associated power values, while Expert 2 is the location-free model, trained with the estimated ToA features and the associated power values. A gating network, which is also a DNN, is also trained to combine the output of both experts and estimate the final radio maps. This MoE-based radio map model is transferred to the new wireless environments and fine-tuned to estimate the corresponding radio maps. The proposed MoE-based TL method performs better than Expert 1 and Expert 2 experimentally and saves a substantial amount of sensor measurement data, as compared to the scenario without using TL.

In **Chapter 6**, we cover TL between solvers of similar resource allocation problems. In particular, we assume that it is possible to obtain a large amount of training dataset for resource allocation (RA) scenarios with perfect CSI as compared to scenarios with imperfect CSI. We propose a TL-based method for joint resource allocation (channel assignments and power allocation) over D2D communications. A baseline model is learned with the data corresponding to the perfect CSI, then transferred and fine-tuned for the imperfect CSI scenarios. We also show how the desired outage probability in the RA problem can be used to measure the similarity between the perfect and the imperfect CSI scenarios. The total sum-rate of both the cellular users (CUs) and the D2D pairs obtained by training the imperfect CSI solver using TL, is higher than the DNN without TL, for the same amount of training data. A summary of the key differences between different methods proposed in the various chapters is presented in Table 7.1.

7.2 Future Work

In this dissertation, we have studied the design of data-driven TL methods in wireless networks, including the design of similarity measures, for estimating accurate radio maps, and performing accurate resource allocation tasks. In terms of future work, there are several key challenges that need to be looked into in the future.

- Adapting our proposed TL method of radio map estimation to select the best baseline model out of multiple potential baselines. For example, different classes (clusters) of source wireless environments which are used to design multiple baseline models (cluster head) for the TL operation may comprise multiple objects or different numbers of transmitters to design multiple baselines. When a new environment comes, the proposed method should pick (based on the similarity) the closest cluster/class and use its baseline for the TL operation.

- Investigating the suitability of our proposed TL method of radio map estimation in outdoor wireless network scenarios.

- Adapting our proposed TL method of radio map estimation in more challenging real-world wireless environments. For example, dynamic changes in wireless environments over time. In other words, how to update the proposed TL model in real-time using the least amount of sensor measurements if the objects are continuously moving in the wireless environments.

- Incorporating deep generative models, such as generative adversarial networks (GANs) [16] together with our TL methods to estimate radio maps in wireless environments. This two-phase TL method may first explore global information to extract radio propagation patterns and then local features to estimate different wireless effects on radio maps.

- Incorporating tensor-based methods, such as [17], together with our TL methods to work in the scenarios of heavily quantized sensor measurements in wireless environments for estimating radio maps.

- Designing data-driven similarity measures to perform TL operation for different types of network resource allocation problems so that it is possible to perform an efficient TL to save a substantial amount of sensor measurements and network resources.

Appendix A

PAPER A

Title: Deep Transfer Learning Based Radio Map Estimation for Indoor Wireless Communications

Authors: **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano

Conference: IEEE SPAWC 2022

Deep Transfer Learning Based Radio Map Estimation for Indoor Wireless Communications

R. Jaiswal, M. Elnourani, S. Deshmukh, B. Baltasar Lozano

Abstract: **This paper investigates the problem of transfer learning in radio map estimation for indoor wireless communications, which can be exploited for different applications, such as channel modelling, resource allocation, network planning, and reducing the number of necessary power measurements. Due to the nature of wireless communications, a radio map model developed under a particular environment can not be directly used in a new environment because of the changes in the propagation characteristics, thus creating a new model for every environment requires in general a large amount of data and is computationally demanding. To address these issues, we design an effective novel data-driven transfer learning procedure that transfers and fine-tunes a deep neural network (DNN)-based model for a radio map learned from an original indoor wireless environment to other different indoor wireless environments. Our method allows to predict the amount of training data needed in new indoor wireless environments when performing the operation of transfer learning using our similarity measure. Our simulation results illustrate that the proposed method achieves a saving of 60-70% in sensor measurement data and is able to adapt to a new wireless environment with a small amount of additional data.**

A.1 Introduction

In wireless communications, accurate estimation of channel gain/path loss is important in network design for optimizing the distribution of towers, channel modelling, allocating resources and meeting the expected quality of service (QoS) requirements of the end user. Path loss measures the loss of signal strength (reduction in power, or attenuation) between a transmitter (Tx) and a receiver (Rx) due to large-scale effects. Different factors may cause the attenuation in signal power, for example, free-space propagation loss, reflections, diffractions, etc. from buildings, and objects blocking the line of sight (LOS) between Tx and Rx. Path loss is obtained for each Tx-Rx pair location and is sometimes referred to as path gain or radio map.

Wireless communication is dynamic in nature. For example, a good model for

a radio map that is appropriate in a particular wireless environment, may not be appropriate for a new wireless environment due to changes in propagation characteristics. On the other hand, notice that knowing the strength of signal or received power in one wireless network area (source area) at several spatial locations (sampled radio map), one can make a smart utilization of a pre-developed model to estimate the radio map in another new wireless network area (target area) by exploiting transfer learning (TL) [19] and reduce the need of new measurements at the new environment.

The exploitation of knowledge acquired in the source area for the learning task in the target area is referred to as transfer learning (TL) [19]. It handles the data scarcity issue in the target area. In the context of radio map estimation, one may have configured or setup the wireless system in one indoor environment but need to perform deployment in another environment. TL can be exploited to achieve a good solution in the target area instead of learning a solution from the very beginning by exploiting the knowledge from the source model.

Some of the works that have successfully adopted TL include [3, 20–23, 32]. To capture traffic pattern diversity in cellular data of different cities, a spatial-temporal cross-domain neural network (STCNet) model is proposed in [20]. Model-based TL is exploited for similarities between different kinds of cellular traffic. In [21], TL is exploited to improve the robustness of deep neural network (DNN) based spectrum sensing in cognitive radio with the assumption that the data collected under different characteristics belong to different but related distributions/domains. A TL method via self-imitation is proposed in [32] to tackle the NP-hard mixed-integer nonlinear programming problems of resource allocation. In [22], downlink channel state information (CSI) prediction from uplink CSI using direct model-based TL, is proposed for frequency division duplexing (FDD) in a massive multiple-input multiple-output (MIMO) framework. A TL method based on Wasserstein distance [51] is applied to the wireless fingerprinting localization in [23]. In fact, [3] adopts a RadioUNet model which is a modified UNet architecture [14] (originally designed for biomedical image processing) for estimating radio maps in the urban environment. However, the work in [3] requires training the model from the very beginning for each new wireless environment.

Since users may experience different indoor wireless environments, data collection and training of DNN models from the very beginning are required in new wireless environments. Typically, a large amount of samples and training epochs are required for training a DNN, and thus training a new DNN for each new wireless environment may demand a substantial computational time and data acquisition cost.

In this work, we address the problem of estimating radio maps in indoor wireless communications using TL when the wireless environment changes. We assume that fast fading is averaged out in the measurements. The main contributions of this paper are the following:

- Design of an effective data-driven TL procedure that transfers and fine-tunes a DNN-based model for a radio map learned from an original indoor wireless

environment to other different indoor wireless environments.

- Formulation of a data-driven similarity measure model that predicts the mean square error (MSE) that will be achieved for the estimated radio map in a new wireless environment when performing TL from a baseline (source) environment.
- Prediction of the amount of training data needed in new wireless environments depending on a certain criterion of the MSE threshold for the radio map estimation, when performing the operation of TL using our similarity measure model.
- Extensive testing of our algorithms using simulated data from the Remcom simulator [33]. Numerical experiments demonstrate that the proposed TL method achieves a high success-rate in estimating radio maps accurately for each new indoor wireless environment while using a small amount of training data. We also show experimentally that the Wasserstein distance (WD), widely used in TL, is not applicable to our radio map application.

The remainder of this paper is structured as follows: Section A.2 describes the environments for transfer learning. Section A.3 presents the proposed transfer learning-based radio map estimation. Section A.4 presents and discusses the results followed by concluding remarks in Section A.5.

A.2 Description of Environments for Transfer learning

In order to obtain power measurements that are sufficiently representative for real indoor wireless environments, we use the standard high accuracy ray-tracing *X3D*

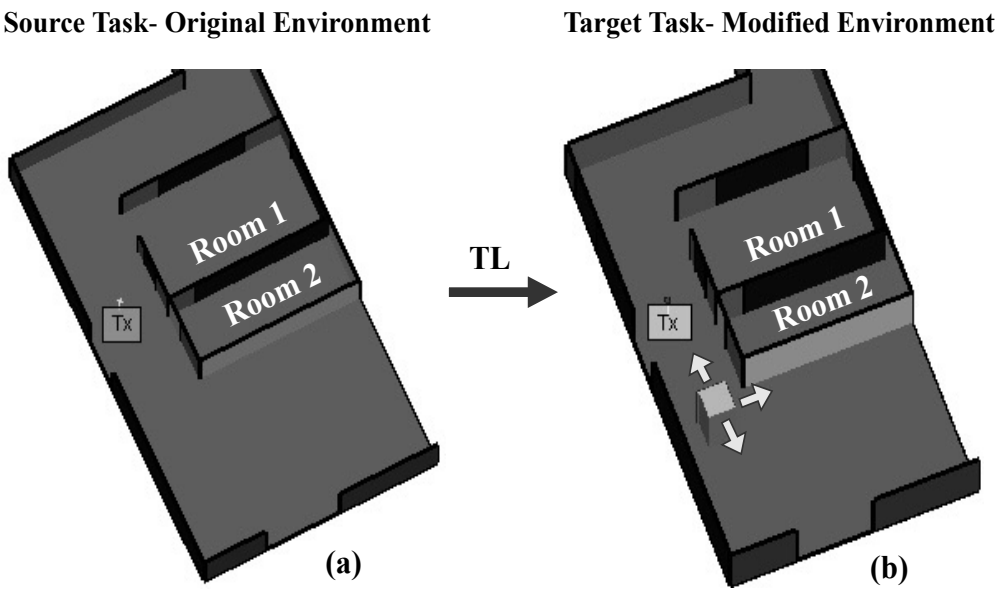


Figure A.1: Environments for TL: Original and modified environments.

Table A.1: Parameters used in Remcom for data generation.

Waveform	Narrow-band Sinusoidal
Carrier frequency	900 MHz
Bandwidth	1.0 MHz
Antenna type	Omni-directional
Tx location	(1.5, 10, 1.3)m
Tx height	1.3m
Tx power	27.73 dBm
Receiver threshold	-250 dBm
Voltage standing wave ratio (VSWR)	1.0
Transmission line loss	0 dB
Space between two Rx points	15cm
Space between two Rx routes	15cm
Noise figure	3 dB
Ray tracing	X3D ray model
Ray spacing	0.2°
Number of reflections	3
Number of transmissions	2
Number of diffractions	0
Volume of the object (single cube)	$1m^3$
Number of Rx locations	6678
Floor dimension (width x length x height)	(9.5 x 20.6 x 2.88)m
Number of wireless environments	250
Image size of floor environment (width x height)	(160 x 275) pixels

ray model [41], computed using the software Remcom [33]. We first consider an environment of a single floor which is comprised of two rooms, in order to obtain the radio map as shown in Figure A.1. We place a transmitter "Tx" at one fixed location (x^t) and then multiple receivers "Rx" at uniformly spaced locations (x_i^r) to obtain the power corresponding to each Rx location ($P(x_i^r)$), where i is the index of the receiver. This results in 6678 Rx locations and corresponding power values. Furthermore, to create different wireless environments, we incorporate an object at different locations, that is, a single solid cube block (made of metal) having volume $1m^3$ with the original environment[1]. The location of this object is changed horizontally, vertically, and diagonally as shown in Figure A.1(b), with an incremental spacing of 15cm. For each location, a new indoor wireless environment is created and the ray-tracing model provides the power for each particular environment with similar Rx locations. A total of 250 different indoor wireless environments are created by changing the location of the object in different directions and the images of each indoor wireless environment are saved as 160 x 275-pixel images. These images are later used also to investigate the suitability of the WD as a similarity

[1]The inclusion is made for the sake of simplicity to build different wireless environments conceptually.

measure for our problem and to develop a data-driven similarity measure between different indoor wireless environments. The parameters used in the Remcom for data generation are summarized in Table A.1.

A.3 Transfer Learning Based Radio Map Estimation

The TL system design proposed for radio map estimation from a source task to a target task is shown in Figure A.1. The arrows in Figure A.1(b) show the direction in which the object, that is, a cube, is moved. Each object location creates a new indoor wireless environment. The fundamental requirement in TL is the notion of similarity between the source and the target task over which TL is performed. Therefore, we now define the source task in which the baseline DNN model is trained and the target tasks in which the baseline/pre-trained model is transferred and fine-tuned, and then establish a similarity measure between the source and the target tasks.

A.3.1 Baseline DNN Model and TL Approach

We design and develop a DNN model with the power values obtained at multiple locations of receivers under the original indoor wireless environment (see Figure A.1(a)). For this purpose, we train a fully-connected DNN model (see Table A.2) and refer to it as "Baseline model", as shown in Figure A.2.

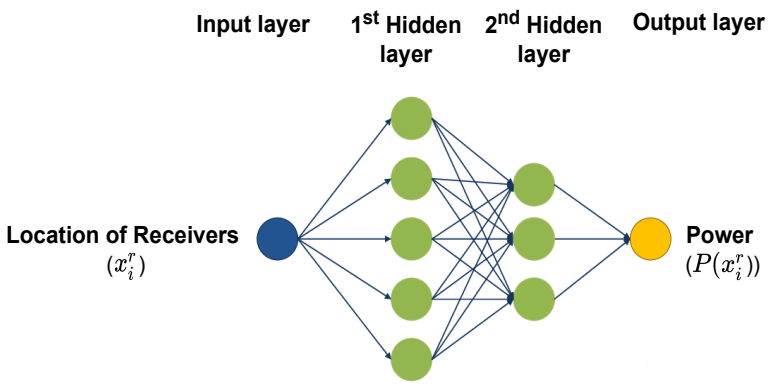


Figure A.2: Radio map estimation DNN model.

Next, we transfer the pre-trained (baseline) model to all new indoor wireless environments one-by-one (total 250) and fine-tune each of them individually. *Our hypothesis is that there is no sufficient amount of training data to train a DNN from the very beginning for the new wireless environments and hence, TL can be exploited to be able to train a DNN with a small amount of training data for estimating the radio map.*

We normalize the data to be between 0 and 1 for the faster training of DNN [3]. We use the rectified linear unit (ReLU) as an activation function and a mini-

batch of 32 samples. The usual MSE for the normalized data, defined as MSE = $\frac{\sum_{i=1}^{r}(P(x_i)-\hat{P}(x_i))^2}{r}$, is used as the loss function, where $P(x_i)$ and $\hat{P}(x_i)$ are the actual and the predicted power samples, respectively, and r is the number of Rx locations. Adam optimizer with a learning rate of 0.001 is used for stochastic optimization. Generally, the training data required to perform TL depends on the similarity, the more similar the environment, the less training data is required. In order to obtain the percentage of training data needed to perform the operation of TL, the whole data is split into eight different train-test split ratios, such as 0.05:0.95, 0.10:0.90, 0.15:0.85, 0.20:0.80, 0.25:0.75, 0.30:0.70, 0.35:0.65, and 0.40:0.60. Here, for instance, 0.05:0.95 specifies 5% training and 95% testing data. After training DNNs for each new indoor wireless environment, we store the test MSE obtained using TL and the number of epochs used in training the DNN model in order to establish the similarity between source and target tasks.

A.3.2 Similarity Measure using Wasserstein Distance

Similarity measures between the source and the target tasks such as Wasserstein distance (WD) [23] have been widely used in TL. WD computes the distance between two distributions. In our case, we compute WD between two images of different wireless environments. For two random variables U and V (U being the image of the environment with the cube present at the left-hand side bottom corner, and V being the image of remaining environments with the cube present at different locations) with respective cumulative distribution function (CDF) $F_U(u)$ and $F_V(v)$, the WD, which we denote as d, is defined as [51]

$$d(U,V) = \inf_{F_{UV}\in\mathcal{F}}\{\mathbb{E}_{F_{UV}}|U-V|\} \tag{A.1}$$

where, $\mathcal{F}$ is the collection of all joint CDFs, and $\mathbb{E}_{F_{UV}}$ is the expectation of joint CDF. In practice, the CDFs are computed empirically from the corresponding histograms.

A.3.3 Data-driven Similarity Measure

In order to formulate a data-driven similarity measure between the source and the target tasks, we train a convolutional neural network (CNN) regression model under two different cases and set a threshold empirically for its comparison with the obtained value.

Case A: In this case, the input of the CNN is composed of the images (first layer of Figure A.3) of different indoor wireless environments (total 250) and the output is the test MSE corresponding to TL obtained for each wireless environment when the baseline model is transferred and re-trained. The size of each image is 160 x 275 pixels. Each image is converted into a grayscale image before injecting it into the CNN.

Case B: In this case, the input of the CNN is the same, but the output is now both the test MSE corresponding to TL and the number of training epochs obtained for each wireless environment when the baseline model is transferred and re-trained.

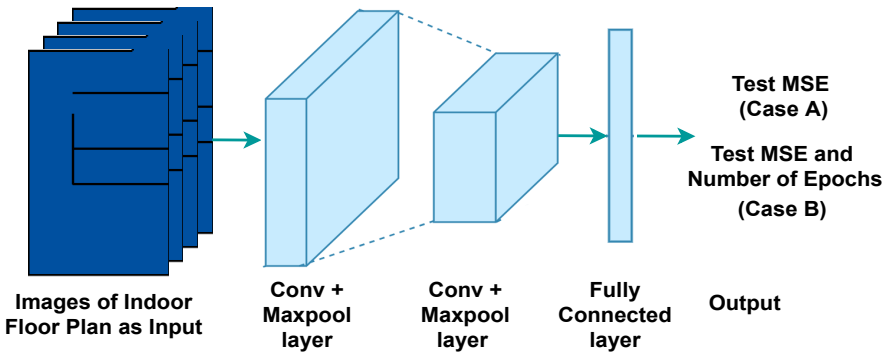


Figure A.3: Network structure for the CNN.

This case has a larger CNN model and it needs larger feature sets. This makes it computationally demanding. The network structure we consider in both cases is shown in Figure A.3.

Furthermore, to decide whether TL is beneficial or not between two tasks, in case A, we empirically set a threshold for the test MSE corresponding to TL as 1% (0.01) and compare it with the obtained test MSE using TL. If the value of obtained test MSE using TL is less than this threshold, then two tasks are recognized as similar and TL can be performed, otherwise, we assume that TL is not beneficial.

Similarly, in case B, we empirically set a threshold for the test MSE corresponding to TL as 1% (0.01) and the number of training epochs as 20, and compare the thresholds with both the obtained test MSE using TL and the obtained number of training epochs. If the obtained values of test MSE using TL and the number of training epochs are less than the thresholds, then two tasks are recognized as similar and TL can be performed, otherwise, it is assumed that TL is not beneficial.

A.4 Results and Discussions

The baseline DNN model used for TL is presented in Table A.2. It can be observed that the training of the baseline DNN model is satisfactory for the testing, that is, the test MSE is comparable to the training MSE, thus resulting in no over-fitting in the model.

Next, to check the potential usefulness of the WD, we first compute the WD between the floor image corresponding to the indoor wireless environment when the cube is positioned near the Tx (see Figure A.1(b)), and the different floor images (total 250) corresponding to indoor wireless environments when the cube is positioned at different locations. Similarly, we also compute the WD between the corresponding sampled radio map (power values) obtained when the cube is positioned near the Tx, and the sampled radio maps corresponding to the rest of the indoor wireless environments when the cube is positioned at different locations. However, if we measure the Pearson's correlation coefficient[2] (PCC), which is a widely adopted

[2]

$$\mathrm{PCC}(x,y) = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \tag{A.2}$$

Table A.2: The Baseline DNN model

Number of neurons in input layer	2
Number of hidden layers	2
Number of neurons in each hidden layer	16, 8
Hidden layers activation function	ReLU
Dropout after each hidden layer	0.20
Number of neurons in output layer	1
Output layer activation function	ReLU
Optimizer used	Adam
Loss function	MSE
Batch size	32
Train-Test split ratio	80:20
Test MSE	0.0102
Train MSE	0.0101

metric for measuring correlations [20], between both types of WDs (floor images and associated sampled radio maps) across the various considered environments, then it results in a value of 0.08, reflecting poor correlation. This implies that the WD is not an effective similarity measure for our TL problem in the context of indoor wireless radio maps, which justifies further the use of our data-driven similarity measure for TL.

Along these lines, Table A.3 and Table A.4 present the model learning for the CNN regression models under case A and case B, respectively. In case B, the settings to train the CNN regression model are the same as in case A except for the number of neurons in the output layer of the neural network is 16 due to aggregation of the number of training epochs as well as the test MSE obtained using TL as output features of the CNN. Both tables show that the CNN is trained appropriately in both cases, that is, the test MSE, is comparable to training MSE.

Following our designed TL decision strategy described earlier for 250 different indoor wireless environments, we now obtain a decision for each of the two CNN cases.

Table A.5 and Table A.6 present the TL decisions for case A and case B, respectively. Each table demonstrates training data needed to perform TL following the proposed TL strategy and the corresponding number of indoor wireless environments.

It can be noticed from Table A.5 that 246 out of 250 different indoor wireless environments satisfy our criteria and are recommended for TL, resulting in a training data requirement of 30-40% only. This leads to a TL success-rate of 98.4%. Similarly, Table A.6 shows that 217 out of 250 different indoor wireless environments satisfy our criteria and are recommended for TL, resulting in a training data requirement of 20-35% only. This leads to a TL success-rate of 86.8%. Both results signify that TL can be used effectively to estimate the radio map in the new indoor wireless

x = WD of image, y = WD of power values, n = number of environments.

Table A.3: The CNN model in Case A

Total indoor wireless environments	250
Image size of floor environment (width x height)	160 x 275
Number of convolutional layers	2
Number of filters in first and second convolutional layer	32,64
Filter size in first and second convolutional layer	3 x 3
Number of max pooling layers	2
Filter/pool size in each max pooling layer	2 x 2
Number of strides in first and second convolutional layer	2
Activation function in first and second convolutional layer	ReLU
Dropout after second max pooling layer	0.3
Number of neurons in the output layer of neural network	8
Activation function in the output layer of neural network	ReLU
Optimizer used	Adam
Loss function	MSE
Batch size	32
Train-Test split ratio	80:20
Test MSE	0.0411
Train MSE	0.0314

Table A.4: The CNN model in Case B

Number of neurons in the output layer of neural network	16
Test MSE	0.0423
Train MSE	0.0348
All other parameters are the same as in Table A.3 (Case A).	

Table A.5: TL Performance in Case A

Training data needed	Number of environments
40%	23
35%	206
30%	17

Table A.6: TL Performance in Case B

Training data needed	Number of environments
35%	31
30%	152
25%	31
20%	3

environment where a small number of sensor measurements are available.

Furthermore, in order to see how good and reliable our proposed model is, we

Table A.7: Transferability, F1-score, and Accuracy

	Case A	Case B
TP	166	144
TN	19	17
FP	52	66
FN	13	23
Transferability	0.927	0.862
F1-score	0.836	0.763
Accuracy	0.740	0.644

calculate transferability[3], F1-score, and accuracy [69] in each case. Transferability shows how good our model is at recommending a correct TL. F1-score is a measure of the test accuracy of the model. Accuracy measures the degree of veracity of the model.

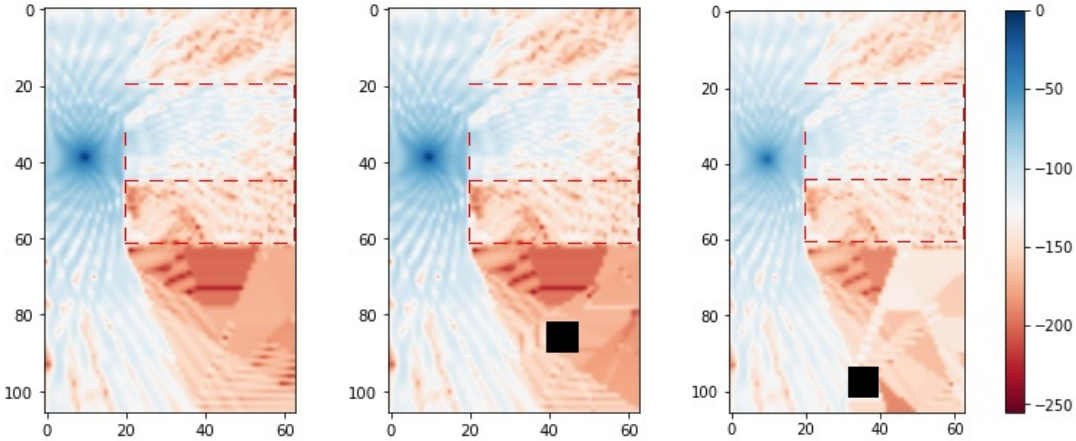
Table A.7 presents the true positive (TP), true negative (TN), false positive (FP), false negative (FN), transferability, F1-score, and accuracy for each case. It can be observed that the transferability and F1-score of our proposed model in case A are higher than in case B. Moreover, the accuracy of our proposed model in case A is 74%, which is higher than in case B. This means that for the different modified indoor wireless environments tested, the case A model provides a good performance. The reason for case B having lower accuracy is that the training epochs are not highly correlated with the test MSE corresponding to TL for the similarity measure. The analysis of results signifies that the data-driven similarity measure between the source and the target tasks is performing well and TL can be used effectively to estimate the radio map in the new indoor wireless environments requiring only a small number of sensor measurements.

As an illustration, Figure A.4 shows the radio maps corresponding to the original environment (no cube present) and two different indoor wireless environments (cube present). The cube in the radio maps is represented in black colour and the walls of the rooms in the radio maps are represented in dashed red colour. The radio maps in Figure A.4(b) and Figure A.4(c) represent the environments where only 20% and 30% of training data, respectively, are required when executing the operation of TL, as compared to the original radio map without executing the operation of TL in Figure A.4(a). The differences in the radio maps among the three environments can be easily visualized.

A.5 Conclusion

In this paper, we present a TL-based method to estimate the radio map for indoor wireless networks. Moreover, a data-driven similarity measure is developed to quantify the similarity between two tasks which is later used to decide whether

[3]Instead of the usual name of sensitivity in the context of medical diagnostic tests [69], we use the term "transferability" for the sake of clarity given the context of our work here.



(a) (b) (c)

Figure A.4: Radio map in (a) original indoor wireless environment, (b) environment where 20% of training data is required when executing the operation of TL, and (c) environment where 30% of training data is required when executing the operation of TL.

to recommend performing TL given only the image of the environment. A unique advantage of transfer learning is that it will reduce the number of training samples that are necessary for estimating the radio map in similar environments, saving a large amount of data acquisition requirements.

Appendix B

PAPER B

Title: A Data-driven Transfer Learning Method for Indoor Radio Map Estimation

Authors: **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano

Journal: IEEE TWC 2023 (Submitted)

A Data-driven Transfer Learning Method for Indoor Radio Map Estimation

R. Jaiswal, M. Elnourani, S. Deshmukh, B. Baltasar Lozano

Abstract: **Estimating accurate radio maps is important for various tasks in wireless communications, such as localization, resource allocation, and network planning, to name a few. Due to the changes in the propagation characteristics of the wireless environments, a radio map model learned under a particular wireless environment cannot be directly used in a new wireless environment. Moreover, learning a new model for every environment requires, in general, a large amount of data and is computationally demanding. In this work, we design an effective novel data-driven transfer learning method that transfers and fine-tunes a deep neural network (DNN)-based radio map model learned from an original indoor wireless environment to other indoor wireless environments with a certain level of similarity, allowing the radio map to be estimated with less amount of training data. As opposed to other widely used similarity measures that do not take into account the wireless propagation characteristics, we design a data-driven similarity measure that predicts the mean square error (MSE) and the amount of training data needed when learning a radio map in a new wireless environment. The proposed solution is corroborated by extensive simulations over a range of environments, achieving savings of approximately 40-90% in sensor measurement data.**

B.1 Introduction

To predict and improve the performance of future wireless communications, accurate estimation of radio maps is essential for efficient network operations. The radio map contains meaningful information about the wireless network and its propagation channels. Some of the applications of radio maps include resource allocation, localization, network design and planning to optimize the distribution of base stations, among others. For example, in the case of network planning, an area having inadequate power in some region (weak coverage) can be covered appropriately by adding a new base station in that region. In the most general sense, a radio map portrays an estimate of the power spectral density (PSD) over a required geographi-

cal area as a function of location, time, and frequency. Moreover, the radio map also depicts the various characteristics of the wireless environment over a certain region, which is governed by various factors, such as reflections, diffractions from large-scale objects and buildings, and scattering from small-scale objects, which result in fluctuations in the received radio signals. In this work, for the sake of simplicity in the explanation of our design methodology, we focus on data-driven radio map estimation for indoor wireless environments. However, the general framework can also be applied to estimate radio maps in outdoor wireless environments.

Estimating an accurate radio map for each new wireless environment using the data-driven method, in general, requires a huge amount of sensing data. However, in practice, there is data scarcity, that is, lack of data as it is difficult and expensive to obtain measurements in each new wireless environment. Moreover, collecting a large amount of sensing data for each new wireless environment and learning the corresponding radio map model each time is time-consuming. A good radio map that has been learnt in a particular wireless environment, may not be appropriate for another different wireless environment because the wireless propagation characteristics change. On the other hand, notice that by knowing the strength of signal or received power in one wireless environment (source environment) at several spatial locations (sampled radio map), one can make smart utilization of a pre-developed (learned) radio map model from the radio map samples in a source environment to estimate the radio map in another new environment (target environment) by using the concept of transfer learning (TL) [19], with the goal of reducing the need of new measurements at the new wireless environment.

The use of knowledge acquired in the source environment to assist the learning task in the target environment is referred to as transfer learning [19]. Hence, TL can handle the data scarcity issue in the target environment. In this regard, one needs to answer three main questions while performing the operation of TL, namely *"what to transfer"*, *"how to transfer"*, and *"when to transfer"*. In the context of radio map estimation, one may have configured or set up, for example, the wireless system in one environment (source) but need to perform deployment in another environment (target). Regarding *"what to transfer"*, the specific TL approach is referred to as *transductive transfer* [19] in which the baseline model trained under the source environment is transferred to a similar target environment. Similarly, for *"how to transfer"*, the approach is *parameter transfer* in which the parameters from the trained baseline model under the source environment are transferred to similar target environments and fine-tuned with an additional amount of training samples. Our proposed scheme also estimates the amount of additional training samples required. Moreover, the *"when to transfer"* is associated with the similarity measure between the source and the target environment, that is, the transfer is performed when both environments are sufficiently similar. Our goal in this paper is to design a data-driven TL method to be able to learn an accurate radio map solution in the target environment by exploiting the knowledge learned from the source model instead of learning a solution directly based only on measurements from the new environment. In this work, we address all these three questions pertaining to TL for

radio map estimation. Moreover, we also target identifying the amount of training data required for the new wireless environment to achieve a certain accuracy in the estimated radio map. Notice that even though we focus in this paper on power radio maps, our work can be also generalized to channel gain maps for every pair of relative transmitter-receiver `Tx-Rx` positions.

B.2 Related Works and Motivation

It is worth noting that there have been a number of works that have successfully applied TL for various tasks in wireless networks. For capturing traffic pattern diversity in the cellular data of distinct cities, the model in [20] proposes a spatial-temporal cross-domain neural network (STCNet). Model-based TL, where the complete source model is transferred to the target environment, is employed for the similarities between distinct types of cellular traffic across distinct cities. For improving the robustness of deep neural network (DNN) based spectrum sensing in a cognitive radio scenario with the assumption that the data collected under distinct characteristics belong to distinct but related distributions, TL is used in [21]. For tackling an NP-hard mixed-integer nonlinear programming problem of resource allocation, a TL method via self-imitation is proposed in [32]. In [22], downlink channel state information (CSI) prediction from uplink CSI using direct model-based TL, is proposed for frequency division duplexing (FDD) in a massive multiple-input multiple-output (MIMO) framework. In [23], wireless fingerprinting localization uses a TL method incorporating Wasserstein distance [24]. A TL-based method incorporating the Kullback-Leibler (KL) divergence index as a similarity measure, is used for predicting the antenna tilt-dependent radio map in [25]. A two-phase TL generative adversarial network (TPTL-GAN) for estimating power spectrum maps for underlay cognitive radio networks, is proposed in [26]. Moreover, TL has been shown to save a good amount of training data for allocating channels and power in underlay D2D wireless communications [36]. For estimating the radio map in indoor wireless networks, a TL-based approach is used in [34]. Preliminary experiments for location-free radio map estimations are performed in [8]. However, the work in [8, 34] is limited to a single base station and a very simple wireless environment. Our work considers complex indoor wireless environments with multiple base stations and changes in the locations of the base stations. Our core objective is to save sensor measurement data for estimating accurately the radio map in the new wireless environment, by using TL.

Since users may experience different wireless environments $\mathcal{E}$, without TL, data collection is required for effective model learning at each given new wireless environment. In fact, in practice, in the case of a DNN-based model, typically, a large number of samples and training epochs are required to train a DNN, and thus training a new DNN for each new wireless environment may demand a substantial computational time and data acquisition cost. Given these, a solution which is particularly appealing is, if a model is well trained first in a given wireless environment, then by transferring that trained model and fine-tuning it in another similar but still

different wireless environment, can effectively generate an estimated radio map using a much smaller amount of samples, reducing significantly data acquisition cost.

Motivated by this, in this paper, we propose a transfer learning method to estimate radio maps in a new environment $\mathcal{E}$ where there is a scarcity of measurement samples. Our proposed method is that if we have an environment with an accurate radio map then by using TL, we can exploit this accurate radio map in estimating the radio map for a new wireless environment where there is data scarcity. The first challenge is how to quantify the similarity between two wireless environments since this similarity controls the TL operation. From a human cognizance perspective, one can visualize two wireless environments and intuitively judge their similarity. In principle, one could consider the images of the wireless environments and quantify this similarity by using existing techniques such as KAZE [54], scale-invariant feature transform (SIFT) [56], oriented FAST and rotated BRIEF (ORB) [57], binary robust invariant scalable keypoints (BRISK) [60], peak signal-to-noise ratio (PSNR) [64], structural similarity (SSIM) [64], and Wasserstein distance (WD) [51], among others. These similarity measures perform different feature extraction algorithms for finding the similarity between two images, however, they can not guarantee the similarity between their corresponding radio maps (see Section B.5.4.2). This incoherence between the similarity of wireless environment images and wireless environment radio maps is due to the dependence of radio maps on the location of transmitters, reflecting surfaces, diffracting edges and scattering objects which are not accommodated in the above currently used similarity measures. This motivates us to design a data-driven similarity measure (DDS) (see Section D.3.2), which can learn all these elements directly from the data.

In order to motivate further the necessity of designing a new data-driven similarity measure, we consider a simple toy example as shown in Figure B.1.

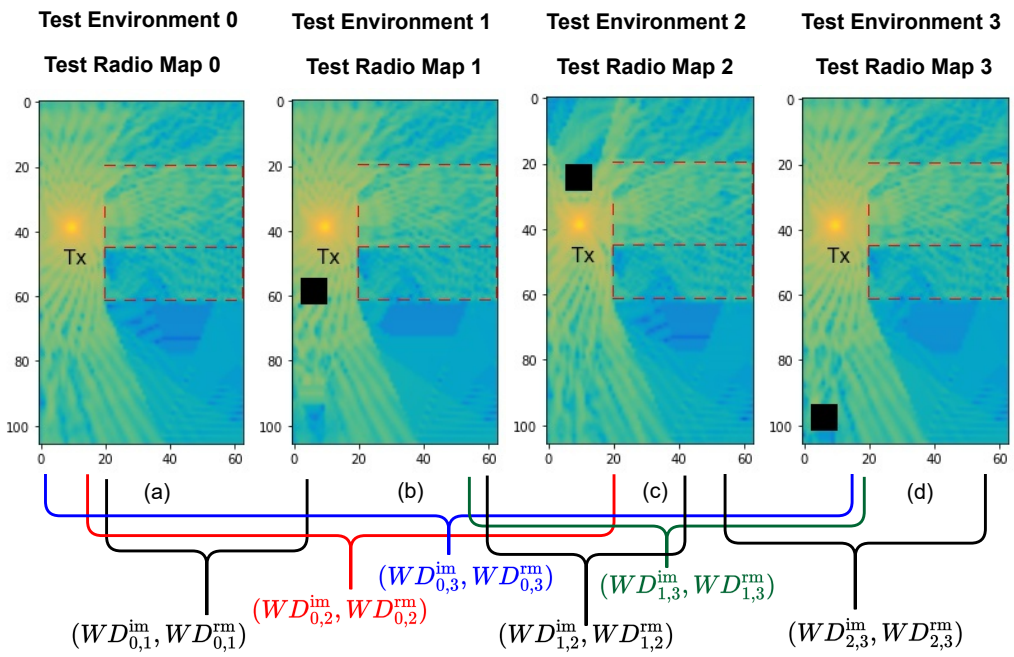


Figure B.1: Toy example illustrating different radio maps corresponding to different indoor wireless environments.

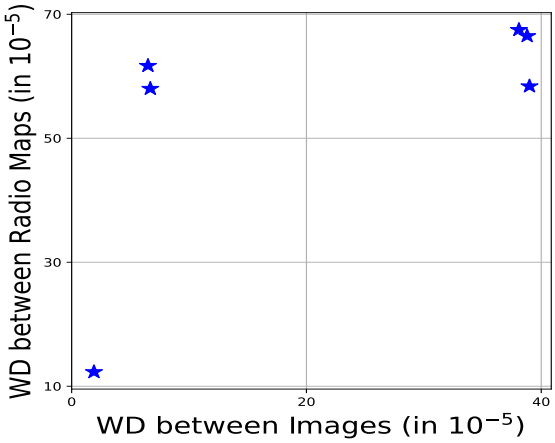


Figure B.2: Correlation between inter-image WD and inter-radio map WD.

Here, we first discuss the widely studied Wasserstein distance (WD) [51] measure[1] (see Section B.4.2). Let us consider an indoor floor plan comprising of two-rooms in which a `Tx` is placed at a fixed location. This is denoted as the reference or test environment 0 (TE 0). Then, we create three other different environments by incorporating a cube, made up of metal, at three different locations (see Section D.4 for more details), namely, Test Environment 1 (TE 1), Test Environment 2 (TE 2), and Test Environment 3 (TE 3), respectively. These test environments are created by incorporating a cube slightly below the `Tx`, slightly above the `Tx`, and at the left bottom corner. Further, the radio map (densely sampled) of each test environment from their corresponding power values, is obtained. The walls of both the rooms and the cube are represented in dashed red, and solid black colour, respectively, in each radio map. The illustration of each radio map corresponding to each environment $\mathcal{E}$, is shown in Figure B.1.

Let $\text{WD}^{\text{im}}_{i,j}$ denotes the WD between the images[2] (im) of TE i and TE j where i, $j \in \{0,1,2,3\}$, and $i \neq j$. For instance, $\text{WD}^{\text{im}}_{0,1}$ denotes the Wasserstein distance between the images of test environment 0 (TE 0) and test environment 1 (TE 1). Similarly, $\text{WD}^{\text{rm}}_{i,j}$ denotes the WD between radio maps (rm) of TE i and TE j. The WD between images of test environments and their corresponding radio maps are shown in Figure B.2.

We can see from Figure B.2 that there is a mismatch between the inter-image WD and the inter-radio map WD for the toy example, which gives us 6 points, given that WD is a symmetric distance. The values of WDs (shown by blue stars) show no correlation between the distances in the image domain (floor plan) and the radio maps domain. Ideally, for the WD to work appropriately as a similarity measure, both WDs should be properly aligned, as the cube location changes. This signifies that the WD is not a suitable similarity measure in our proposed TL problem. The unsuitability of WD for all our environments is shown via correlation heatmap in Section B.5.4.2. Therefore, we motivate to design the data-driven similarity

[1]The same conclusion holds for the other similarity measures.
[2]We use a coloured image of the floor plan.

measure (DDS) as described in Section B.4.2. To the best of our knowledge, the proposed scheme is the first of its kind in the literature which employs a data-driven transfer-learning method to estimate radio maps when there is a data scarcity of radio measurements. The main contributions of this paper are as follows:

- Design of an effective data-driven TL method that transfers and fine-tunes a DNN-based model for a radio map learned from a given source environment to another environment.
- Design of a data-driven similarity measure that is used to decide whether to perform the TL operation in a new target wireless environment, as compared to other widely used similarity measures.
- Prediction of the amount of training data needed to estimate the radio map in a different target environment, when performing the TL operation, using data-driven similarity measure.
- Analysis of the robustness of the proposed TL scheme under noisy environments when the locations of receivers are estimated and not perfectly known, and the corresponding power values are noisy. This scheme shows satisfactory performance comparable to the noiseless case.
- Experimentally measuring the temporal window in which fine-tuning the model is unnecessary.
- Extensive testing of the proposed TL scheme using simulated data from the Remcom simulator [33] (office area and cafe area). Results demonstrate that our proposed scheme can perform effective TL in the most tested environments while using a small amount of additional training data, and outperforms the state-of-the-art. Moreover, the proposed data-driven similarity measure can accurately recommend TL guaranteeing a satisfactory radio map estimation while reducing the number of necessary training samples.

B.3 Problem Formulation

This section presents the problem formulation of radio map estimation. The problem is defined as: "how to estimate the radio map of a given wireless environment with limited available sensor data by leveraging the information from other environments that have a certain similarity."

Let $\mathcal{E}_s$, and $\mathcal{E}_t$ denote the source and target wireless environment, respectively, as shown in Figure B.3. Next, let us assume that only a small amount of aggregate power values[3] are available in $\mathcal{E}_t$, as compared to the number of samples in $\mathcal{E}_s$. The radio map consists of aggregate power measurements $P(x_i)$ corresponding to a set of N locations $\{x_i\}_{i=1}^{N}$, where i is the index of the receiver.

[3]To obtain aggregate power, at each `Rx` location, multiple signals from the different `Txs` are summed with different phases.

Using a DNN, we create a predictive function that predicts the power values at any location in a radio map for a given wireless environment. More specifically, for the source environment $\mathcal{E}_s$, we first learn predictive function $f_s(\cdot)$ that is approximated by a DNN with weight parameters $\theta_s^\star$. Note that $\theta_s^\star$ is the optimal value of the weight parameters in the source environment $\mathcal{E}_s$. Similarly, for the target environment $\mathcal{E}_t$, one can also learn the predictive function $f_t(\cdot)$ that is approximated by a DNN with weight parameters $\theta_t^\star$. Let us also assume that $\mathcal{E}_s$ and $\mathcal{E}_t$ have some similarity in terms of wireless propagation characteristics.

Instead of learning $f_t(\cdot)$ directly without exploiting any previous knowledge, we perform first TL by exploiting the already learned function $f_s(\cdot)$. This allows for improving the learning of the target predictive function $f_t(\cdot)$ by using a smaller amount of additional measurement samples, as compared to not exploiting the previous information.

In the source environment $\mathcal{E}_s$, we learn the optimal weight parameters $\theta_s^\star$ for the power value prediction using a source DNN by minimizing the following loss function:

$$\theta_s^\star = \arg\min_{\theta\in\Theta}[Loss(P_s(x_i), \hat{P}_s^\theta(x_i))] \tag{B.1}$$

where Θ is the space of parameters of θ, $P_s(x_i)$ and $\hat{P}_s^\theta(x_i) = f_s(x_i, \theta)$ are the actual and predicted power values at i^{th} location, respectively, at the $\mathcal{E}_s$. We denote the loss function as $Loss(P_s(x_i), f_s(x_i, \theta))$, which is chosen to be the Mean Square Error (MSE), defined as:

$$\text{MSE} = \frac{\sum_{i=1}^{N_s}[P(x_i) - \hat{P}(x_i)]^2}{N_s} \tag{B.2}$$

where N_s is the number of receivers. Note that the number of receivers may be also different in each environment.

After this, we can learn a DNN model for the target environment $\mathcal{E}_t$ by first transferring and initializing the DNN model parameters as $\theta^{[0]} = \theta_s^\star$, and then solving with the usual training iterations to minimise the same loss function in the target environment $\mathcal{E}_t$. This training after performing TL, can be considered as "fine-tuning".

$$\theta_t^\star = \arg\min_{\theta\in\Theta}[Loss(P_t(x_i), \hat{P}_t^\theta(x_i))] \tag{B.3}$$

where $P_t(x_i)$ and $\hat{P}_t^\theta(x_i) = f_t(x_i, \theta)$ are the actual and predicted power values at i^{th} location of the receiver, respectively, at the $\mathcal{E}_t$. If the TL is effective, it is expected that the number of samples $\{P_t(x_i)\}$ to estimate an accurate radio map will be smaller as compared to the number of samples that is used to train the source $\mathcal{E}_s$ environment. Note that the loss function $Loss(\cdot, \cdot)$ is typically chosen to be the same for both source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments, namely MSE.

B.4 Proposed Radio Map Estimation Model

The overall architecture of the radio map estimation model considered in this work is shown in Figure B.3 and Figure B.4. It mainly consists of two phases: (i) the training

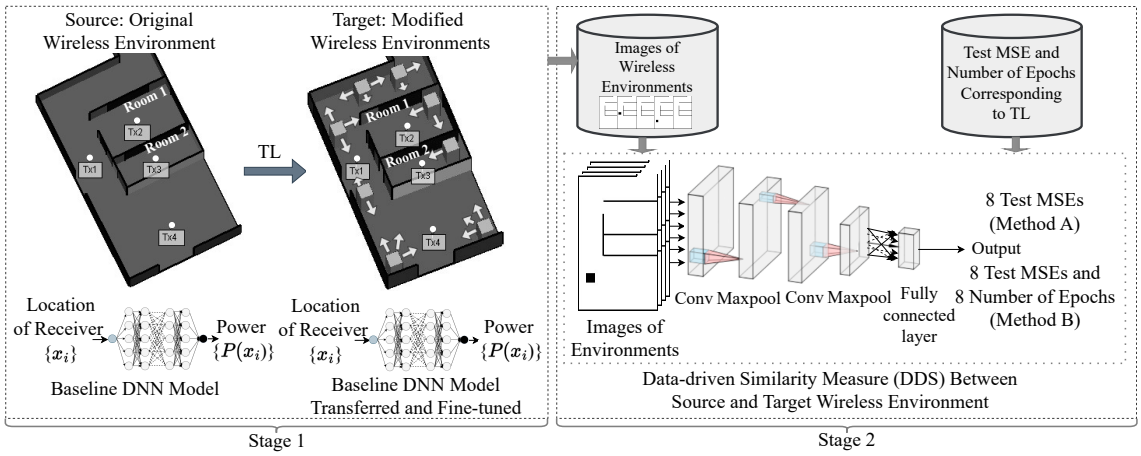


Figure B.3: Data-driven TL-based radio map estimation model (training phase).

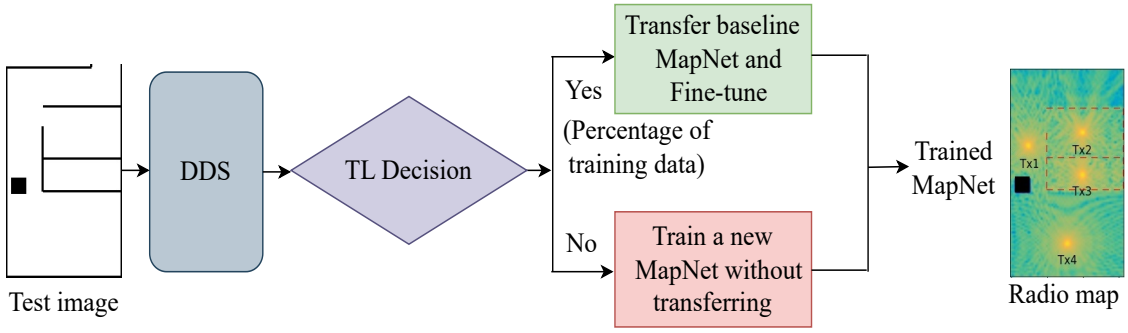


Figure B.4: Data-driven TL-based radio map estimation model (execution phase).

phase, and (ii) the execution phase. The training phase consists of two stages: (i) the development of the baseline DNN model in the source environment, followed by the possibility of transferring it to the target environment by first transferring the baseline model and then fine-tuning, and (ii) the establishment of similarity measure between the source and the target environment. During the execution phase, the image of a test wireless environment is given as input to the DDS. The DDS predicts different MSE values and the number of training epochs for the TL operation at different data splits (number of training samples). Next, in the TL decision block, these values are compared with the set MSE and training epochs thresholds to decide further whether to perform TL operation with a specific data split. If the thresholds are satisfied then the TL can be performed with that specific data split, else, TL will not be performed and a new baseline model *MapNet* needs to be trained from the very beginning.

An illustration of a TL system design to estimate the radio map from the source to the target environment is shown in Figure B.5. The target environment in Figure B.5(b) depicts one of the environments having four `Txs` and three objects represented by three cubes[4]. The details of generating different target environments for TL are discussed in Section B.5.1.

[4]The choice of cubes is done for the sake of simplicity without loss of generality. Similar results and conclusions are obtained for more complex objects as shown in later experiments.

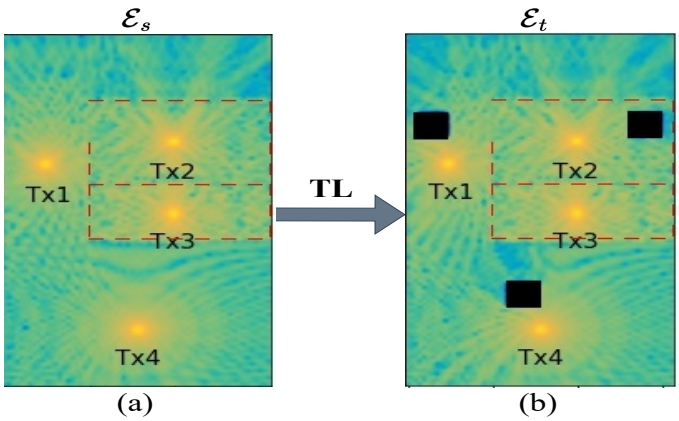


Figure B.5: Illustration of radio maps: (a) source (original), and (b) target environment with `4 Txs` and 3 objects represented by 3 cubes.

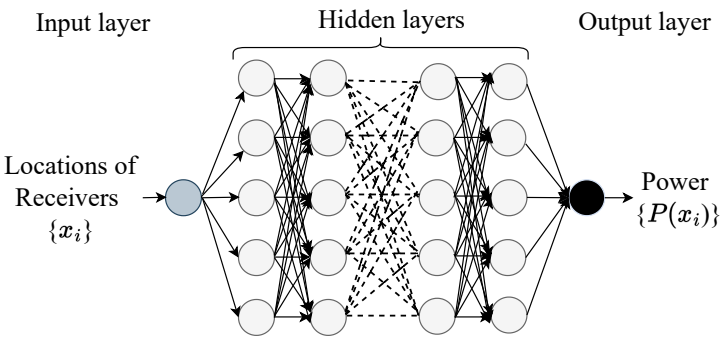


Figure B.6: MapNet: Baseline DNN model.

B.4.1 Baseline DNN Model and Transfer Learning Approach

We first design a feed-forward DNN [73] model with the power values obtained at multiple locations of receivers under the original environment (see Figure B.5(a)). For this purpose, we train a fully-connected DNN model as shown in Figure B.6 and refer to it as *MapNet* which is our baseline model. A high-level description of the flow of the baseline DNN *MapNet* in the form of pseudo-code is presented in **Algorithm 5**. Our model should map the location of receivers $\{x_i\}$ to the corresponding power values $\{P(x_i)\}$. We choose a DNN-based model since DNNs are universal function approximators. Moreover, DNN can learn high-level features from the data, which do not require domain expertise or manual feature extraction as compared to other machine learning techniques [49]. Moreover, DNN models are highly adaptable, that is, they can be fine-tuned to new tasks with an extra amount of data by leveraging information from the previous tasks, by means of transfer learning.

Then, given the baseline model *MapNet* shown in Figure B.6, we first decide whether to perform TL operation based on the similarity measure. However, since our similarity measure is data-driven, a sufficient amount of data obtained from performing TL operation is also necessary to learn this similarity measure. To this end, we transfer the *MapNet* to each target environment $\mathcal{E}_t$ individually, followed up by fine-tuning using a small amount of additional training data. A high-level

Algorithm 5: Baseline DNN Model *MapNet*

Input: $D := \{x_i, P(x_i)\}$: dataset of n locations of `Rxs` x_i, and power values $P(x_i)$
B: Batch size, α: Learning rate, N_{EP}: Number of epochs
Output: θ: Trained DNN model parameters, MSE: MSE of trained DNN based on test data
Data-Splitting: Splitting data into training D_{train} and test D_{test}

1: `Training stage`
2: Randomly initialize the network parameters
3: **for** k = 1,...,N_{EP} **do**
4: **for** j = 1,...,N_B = ceil($|D_{\text{train}}|/B$) **do**
5: Randomly select B training samples from training data D_{train} as the training batch
6: Update DNN parameters θ with learning rate α to minimize $Loss = \text{MSE} = \frac{\sum [P(x_i) - \hat{P}^{\theta}(x_i)]^2}{|B|}$
7: **end for**
8: **end for**
9: `Testing stage`
10: Initialize MSE: MSE $\leftarrow 0$
11: Use test data D_{test}
12: Predict power values $\hat{P}(x_i)$ on given locations of `Rxs` $\{x_i\}$
13: Calculate MSE as MSE $= \frac{\sum [P(x_i) - \hat{P}^{\theta}(x_i)]^2}{|D_{\text{test}}|}$
14: Save trained DNN model parameters θ.

description of the flow of the TL method in the form of pseudo-code is presented in **Algorithm 6**. Generally speaking, the amount of training data required when performing the TL operation, depends on the similarity between the $\mathcal{E}_s$ and $\mathcal{E}_t$, that is, the more similar environments, the less additional training data is required. The required training data for the target environments vary from 5% to 40% of the amount of training data that is necessary for the baseline model *MapNet* of the source environment. This training data needed for fine-tuning the *MapNet* in the target environment may be referred to as "fine-tuning data". After training and fine-tuning the transferred baseline model *MapNet* for each new environment, we store the test MSE and the number of training epochs while satisfying an early stopping [74] convergence criterion, i.e., MSE is approximately stable for a certain number of consecutive iterations.

Next, let us assume first that our similarity measure recommends performing TL and we actually perform TL. We explain our data-driven similarity measure (DDS) in Section B.4.2, where we use the test MSE and the number of training epochs. We first provide a description of the possible similarity measures that could be used in TL.

Algorithm 6: TL Method

Input: $\mathcal{E}_s$: Source environment, $\mathcal{E}_t$: Target environment, $MapNet$: baseline DNN model $\theta_s^\star$, α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}: Number of epochs

Output: $MapNet$: DNN model $\theta_t^\star$ for target environment $\mathcal{E}_t$, MSE($\mathcal{E}_t$): MSE of transferred model based on the test data, $N_{EP}(\mathcal{E}_t)$: Number of training epochs for the transferred model

Data-Splitting: Splitting data into training D^t_{train} and test D^t_{test} with splitting ratio r

1: `Fine-Tuning and Testing stage`
2: Initialize MSE($\mathcal{E}_t$): MSE($\mathcal{E}_t$) $\leftarrow 0$
3: Initialize $N_{EP}(\mathcal{E}_t)$: $N_{EP}(\mathcal{E}_t) \leftarrow 0$
4: `Fine-Tuning stage`
5: **for** k = 1,...,N_{EP} **do**
6: **for** j = 1,...,N_B = ceil($|D^t_{\text{train}}|/B$) **do**
7: Randomly select B training samples from training data D^t_{train} as the training batch
8: Update DNN parameters θ with learning rate α to minimize $Loss = \text{MSE} = \frac{\sum[P(x_i)-\hat{P}^\theta(x_i)]^2}{|B|}$
9: **end for**
10: Check early stopping criterion and stop when satisfied
11: **end for**
12: Save number of training epochs N_{EP}
13: `Testing stage`
14: Predict power values $\hat{P}(x_i)$ on given locations of `Rxs` $\{x_i\}$
15: Calculate MSE($\mathcal{E}_t$) $= \frac{\sum[P(x_i)-\hat{P}^\theta(x_i)]^2}{|D^t_{\text{test}}|}$

B.4.2 Similarity Measures

A crucial element of TL is that one needs to have a measure of similarity between two wireless environments because similarity affects the effectiveness of the TL operation. The more similar the source and the target environments are, the more features can be shared between both environments, resulting in better transferability between both environments and vice-versa. To this end, one can consider different traditional similarity measures, such as KAZE [54], SIFT [56], ORB [57], BRISK [60], PSNR [64], SSIM [64], and WD [51]. Since these similarity measures can be applied to any type of image representing source environment $\mathcal{E}_s$, and they do not consider the radio environment characteristics and propagation properties, these similarity measures may not be applicable, in general, for TL of radio map models. If one uses an inaccurate similarity measure then the TL decision might be incorrect. This may result in one of two cases; (i) performing TL when there is no benefit of TL, resulting in a worse estimation of radio maps, and (ii) training the model from the very beginning when TL could have been performed, result-

ing in requiring a large amount of training data and more computation resources. This motivates us to design a data-driven similarity measure that learns the wireless environment characteristic from the data, which is the second component of our proposed scheme, as shown in Figure B.3.

Next, we first introduce the WD, which is one of the widely used similarity measures, and then the designed DDS that we propose for the TL-based radio map estimation problem.

Wasserstein distance (WD) [51] is a popular metric to calculate the distance between two probability distributions. It has been widely used in TL due to its symmetry, smooth gradients, and good numerical results [52]. It is based on the minimal cost that is incurred to transform one distribution (e.g., represented by an image histogram, in our case, the image of environment $\mathcal{E}$) into another one. It is also called earth mover's distance or transportation distance and it is an outcome of the optimal transport theory [53].

In the context of TL, for a random variable[5] in the source $\mathcal{E}_s$ and target environments $\mathcal{E}_t$, denoted by $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$, with distribution functions $f_{X_{\mathcal{E}_s}}(x_{\mathcal{E}_s})$ and $f_{X_{\mathcal{E}_t}}(x_{\mathcal{E}_t})$, and joint distribution function $f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(x_{\mathcal{E}_s}, x_{\mathcal{E}_t})$, the WD between $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$ is defined as [51]:

$$\begin{aligned} W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) &= \inf_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}} \in \mathcal{F}} \int_{\mathcal{X}_{\mathcal{E}_s}\mathcal{X}_{\mathcal{E}_t}} |X_{\mathcal{E}_s} - X_{\mathcal{E}_t}| \, df_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(x_{\mathcal{E}_s}, x_{\mathcal{E}_t}) \\ &= \inf_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}} \in \mathcal{F}} \mathbb{E}_{f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}} |X_{\mathcal{E}_s} - X_{\mathcal{E}_t}| \end{aligned} \tag{B.4}$$

where $\mathcal{F}$ is the collection of all possible joint distributions. $f_{X_{\mathcal{E}_s}}$ and $f_{X_{\mathcal{E}_t}}$ are the marginal distributions. $df_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}(X_{\mathcal{E}_s}, X_{\mathcal{E}_t})$ is the derivative of joint distribution of variables $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$. $\mathbb{E}_{f_{X_{\mathcal{E}_s}X_{WE_t}}}$ is the expectation operation with respect to $f_{X_{\mathcal{E}_s}X_{\mathcal{E}_t}}$. Notice that WD is symmetric, that is, $W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) = W(X_{\mathcal{E}_t}, X_{\mathcal{E}_s})$.

Equivalently, it is shown in [51] that equation (B.4) can be re-written as:

$$W(X_{\mathcal{E}_s}, X_{\mathcal{E}_t}) = \int_{[0,1]} |f^{-1}_{X_{\mathcal{E}_s}}(X) - f^{-1}_{X_{\mathcal{E}_t}}(X)| \, dX \tag{B.5}$$

where, $f^{-1}_{X_{\mathcal{E}_s}}(X)$, and $f^{-1}_{X_{\mathcal{E}_t}}(X)$ are the inverse distribution functions in the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments, respectively.

Given a sufficiently large number of realizations (samples) of variables $X_{\mathcal{E}_s}$ and $X_{\mathcal{E}_t}$, (B.5) can be numerically estimated.

For the sake of simplicity, let us consider two images corresponding to two different wireless environments obtained by two different positions of an object such as a cube (see Figure B.1). We can calculate the WD between the floor image corresponding to the $\mathcal{E}_s$ when the cube is positioned at one location, and the different floor images corresponding to the remaining $\mathcal{E}_t$ when the cube is positioned at different locations. In the same manner, we can also calculate the WD between the corresponding sampled discretized radio map (power values) obtained when the

[5]In our case, one of these random variables models the pixel value distributions of the images representing the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environment.

cube is positioned at one location, and the sampled radio map corresponding to another environment where the cube is positioned at a different location.

Intuitively speaking, an effective similarity measure for our problem should have the property that it keeps a correlation between $\mathrm{WD}(X_{\mathcal{E}_s}, Y_{\mathcal{E}_t})$ and $\mathrm{WD}(\mathrm{RM}_{\mathcal{E}_s}, \mathrm{RM}_{\mathcal{E}_t})$. To see this, we can compute the Pearson's correlation coefficient (PCC) [20], which is a widely adopted metric for measuring correlations. For a given n pairs of input-output data
$\{(\mathrm{WD}_1^{\mathcal{E}}, \mathrm{WD}_1^{\mathrm{RM}}), (\mathrm{WD}_2^{\mathcal{E}}, \mathrm{WD}_2^{\mathrm{RM}}), ..., (\mathrm{WD}_n^{\mathcal{E}}, \mathrm{WD}_n^{\mathrm{RM}})\}$, the PCC is defined as [20]:

$$PCC(\mathrm{WD}_i^{\mathcal{E}}, \mathrm{WD}_i^{\mathrm{RM}}) = \frac{\sum_{i=1}^{n}(\mathrm{WD}_i^{\mathcal{E}} - \mu_{\mathrm{WD}_i^{\mathcal{E}}})(\mathrm{WD}_i^{\mathrm{RM}} - \mu_{\mathrm{WD}_i^{\mathrm{RM}}})}{\sqrt{\sum_{i=1}^{n}(\mathrm{WD}_i^{\mathcal{E}} - \mu_{\mathrm{WD}_i^{\mathcal{E}}})^2}\sqrt{\sum_{i=1}^{n}(\mathrm{WD}_i^{\mathrm{RM}} - \mu_{\mathrm{WD}_i^{\mathrm{RM}}})^2}} \quad \text{(B.6)}$$

where $\mathrm{WD}_i^{\mathcal{E}}$ and $\mathrm{WD}_i^{\mathrm{RM}}$ are different values/samples of WDs corresponding to different environments. $\mu_{\mathrm{WD}_i^{\mathcal{E}}}$ and $\mu_{\mathrm{WD}_i^{\mathrm{RM}}}$ are their corresponding means. Here, n is the number of environments.

Data-driven Similarity Measure (DDS): In the context of radio map estimation, classical similarity measures (e.g., WD) do not perform well, as shown numerically in Figure B.1 under Section B.1. Moreover, our experimental results in Section B.5.4.2 also confirm that other similarity measures including WD do not perform well. Thus, we propose formulating a DDS tailored to the TL-based radio map estimation problem. To this end, we provide a DDS that incorporates explicitly the TL-operation effect in terms of the MSE and/or the number of training epochs that are required to achieve a certain accuracy (MSE) of the radio map in the target environments, along with the images of environments. We also train this DDS model to predict TL accuracy (MSE) and training epochs in order to capture all the variations of the environments and their impacts on both the radio map estimation and the TL operation.

To formulate our DDS, $DDS_{\mathcal{E}_s}(\mathrm{Im}_{\mathcal{E}_t})$, between the source environment $\mathcal{E}_s$ and the target environment $\mathcal{E}_t$, as shown in our proposed scheme (Figure B.3), we train a convolutional neural network (CNN) [50] regression model under a certain loss function. A high-level description of DDS in the form of pseudo-code is presented in **Algorithm 7**. We consider two choices for the loss function. Loss function A $L_A = |\mathrm{MSE}(\mathcal{E}_t) - \widehat{\mathrm{MSE}}_t|^2$ is the squared difference between the actual test MSE obtained previously from TL, denoted by $\mathrm{MSE}(\mathcal{E}_t)$, and the predicted test MSE, denoted by $\widehat{\mathrm{MSE}}_t$, which is obtained when performing the TL.

Alternatively, a different loss function can be used to control the sample complexity by considering also the number of epochs. Loss function B $L_B = ||[\mathrm{MSE}(\mathcal{E}_t) - \widehat{\mathrm{MSE}}_t, N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]||^2$ contains both the test MSE and the number of training epochs which specify the error in power values and the need for training time to perform TL, respectively. Note that $N_{EP}(\mathcal{E}_t)$, and $\widehat{N_{EP_t}}$ are actual and predicted TL training epochs, respectively. The test MSE and the number of training epochs are normalized separately to 0 and 1. Minimising this loss function provides us with the radio maps that have the least error and converge in the least number of iterations.

Algorithm 7: Data-driven Similarity Measure *SimNet*

Input: $D := \{\text{Im}_t, \text{MSE}_t, N_{EP_t}(\text{only for Method B})\}$: dataset comprising three-level images of target environments, MSE after TL for 8 different data splits (5% to 40%) (**Algorithm 6**), Number of training epochs required after TL for 8 different data splits (5% to 40%) (**Algorithm 6**), for each environment, respectively. MSE_{thr}: MSE threshold, $N_{EP_{thr}}$: Threshold for number of training epochs, α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}^{cnn}: Training epochs of *SimNet*

Output: $SimNet(\theta_{sim})$, $\widehat{\text{MSE}_t}$, $\widehat{N_{EPt}}$ are the predicted MSE and number of training epochs, respectively, for 8 different data splits (5% to 40%), TL decision: Yes/No, the required data split

Data-Splitting: Splitting data into training D_{train} and test D_{test} with splitting ratio r

1: `Training stage`
2: Use data from D_{train}
3: Normalize image values Im_t, MSE_t and N_{EP_t} to $[0, 1]$
4: Randomly initialize θ_{sim}
5: **for** k = 1,...,N_{EP}^{cnn} **do**
6: **for** j = 1,...,$N_B = \text{ceil}(|D_{\text{train}}|/B)$ **do**
7: Randomly select B training samples from D_{train} as the training batch
8: Update θ_{sim} with learning rate α to minimize *loss* given by the

$$\begin{aligned} \text{MSE}^{cnn} = L_A &= |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}_t}|^2 \quad \text{(Method A)} \\ = L_B &= |[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}_t}, \\ &\qquad N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]|^2 \quad \text{(Method B)} \end{aligned} \tag{B.7}$$

9: **end for**
10: **end for**
11: `Testing stage`
12: Initialize MSE^{cnn}: $\text{MSE}^{cnn} \leftarrow 0$
13: Use test data from D_{test}
14: Predict $\widehat{\text{MSE}_t}$ and $\widehat{N_{EPt}}$ for 8 different data splits (5% to 40%) on a given image Im_t
15: `TL decision`
16: **for** q = 1,...,$|D|$ **do**
17: **if** $\widehat{\text{MSE}_t} \leq \text{MSE}_{thr}$ (for Method A) **then**
18: TL decision ← Yes and output the corresponding data split, **else**, TL decision ← No
19: **if** $\widehat{\text{MSE}_t} \leq \text{MSE}_{thr}$ & $\widehat{N}_{EP_t} \leq N_{EP_{thr}}$ (for Method B) **then**
20: TL decision ← Yes and output the corresponding data split, **else**, TL decision ← No
21: **end if**
22: **end if**
23: **end for**

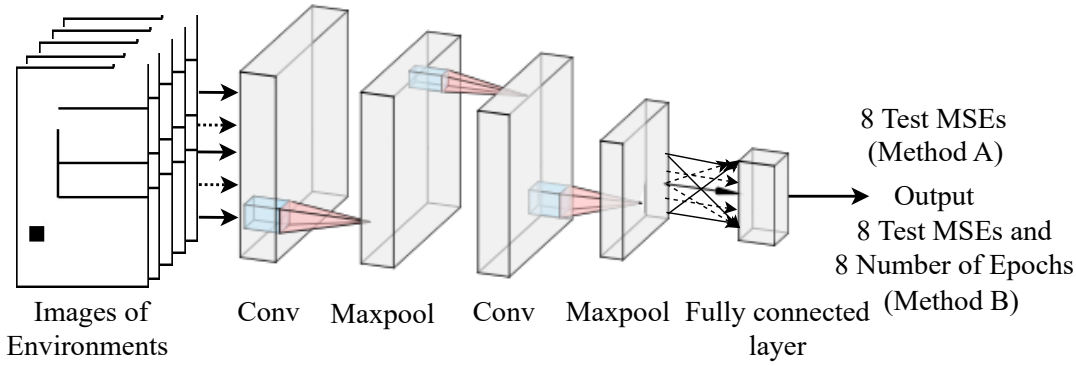


Figure B.7: CNN-based neural structure for the similarity measure.

Method A: This method comprises the CNN having the images (see Figure B.7) of distinct target environments $\mathcal{E}_t$ as an input, and eight sets of test MSEs, i.e., $MSE(\mathcal{E}_t)$ obtained after performing TL for each target environment when *MapNet* is transferred and re-trained at different data splits (5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%), as output. This method uses the loss function L_A. The corresponding learned CNN model in this case is referred to as $SimNet_A$. Each environment image has two parts. The first part is the floor image which is a three-level image in which free space is white (gray value 255), walls are assumed gray value 128, and objects (e.g., cubes) are black (gray value 0). These differences in intensity levels are useful to differentiate three elements (free space, walls, and objects). The second part is a black and white image of the `Tx` locations which is created by assigning a value of 255 (white) wherever the transmitters are located, and the rest pixel values are set as 0 (black).

Method B: This method comprises the CNN having the same input, however, the output is now the combination of eight sets of test MSEs, i.e., $MSE(\mathcal{E}_t)$ obtained after performing TL for each target environment with different data splits (5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%) and eight sets of $N_{Ep}(\mathcal{E}_t)$ of training epochs required for each environment when *MapNet* is transferred and re-trained. This method uses the loss function L_B. The corresponding learned CNN model in this case is referred to as $SimNet_B$.

Method A considers only the accuracy of the proposed scheme and the network trained under Method A is relatively smaller as compared to Method B. However, Method B considers both the accuracy and time complexity, resulting in a larger network. This makes it more computationally demanding. The choice of the Method depends on the application, i.e., how much information (accuracy and time complexity) is needed and how much computations can be afforded.

For any similarity measure, to obtain a similarity decision whether TL is beneficial (in our case, small MSE($\mathcal{E}_t$), and small $N_{EP}(\mathcal{E}_t)$) or not between two environments, we need to set a threshold. In the case of our DDS, under Method A, we can set a threshold for the test MSE (MSE$_{thr}$) and compare it with the predicted test MSE ($\widehat{MSE}(\mathcal{E}_t)$) obtained after TL. Similarly, under Method B, we can also set a threshold for both the predicted test MSE obtained after TL, i.e., MSE$_{thr}$ and the number of training epochs, i.e, $N_{EP_{thr}}$ and compare these thresholds with the

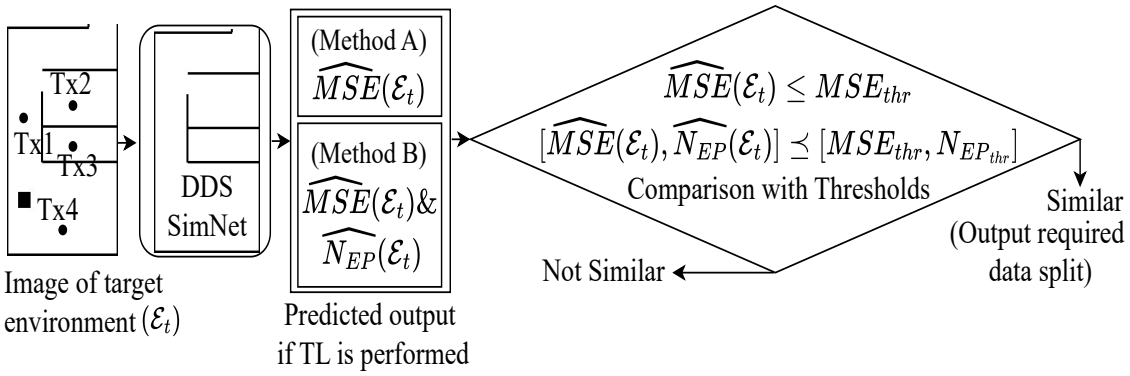


Figure B.8: Data-driven similarity decision between the target $\mathcal{E}_t$ and source $\mathcal{E}_s$ environments.[6]

corresponding predicted values after TL. If the predicted values are less than these thresholds, then two radio maps are recognized as similar and TL can be performed effectively, otherwise, we assume that TL is not effective. Figure B.8 illustrates the similarity decision taken between two environments under each method. Note that, we give an image of the floor plan of a new target wireless environment as input and then map it to the MSE of the TL operation (MSE and N_{EP} in Method B).

Notice that the choices of threshold values (MSE_{thr} and $N_{EP_{thr}}$) provide additional degrees of freedom on the proposed model. For Method A, a very high threshold value MSE_{thr} will result in a less accurate radio map estimation. However, if we choose a very low threshold value, then it will result in a better radio map estimation, but a lower recommendation rate[7]. Similarly, for Method B, higher threshold values MSE_{thr} and $N_{EP_{thr}}$ will result in both estimating a poorer radio map and having a longer convergence time. However, if we choose very low threshold values, then it will result in a better radio map estimation and a shorter convergence time, but the recommendation rate will be lower. Moreover, it should be noticed that the choice of these thresholds will also depend on the application scenario, that is, the complexity of the environment patterns. To select the value of these thresholds, we have to trade-off between the radio map quality, convergence time (Method B), and the recommendation rate of the TL.

B.5 Numerical Evaluation

B.5.1 Generation of Indoor Wireless Environments

This subsection presents the adopted procedure to obtain power measurements at multiple receiver locations corresponding to different types of environments, assuming a dense spatial sampling.

We use the standard high accuracy ray-tracing *X3D ray model* [41], computed using the software Remcom [33]. We first consider an environment of a single floor

[6]The symbol $\preceq$ indicates component-wise less than or equal operation.

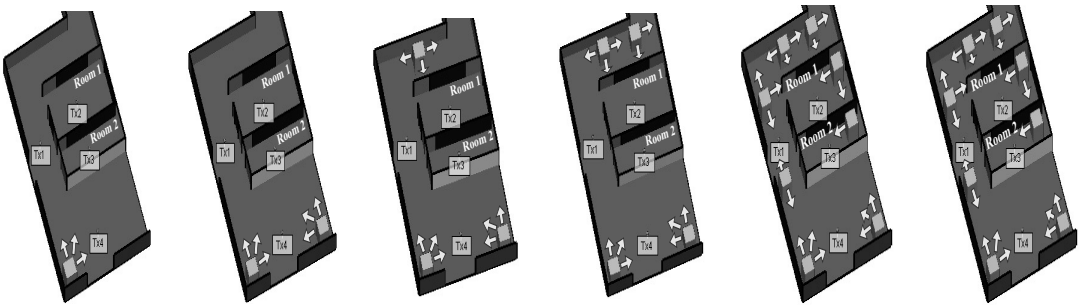
[7]The recommendation rate of TL is calculated as the number of environments for which the TL operation is recommended divided by the total number of environments.

Table B.1: Parameters used in Remcom for generating the data.

Waveform	Narrow-band Sinusoidal
Carrier frequency	900 MHz
Bandwidth	1.0 MHz
Antenna type	Omni-directional
`Tx1` location `Tx2` location `Tx3` location `Tx4` location	(1.5, 10.0, 1.3) m (6.0, 11.0, 1.3) m (6.0, 7.75, 1.3) m (4.75, 2.5, 1.3) m
`Tx` height	1.3 m
`Tx` power	27.73 dBm
Space between two `Rx` points	15 cm
Space between two `Rx` routes	15 cm
Ray tracing	X3D ray model
Ray spacing	0.2°
Number of reflections, transmissions, and diffractions	3, 2, 0
Volume of the object (single cube)	1 m^3
Number of `Rx` locations	6678
Floor dimension (width x length x height)	(9.5 x 20.6 x 2.88) m
Image size of floor (width x height)	(160 x 275) pixels

which comprises two rooms, for generating the radio map shown in Figure B.5(a). We place N_t transmitters `Txs` at different locations $\{x_i^t\}_{i=1}^{N_t=4}$ and then multiple receivers `Rxs` at uniformly spaced locations $\{x_j^r\}$ with 15 cm spacing to obtain the aggregate power values $\{P(x_j^r)\}$ at each `Rx` location. This results in (63 x 106) 6678 `Rx` locations and we generate the aggregated power values at these locations. Table B.1 summarizes the parameters used in Remcom for generating the data.

Next, to create different types of environments with 4 `Txs`, we place certain objects at different locations. For the sake of simplicity, we consider objects represented by a single solid cube block, placed in the environment. We denote the inclusion of one cube block within the original environment as *Type I* $\mathcal{E}$, two cube blocks as *Type II* $\mathcal{E}$, and so on. The inclusion of an increasing number of cube blocks makes the original environment more complex, providing a good target environment $\mathcal{E}_t$ to investigate the suitability of our proposed TL method in estimating the respective radio maps. We assume that the cube is made of metal having a volume of 1 m^3. The location of this object is changed horizontally, vertically, and diagonally, with an incremental spacing of 15 cm generating also different environments of each type. For each change in location, a new environment is created and the ray-tracing model provides the power values for that particular environment with the same `Rx` locations (expect that the `Rx` can not be on the top of the cubes). 1046 distinct environments are created by changing the location of the object in distinct directions and the image of each environment is saved as a 160 $\times$ 275-pixel three-level image in which free space is white (gray value 255), walls are assumed 128, and obstacles/objects



(a) (b) (c) (d) (e) (f)

Figure B.9: Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes, (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes), and (f) Type VI $\mathcal{E}$.

Table B.2: Summary of wireless environment $\mathcal{E}$.

Type of $\mathcal{E}$	Description of $\mathcal{E}$	Number of $\mathcal{E}$
Type I	`4 Txs`, 1 Cube	250
Type II	`4 Txs`, 2 Cubes	252
Type III	`4 Txs`, 3 Cubes	248
Type IV	`4 Txs`, 4 Cubes	254
Type V	`4 Txs`, 8 Cubes	42
Type VI	Changed layout	130
Total		1176

(e.g., cubes) are black (gray value 0).

Additionally, we create 130 different *Type VI* wireless environments that have the following changes: (a) changing the carrier frequency of the waveform from 900 MHz to 950 MHz, (b) changing the location of each `Tx` by 1 meter, (c) adding a wall 2 meter below to the Room 2, and (d) adding two objects in the floor plan and changing its locations horizontally, and vertically, with a spacing of 15 cm. As a result, the total number of wireless environments becomes 1176. The summary of different types of wireless environments with the respective number of environments is presented in Table B.2 and shown in Figure B.9. These images are later used to investigate the suitability of different similarity measures, as discussed in Section B.4.2, in particular, the WD and our DDS.

B.5.2 Performance Measures

We evaluate the performance of the proposed scheme based on the overall model accuracy and noise robustness.

B.5.2.1 Model Accuracy

To evaluate the performance of the proposed scheme, we calculate the transferability, F1-score, and accuracy [69] for *Type I* environment $\mathcal{E}$ and all types of wireless

environments grouped together (see Figure B.9 and Table B.2). We assume that on average, moving from the *Type I* environment $\mathcal{E}$ to all types of environments $\mathcal{E}$, the environments become more complex, resulting in less similarity between $\mathcal{E}_s$ and $\mathcal{E}_t$. The transferability presents how good the considered model is in recommending a correct TL for a new $\mathcal{E}_t$ (recommending a correct TL means that the actual TL operation is successful). The higher value of transferability means a better model. The F1-score is a measure of the test accuracy of the model, normalized to be between 0 and 1. The closer it is to 1, the better the model is. Accuracy measures the degree of veracity of the model. All these measures are defined in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) as [69]:

$$\text{Transferability} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{B.8}$$

$$\text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{B.9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{B.10}$$

where TP means $\mathcal{E}_s$ and $\mathcal{E}_t$ are sufficiently similar $\text{MSE}(\mathcal{E}_t) \leq MSE_{thr}$ (Method A) or $[\text{MSE}(\mathcal{E}_t), \text{N}_{EP}(\mathcal{E}_t)] \preceq [MSE_{thr}, N_{EP_{thr}}]$ (Method B) and the decision to transfer is taken $\widehat{\text{MSE}}_t \leq \text{MSE}_{thr}$ (Method A) or $[\widehat{\text{MSE}}_t, \widehat{N}_{EP_t}] \preceq [\text{MSE}_{thr}, N_{EP_{thr}}]$ (Method B). TN means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are not sufficiently similar and the decision of not to transfer is taken. FP means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are not sufficiently similar but the decision is still to transfer. FN means that $\mathcal{E}_s$ and $\mathcal{E}_t$ are sufficiently similar but the decision of not to transfer is taken. For better accuracy, FP and FN should be as small as possible.

B.5.2.2 Noise Robustness

To investigate the robustness of the proposed scheme towards the presence of noise, we investigate two cases of noise: (a) error in the `Rxs` locations (imperfect location estimations), and (b) noise in the power values. These two kinds of inaccuracies are prevalent in wireless networks due to the challenges in estimating accurate `Rxs` locations, and the measurement errors in the device, respectively. To this end, for case (a), we consider a practical scenario where the `Rxs` locations are estimated from the time of arrival[8] (ToA) features of the radio signals using the *trilateration* localization algorithm[9] [1]. The algorithm determines the approximate locations of `Rxs` based on the simultaneous ToAs from multiple `Txs` (at least 3 `Txs`) at known locations. Specifically, it finds the intersection of four circles in the 2D case or the intersection of four spheres in the 3D case, where a system of quadratic (non-linear) equations is involved and solved. For case (b), in order to reflect the inexact power measurements, we add Gaussian noise of zero mean and a certain variance to the power values in each environment. We assume a variance of 0.5dBm. This

[8]Time of arrival (ToA) is the time at which the radio signals from each `Tx` arrive at each of the `Rxs`.

[9]Notice that the particular choice of the algorithm is not important for the evaluation of our TL performance with respect to error locations.

procedure is followed for each $\mathcal{E}_t$ of each type (see Table B.2). Then, we transfer the baseline DNN model *MapNet* to each $\mathcal{E}_t$ and fine-tune it to obtain the test MSE, i.e., MSE($\mathcal{E}_t$), and the number $N_{EP}(\mathcal{E}_t)$ of training epochs corresponding to TL for each $\mathcal{E}_t$. Moreover, we also plot the test MSE corresponding to TL with the percentage of training data after TL, for both the noiseless and noisy environments.

B.5.3 System Setup

All algorithms are programmed in Python 3.8.5. We use the Keras framework on top of TensorFlow 2.2.0 on a Windows 10 laptop having Intel Core i5-5200U CPU 8th generation with 2.20 GHz, Intel ultra high definition (UHD) Graphics 620, and 16 GB of memory. Next, we present different parameters and hyper-parameters used for the *MapNet* and *SimNet*.

For developing an efficient baseline DNN model *MapNet* for TL and a data-driven similarity measure by training a CNN model *SimNet*, we experiment with various choices of parameters and hyper-parameters. We normalize each type of data separately. For example, locations of `Rxs` between 0 and 20 meters are normalized to 0 and 1, power values of `Rxs` between -55 and 25 dBm are normalized to 0 and 1, and images of floor plans between 0 and 255 are normalized to 0 and 1. Normalizing the data is commonly used in machine learning for faster training of models, as shown in [75]. We train each model with a distinct number of hidden layers, each having a variable number of neurons. The rectified linear unit (ReLU) [76] is employed as the activation function to the hidden layers and the output layer of each model since we only need positive values. The weights of each model are initialized randomly. The entire training data is employed to optimize the weights of each model. The absolute MSE [64], which is the standard metric employed in regression tasks, is used as the loss function, since the goal is to minimize the difference between the real and the predicted power values. Adam [77] optimizer with a learning rate of 0.001 is employed for the stochastic optimization of each model. We also perform grid search and random search [47] in order to find the best hyper-parameters. Additionally, 3-fold cross-validation [78] is employed for evaluating the training and the testing accuracy of each model. The various choices of parameters and hyper-parameters for the baseline DNN *MapNet* and CNN $SimNet_A$ trained under Method A and CNN $SimNet_B$ trained under Method B are outlined in Table B.3, and Table B.4, respectively. With these parameters, the models achieve good performance in terms of minimizing the MSE during training with moderately low generalization error for the test dataset.

B.5.4 Results and Discussions

This subsection discusses the results to showcase the effectiveness of the proposed scheme.

Table B.3: Parameters and hyper-parameters for $MapNet$

Number of neurons in input layer	2
Number of hidden layers	6
Number of neurons in each hidden layer	512, 256, 128, 64, 32, 16
Hidden layers activation function	ReLU
Number of neurons in output layer	1
Output layer activation function	ReLU
Optimizer with learning rate	Adam (0.001)
Loss function	MSE
Batch size	8
Training-Testing split ratio	70:30

Table B.4: Parameters and hyper-parameters for $SimNet_A$ and $SimNet_B$

	$SimNet_A$	$SimNet_B$
Floor image size (width x height)	160 x 275	
Number of convolutional layers	2	
Number of filters in first and second convolutional layer	32, 16	
Filter size in first and second convolutional layer	3 x 3	
Number of max pooling layers	2	
Filter/pool size in each max pooling layer	2 x 2	
Number of strides in first and second convolutional layer	2	
Activation function in first and second convolutional layer	ReLU	
Number of neurons in the output layer of neural network	8	16
Activation function in the output layer of neural network	ReLU	
Optimizer with learning rate	Adam (0.001)	
Loss function	MSE	
Batch size	32	
Training-Testing split ratio	80:20	
Number of iterations for early stopping	5	
Threshold MSE_{thr} for MSE	0.005	
Threshold $N_{EP_{thr}}$ for number of training epochs	-	12

B.5.4.1 Baseline DNN Model Analysis

In order to obtain the best baseline DNN model (see Figure B.6) required for performing the TL operation in the new environment $\mathcal{E}_t$, we perform grid search [47] with a variable number of hidden layers (5-10), number of neurons for each layer (8-1024), and learning rate range (0.0001-0.01) for the set of data described in Section D.4 and the best baseline model $MapNet$ (see Table B.3) results in a training and test MSE of 0.0024, and 0.0026, respectively. This reflects that the test MSE is comparable (slightly large) to the training MSE, that is, the baseline DNN model $MapNet$ is indeed an accurate model for the training data and also generalizes well

Figure B.10: Correlation analysis between different types of similarity measures.

for the test data.

B.5.4.2 Experimental Analysis of TL

For TL, we have different types of target environments $\mathcal{E}_t$, for instance, *Type I* (see Table B.2). Firstly, the baseline model $MapNet$ is transferred to all types of environments $\mathcal{E}_t$ (total 1176 environments) and then the individual DNN models are fine-tuned to obtain the test MSE, i.e., MSE($\mathcal{E}_t$) and the number $N_{EP}(\mathcal{E}_t)$ of training epochs corresponding to TL.

Next, to investigate experimentally the potential usefulness of other different similarity measures, as discussed in Section B.4.2, we obtain the similarity between the environment $\mathcal{E}_s$ and all the environments $\mathcal{E}_t$ (total 1176 environments) and plot the correlation heatmap[10], as shown in Figure B.10. Here, WD is the Wasserstein distance between different floor images of environments, where the source environment $\mathcal{E}_s$ assumes that the cube is present at the left-hand side bottom corner of the floor image and the target environment assumes that the cube is located at another position. In this particular experiment, for the sake of simplicity, we only use one cube to have simpler interpretations and intuitions.

It can be observed from Figure B.10 that there is a very poor correlation between all these similarity measures with respect to the test MSE, i.e., MSE($\mathcal{E}_t$) obtained after performing TL. This means that the similarity measures KAZE, SIFT, ORB and BRISK, which usually are applied to detect certain features (e.g., blobs and corners) in the neighbouring pixels, fail here in the case of radio maps, resulting in a mismatch between the images of the source and the target environments. Similarly, the reason for the failure of both SSIM and PSNR[11] is that the differences in the pixel values of the source and the target environment images are unable to provide the measure of closeness, resulting in a poor mismatch. In the same manner, the WD is unable to identify the pattern of the distribution (represented by the histograms) of the power values in a radio map when a cube is moved from one location of the

[10] A correlation heatmap is a graphical representation of the correlation between different variables (here similarity) in the coloured form.

[11] PSNR is a similarity measure defined in [62, 63].

Table B.5: TL performance for Type I environment and all types of environments grouped together.

	(a)		(b)	
	Type I		All Types	
	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
Necessary training data	Number of environments	Number of environments	Number of environments	Number of environments
5%	3	-	-	2
10%	20	7	8	27
15%	25	2	52	33
20%	9	3	51	124
25%	193	148	193	288
30%	-	31	872	196
35%	-	31	-	56
40%	-	20	-	150
TL Recommendation rate (in %)	250/250 (100)	242/250 (96.8)	1176/1176 (100)	876/1176 (74.48)

$\mathcal{E}_s$ to the new location of the $\mathcal{E}_t$. This implies that all these similarity measures including the WD are not effective similarity measures for our TL problem in the context of radio map estimation. Hence, we design our DDS based on CNN.

For comparison, following the same TL strategy, as discussed in Section B.4.1, we also train the CNN regression models for the DDS (see Section B.4.2). Specifically, for illustration, the baseline DNN model $MapNet$ is transferred to a Type I $\mathcal{E}_t$ (`4Txs` and 1 cube) first and then fine-tuned individually for each $\mathcal{E}_t$ to obtain the MSE($\mathcal{E}_t$) and the number $N_{EP}(\mathcal{E}_t)$ of training epochs corresponding to TL that are required to further train the CNN regression models $SimNet_A$ and $SimNet_B$ for the DDS. Finally, the same operation is performed with all $\mathcal{E}_t$ grouped together (a total of 1176 environments) for the DDS. Notice that we consider six different seeds while performing the TL operation to obtain the average value of the result, that is, MSE($\mathcal{E}_t$) and the number $N_{EP}(\mathcal{E}_t)$ of training epochs corresponding to TL.

TL Analysis for Type I: Here, we consider 250 different environments which correspond to 250 different floor images (see Table B.2). While training the CNN regression model $SimNet_A$ under Method A (see Figure B.7 and Table B.4), we obtain training MSE and test MSE of 0.0912, and 0.1017 respectively. This reflects that the test MSE is comparable to the training MSE, implying that the $SimNet_A$ is trained appropriately. Similarly, while training the CNN regression model $SimNet_B$ under Method B (see Figure B.7 and Table B.4), we obtain training and test MSE of 0.0163, and 0.0304 respectively. This also reflects that the $SimNet_B$ is trained appropriately.

Next, Table B.5 (a) presents the TL decision for both methods under Type I

environments. It presents the number of environments that satisfy the proposed TL strategy with the necessary training data required for performing TL operation. It can be observed from Table B.5 (a) that for Method A, all 250 distinct environments satisfy our criteria and are recommended for TL, with a training data requirement in the range of 5-25% as compared to 70% when not using TL. This implies that our proposed method is able to achieve savings of approximately 65-90% in sensor measurement data while achieving the same accuracy. This leads to a TL recommendation rate of 100%. Similarly, for Method B, 242 out of 250 distinct environments satisfy our criteria and are recommended for TL, with a training data requirement in the range of 10-40% as compared to 70% when not using TL. This implies that our proposed method is able to achieve savings of approximately 40-85% in sensor measurement data for the same accuracy. This leads to a TL recommendation rate of 96.8%.

TL Analysis for All Types: Here, we group the floor images from all different environments together having `4 Txs`, and either 1 cube, 2 cubes, 3 cubes, 4 cubes, 8 cubes, and environments with a different layout. This results in a total of 1176 different floor images (see Table B.2). While training the CNN model $SimNet_A$ under Method A (see Figure B.7 and Table B.4), we obtain a training MSE and a test MSE of 0.0254, and 0.032, respectively. Similarly, while training the CNN model $SimNet_B$ under Method B (see Figure B.7 and Table B.4), we obtain a training MSE and a test MSE of 0.0055, and 0.0141, respectively. Once again, this reflects that the training of both CNN models is satisfactory.

Next, Table B.5 (b) presents the TL decision for both methods under all types. Table B.5 (b) reflects that for Method A, all 1176 different environments satisfy our criteria and are recommended for TL, with a training data requirement in the range of 10-30%. This leads to a TL recommendation rate of 100%. Similarly, for Method B, 876 out of 1176 different environments satisfy our criteria and are recommended for TL, with a training data requirement in the range of 5-40%. This leads to a TL recommendation rate of 74.48%. The reason for observing a smaller TL recommendation rate under Method B, is due to the fact that Method B is more strict towards satisfying both the test MSE and the number of training epochs threshold criteria, as compared to Method A, where only the test MSE criterion needs to be satisfied.

B.5.4.3 Model Accuracy

To understand how good and reliable our proposed scheme is, Table B.6 presents the transferability, F1-score, and accuracy for Type I environments, as well as all types of environments grouped together (see Table B.2). It can be noticed from Table B.6 (a) that the transferability, F1-score, and accuracy of the Type I environments under Method A are slightly better than in Method B. The model accuracy under Method A is 94.8%, indicating a high accuracy. On the other hand, the model accuracy under Method B is 93.6%.

Moreover, it can be noticed from Table B.6 (b) when all types of environments $\mathcal{E}_t$

Table B.6: Transferability, F1-score, and Accuracy for Type I environment and all types of environments grouped together.

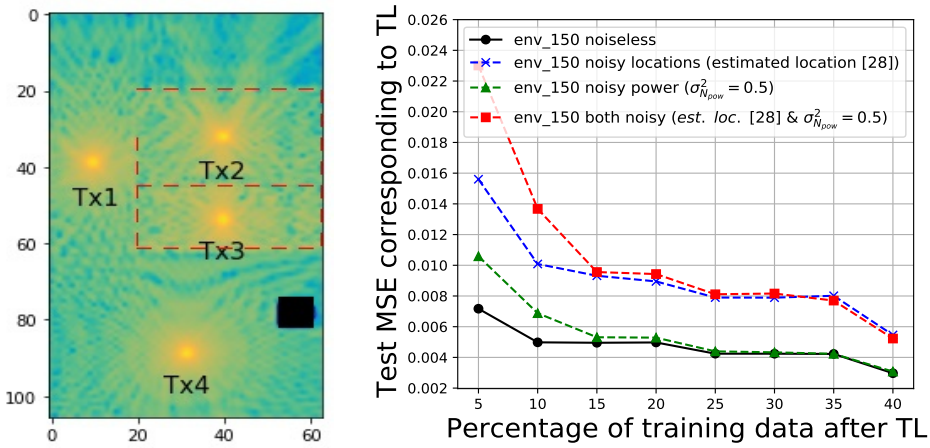
	(a)		(b)	
	Type I		All Types grouped together	
Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	237	226	958	740
TN	0	8	0	253
FP	13	10	218	105
FN	0	6	0	78
Transferability	1.0	0.974	1.0	0.904
F1-score	0.973	0.965	0.897	0.889
Accuracy	0.948	0.936	0.814	0.844

are grouped together, then the transferability and F1-score under Method A are better than in Method B. The model accuracy under Method A is 81.4%. On the other hand, the model accuracy under Method B is 84.4%. This can be attributed to the fact that Method B has a larger network than Method A which may make it more robust. This suggests that our proposed TL method is adaptable to different changes (e.g., changes in carrier frequency, the height of antennas, adding an extra wall, etc.) in wireless environments, and can be used effectively for estimating accurate radio maps in a new wireless environment while requiring a smaller amount of sensor measurements, as compared to the case of training a raw model without TL.

B.5.4.4 Noise Robustness

To see the robustness of our proposed scheme with respect to noise, as discussed in Section B.5.2.2, Figure B.11 shows the plot between the test MSE corresponding to TL and the necessary percentage of training data after the TL operation, for the noiseless and the noisy environments. The plot corresponds to one of the wireless environments (Figure B.11(a)) from the Type I environment $\mathcal{E}$ (see Table B.2).

It can be seen from Figure B.11(b) that as the percentage of training data increases, the test MSE corresponding to TL i.e., MSE($\mathcal{E}_t$) decreases. For the noiseless environment, the MSE($\mathcal{E}_t$) is smaller (shown by the solid black line). However, when we have noisy power values (see Section B.5.2.2), there is a significant increase in the MSE($\mathcal{E}_t$) (shown by the dashed green line) when using a smaller amount of training data (upto 25%) and thereafter, it overlaps with the noiseless test MSE. This signifies that our proposed scheme is robust against noisy power values. On the contrary, when estimated `Rx` locations (see Section B.5.2.2) are used [1], then the MSE($\mathcal{E}_t$) increases significantly (shown by the dashed blue line) and the increment is slightly higher, as compared to the case of having noise in the power values. This means that the errors in the estimated `Rx` locations affect more than the noise in the power values. This signifies that our proposed scheme is less robust against the errors in



(a) (b)

Figure B.11: (a) Radio map for a wireless environment from Type I $\mathcal{E}$, and (b) plot between test MSE corresponding to TL and percentage of training data after TL for the noiseless and noisy environment corresponding to the same environment (a).

location estimations as compared to the noisy power values. Similarly, when we have both estimated `Rx` locations and the noisy power values, then the overall increment in the MSE($\mathcal{E}_t$) is comparatively higher (shown by the dashed red line) than the noiseless environment. This means that our proposed scheme is less robust against having a combination of errors in location estimations and noisy power values. In addition, it can also be seen that for noisy scenarios, more data is needed to compensate for this noise, as expected. For instance, to achieve a MSE($\mathcal{E}_t$) 0.005, our noiseless model requires only 10% of training data. However, to achieve the same amount of MSE($\mathcal{E}_t$), our model trained with estimated `Rx` locations and no noise in power values requires 40% of training data.

B.5.4.5 Time Complexity

To understand the trade-off between the training time taken by the DNN while performing TL operation for each environment (total of 1176 environments), we calculate the amount of savings in training time (expressed in terms of training epochs). Since we have already set a threshold for the number $N_{EP_{thr}}$ of training epochs as 12, we compare this threshold with the number of training epochs obtained by each new $\mathcal{E}_t$ after performing TL operation, as shown in Figure B.12.

Figure B.12 illustrates that most of the environments are saving a large amount of training time by reducing the number of training epochs, and are recommended for TL (see environments above the dashed blue line). However, some of the environments are taking more time due to their increased number of training epochs as compared to the set threshold $N_{EP_{thr}}$, resulting in not being recommended for TL (see environments below the dashed blue line). Moreover, if we decrease the default threshold of the number $N_{EP_{thr}}$ of training epochs from 12 (see Table B.4), then the number of environments recommended for TL is also deceased and correspondingly the saving in training time is also increased because it needs more training epochs to

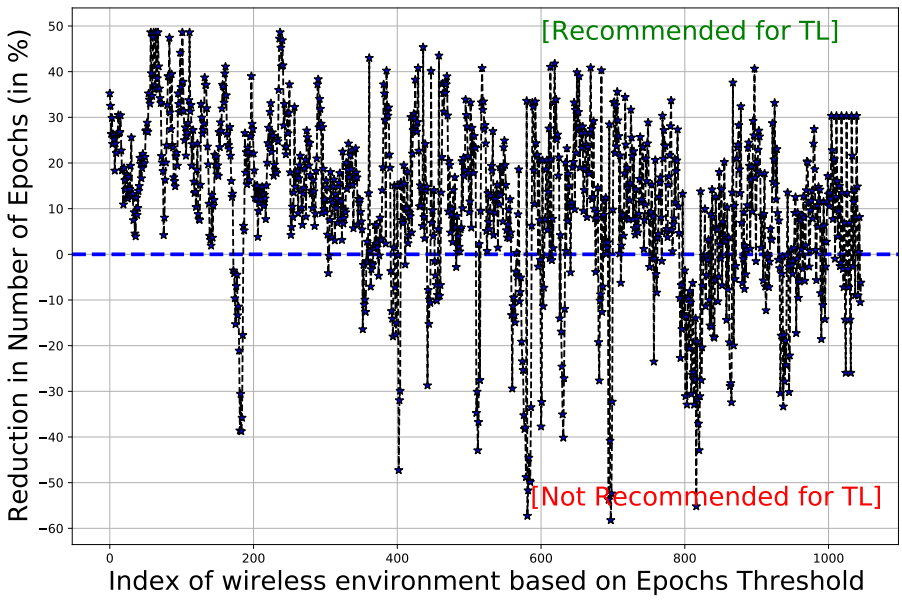


Figure B.12: Reduction in number of training epochs for each new environment.

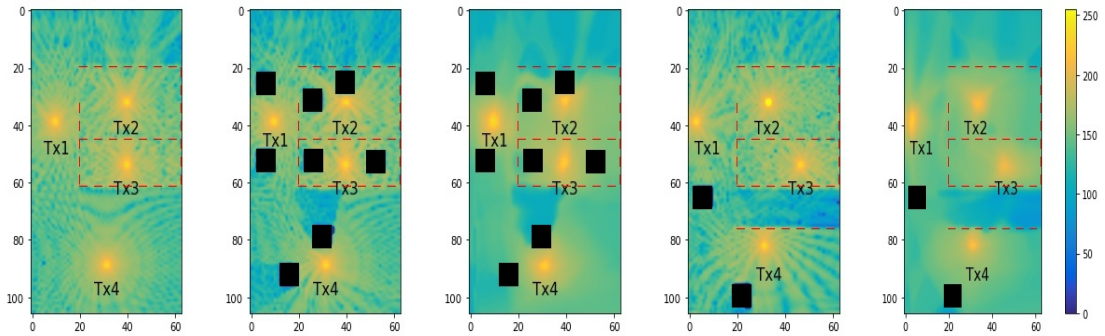
train the new environments, resulting in not satisfying the threshold $N_{EP_{thr}}$ criteria, and vice-versa. Furthermore, Method B has a larger amount of training parameters to train the CNN as compared to Method A. Method B has 41,824 parameters as compared to Method A, which has 23,384 parameters. In the execution phase, Method B has 41,824 weights as compared to Method A, which has 23,384. This takes 1.322 second for Method B as compared to 0.609 second for Method A for 209 wireless environments.

B.5.4.6 Illustration of Estimated Radio Maps

As an illustration, Figure B.13 represents the radio maps corresponding to the original wireless environment without any cube present, with `4Txs` located at four different fixed positions, an environment from Type V (8 cubes present), and the same environment after performing the TL operation where only 40% of the training data is required, an environment from Type VI (changed `Txs` locations and an additional wall), and the same environment after performing the TL operation where only 35% of the training data is required, respectively. The cubes and the walls of the rooms in the radio maps are represented in black and dashed red colours, respectively. The differences in these five radio maps can be easily visualized.

B.5.4.7 Temporal Window Experiment

In this experiment, we have used a *MapNet* model, trained and fine-tuned for one environment for the next 7 environments (without fine-tuning). The cube (object) is moved 15 cm towards the right every time with a speed of 4 km/h (typical walking speed). We observe that the test MSEs are almost the same when transiting across the first six environments (800 ms), but there is a large increase for environment 7 at 945 ms, that is, where the cube moves more than 1 m. Thus, we can say that the temporal window is 800 ms, and fine-tuning is unnecessary during this window.



(a) (b) (c) (d) (e)

Figure B.13: Radio map for (a) original (source) wireless environment with `4 Txs` located at four different fixed positions, (b) a wireless environment from Type V environment, (c) same environment as in (b) after performing TL where only 40% of the training data is required to train the DNN resulting in a test MSE, i.e., MSE($\mathcal{E}_t$) as 0.0048, (d) a wireless environment from Type VI environment (changed `Tx` locations and additional wall), and (e) same environment as in (d) after performing TL where only 35% of the training data is required to train the DNN resulting in a test MSE, i.e., MSE($\mathcal{E}_t$) as 0.005.

B.5.4.8 Comparison with State-of-the-art

To showcase the efficacy of our proposed TL method in estimating accurate radio maps, we compare the obtained results with the widely used RadioUNet [3] method.

RadioUNet is a modified UNet [14] architecture (originally designed for biomedical image processing application), which is used to estimate radio maps in both indoor and outdoor urban environments. It uses convolutional auto-encoders [14] for mapping the images of the wireless environments and `Tx` locations (input) to the corresponding radio maps (output).

RadioUNet mainly comprises two UNets [14]. The input of the first UNet has two feature channels (images of indoor wireless environments and images of `Tx` locations), and the output has one feature channel i.e., estimated radio maps. The input to the second UNet might have additional channels as compared to the first UNet. However, in our experiment, both UNets use the same two feature channels without loss of generality. During training, the first UNet is trained to estimate the radio maps. After that, the weights of the first UNet are frozen, and the second UNet is trained to estimate the radio maps. The different parameters used in the RadioUNet are presented in Table B.7.

In this experiment, we use the RadioUNet method with the parameters as presented in Table B.7 for the dataset of 1176 indoor wireless environments presented in Section B.5.1. Note that the RadioUNet method is trained with several environments at the same time. In the test phase, it estimates the radio map without TL and fine-tuning. The RadioUNet method results in training, validation, and test MSE of 0.0032, 0.0036, and 0.0033, respectively. In comparison, our baseline DNN model *MapNet* has a training and test MSE of 0.0024, and 0.0026, respectively,

Table B.7: Parameters used in the RadioUNet [3]

Layer	Resolution		Channels		Filter	
	First UNet	Second UNet	First UNet	Second UNet	First UNet	Second UNet
Input	256		2		3	
First	256		6	20	5	
Second	128		40	30	5	
Third	64		50	40	5	
Fourth	64		60	50	5	
Fifth	32		100	60	3	
Sixth	32		100	70	5	
Seventh	16		150	90	5	
Eighth	8		300	110	5	
Ninth	4		500	150	4	
Tenth	8		300+300	110+110	4	
Eleventh	16		150+150	90+90	4	
Twelfth	32		100+100	70+70	3	
Thirteenth	32		100+100	60+60	6	
Fourteenth	64		60+60	50+50	5	
Fifteenth	64		50+50	40+40	6	
Sixteenth	128		40+40	30+30	6	
Seventeenth	256		20+6+2	20+2+2	5	
Eighteenth	256		20+2	20+2	5	
Output	256		1	-	-	
Loss function	MSE					
Optimizer	Adam					
Learning rate	0.0001					
Batch size	15					
Epochs	15					
Train-Val-Test data split	76:12:12					

which is smaller than the RadioUNet.

Next, Figure B.14 shows radio maps corresponding to an indoor wireless environment from Type III (3 cubes present), the same environment after performing the TL operation using the proposed TL method where only 35% of the training data is required, and the same environment estimated using RadioUNet. We can observe that the radio map estimated using RadioUNet has a very high value of test MSE (0.4274). However, the radio map estimated using our proposed TL method is better and more accurate with less value of test MSE (0.0038). Similar results are observed for other types of wireless environments (see Figure B.9). This signifies that our proposed TL method outperforms the RadioUNet method and is more efficient in estimating accurate radio maps for new indoor wireless environments. This

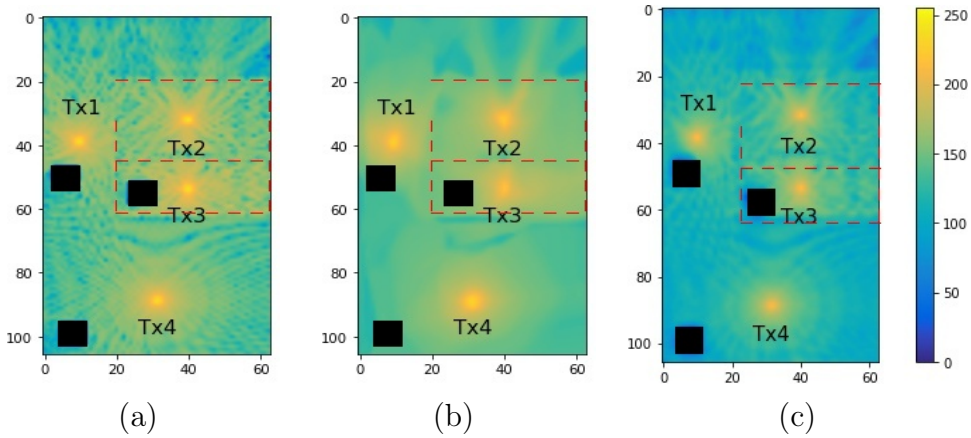


Figure B.14: Radio map for (a) a wireless environment from Type III environment, (b) same environment as in (a) after performing the TL operation using the proposed method with only 35% training data (MSE($\mathcal{E}_t$) = 0.0038), and (c) same environment as in (a) estimated using RadioUNet (MSE($\mathcal{E}_t$) = 0.4274).

is caused by the fine-tuning which is performed for each new target environment after the TL operation in our method, as compared to the RadioUNet where no fine-tuning is performed. In general, the RadioUNet has two CNNs with 18 hidden layers each. On the other hand, our proposed TL method has one DNN with 6 hidden layers and one CNN with 4 hidden layers. Thus, our proposed TL method is smaller and typically requires less training data.

B.5.4.9 TL Performance with Cafe Area Indoor Wireless Environment

In this experiment, we have made more realistic indoor wireless environments corresponding to a typical cafe area, which includes the following objects: a kitchen, a fridge, two tables, twelve chairs, four men sitting, and two standing men moving inside the cafe. The cafe floor dimension (length x width) is 9.5mx16m. It has 5 windows with a size of 4mx1m each, and two doors with a size of 2mx3m each. All environments have 4 `Txs` placed at 4 fixed locations and multiple `Rxs` placed at a uniform spacing of 15 cm. `Tx1`, `Tx2`, `Tx3`, and `Tx4` are placed at (1.5, 10, 3)m, (7.5, 11, 3)m, (4.75, 6.5, 3)m, and (4.75, 1.25, 3)m, respectively.

In the source cafe area wireless environment, all these objects are fixed without the two standing men.

The target cafe area wireless environments are created by moving one or two men. Each time they move a distance of 15 cm, it generates a new target cafe area wireless environment. A total of 202 different cafe area wireless environments are created. The aggregate received power for each target cafe area wireless environment is obtained for 6678 `Rx` locations. The illustration of TL system design between the original and the target cafe area wireless environments is shown in Figure B.15.

To perform the TL operation in the new cafe area wireless environments, we investigate the suitability of the previously developed baseline DNN model (see Section B.5.4.1). However, we experimentally found that the DDS predicts a test MSE of 0.0092 for the baseline cafe area wireless environment, which is higher than

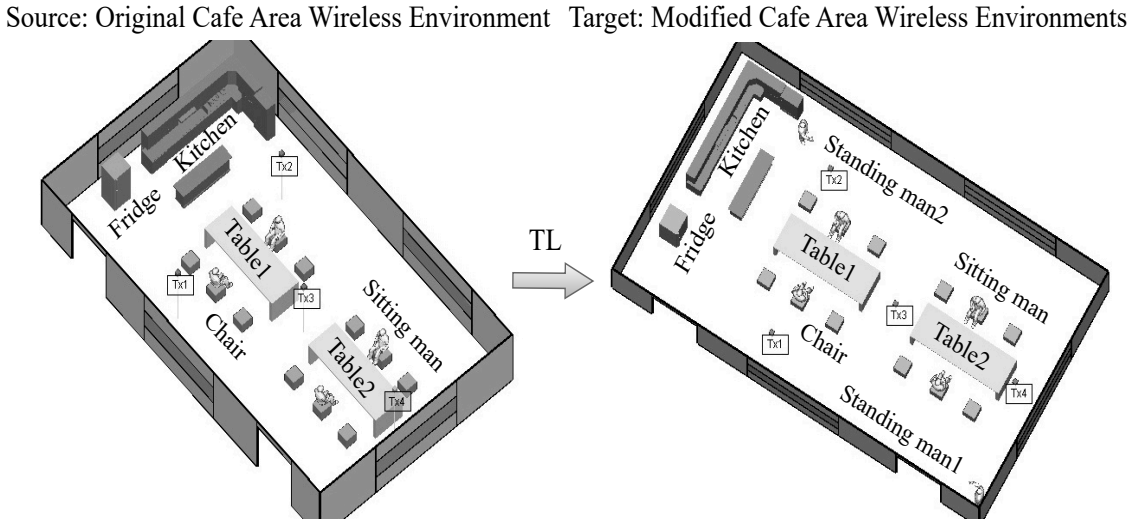


Figure B.15: Illustration of the source and the target cafe area wireless environments.

Table B.8: Parameters for cafe $SimNet_A$ and $SimNet_B$

	$SimNet_A$	$SimNet_B$
Cafe floor image size (width × height)	298 x 500	
Number of convolutional layers	3	
Filters in each convolutional layer	256, 128, 32	256, 128, 64
Filter size in each convolutional layer	3 x 3	
Number of max pooling layers	3	
Filter/pool size in each max pooling layer	2 x 2	
Strides in each convolutional layer	2	
Activation function in convolutional layer	ReLU	
Neurons in Ist flatten layer of neural network	16	32
Dropout after Ist flatten layer	0.2	0.2
Neurons in output layer of neural network	8	16
Activation function in flatten and output layer	ReLU	
Optimizer with learning rate	Adam (0.001)	
Loss function, Batch size	MSE, 16	
Training-Testing split ratio	70:30	
Number of iterations for early stopping	5	
Threshold MSE_{thr} for MSE	0.00385	
Threshold for number of training epochs	-	8

the previously set test MSE threshold (0.005). Hence, we learn a new baseline DNN model for all cafe area wireless environments. The new baseline is trained with the power values obtained at multiple locations of `Rxs` from the source cafe area wireless environment. Using grid search as before (see Section B.5.4.1), the best parameters obtained here are the same as in Table B.3 except the addition of dropouts [79] in the initial four hidden layers as 0.25, batch size as 16 and training-testing data split ratio as 60:40. The obtained training and test MSE are 0.0036 and 0.0038,

Table B.9: Transferability, F1-score, and Accuracy

Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	194	190
TN	0	0
FP	8	12
FN	0	0
Transferability	1	1
F1-score	0.979	0.969
Accuracy	0.96	0.94

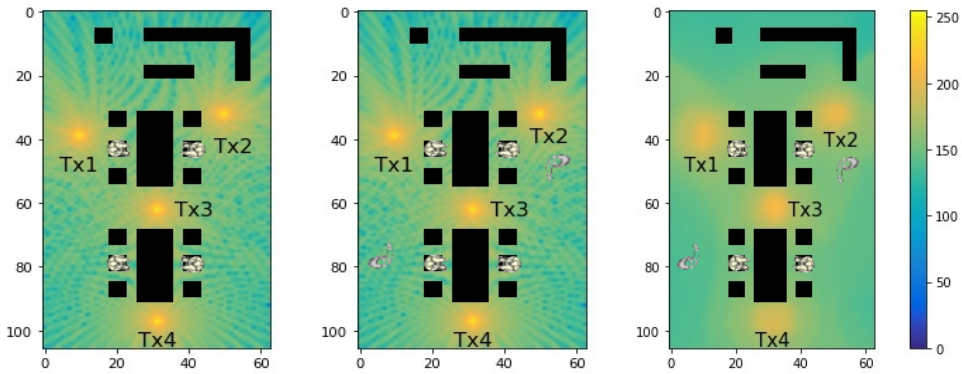
respectively. Further, this baseline model is transferred to each new target cafe area wireless environment and fine-tuned, and the corresponding test MSE MSE($\mathcal{E}_t$) and the number $N_{EP}(\mathcal{E}_t)$ of training epochs are saved, which are then used for the DDS.

TL Performance Analysis: To train the DDS, we consider 202 different cafe area wireless environments which correspond to 202 different cafe area floor images. While training the CNN regression model $SimNet_A$ under Method A (see Table B.8), we obtain a training and test MSE of 0.0198 and 0.0242, respectively. Similarly, while training the CNN regression model $SimNet_B$ under Method B, we obtain training and test MSE of 0.037 and 0.0451, respectively. This reflects that both CNN models are trained appropriately.

Next, we present the DDS results for both methods. Under both Method A and Method B, all 202 (100%) different cafe area environments satisfy our criteria and are recommended for TL, with a training data requirement of only 35%.

Cafe Area SimNet Model Accuracy: Table B.9 reflects that the transferability under both methods is the same. However, the F1-score and accuracy under Method A are slightly better than Method B. The accuracies are 96% and 94% under Method A and Method B, respectively. Intuitively, Method A has to satisfy only the MSE threshold while Method B has to satisfy both the MSE and number of epochs threshold criteria.

Illustration of Estimated Cafe Area Radio Maps: Figure B.16 shows the radio map of the original (source) cafe area indoor wireless environment, one of the target cafe area indoor wireless environments with two men moving, and the same target cafe area environment after performing TL with only 35% training data. The differences in the radio maps can be easily visualized. Note that all objects in the radio maps are shown in solid black colour. This signifies that our proposed TL method is effective for real-world environments, such as cafe area wireless environments. Moreover, the cafe area wireless environments can be easily modified to represent other real-world environments, such as conference area type environments (E.g., the cafe area environment with the addition of a whiteboard acts like a conference area environment).



(a) (b) (c)

Figure B.16: Radio map for (a) original cafe area environment, (b) a target cafe area environment, and (c) same environment as in (b) after performing TL using the proposed method with only 35% training data (MSE($\mathcal{E}_t$) = 0.0036).

B.6 Conclusion and Future work

In this paper, we present a novel TL method to estimate radio maps in indoor wireless environments. A data-driven similarity measure is also designed to quantify the similarity between two wireless environments which is then used to decide whether to recommend performing the TL operation, given only the floor image of the wireless environment. Our data-driven similarity measure outperforms other similarity measures for the TL operation when estimating radio maps. It is shown to be highly effective when the amount of training samples in the new (target) wireless environment is limited. Our proposed TL method reduces the requirement in the number of training samples necessary for estimating accurate radio maps in environments with a certain level of similarity, saving approximately 40-90% of sensor measurement data. In this work, the experimental results are based on simulated data from the Remcom simulator [33]. In the future, we plan to investigate the applicability of the proposed method to outdoor and real (non-simulated) wireless network scenarios using data from the network operators.

Appendix C

PAPER C

Title: Location-free Indoor Radio Map Estimation using Transfer learning

Authors: **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano

Conference: IEEE VTC-Spring 2023

Location-free Indoor Radio Map Estimation using Transfer learning

R. Jaiswal, M. Elnourani, S. Deshmukh, B. Baltasar Lozano

Abstract: **Accurate estimation of radio maps is important for various applications of wireless communications, such as network planning, and resource allocation. To learn accurate radio map models, one needs to have accurate knowledge of transmitter and receiver locations. However, it is difficult to obtain accurate locations in practice, especially, in scenarios having a high degree of wireless multi-path. Alternatively, time of arrival (ToA) features, which are easier to obtain, can be employed for estimating radio maps. To this end, this paper investigates the application of the transfer learning method using ToA features for estimating radio maps under indoor wireless communications. The performance is compared with the scenarios where only the locations of receivers and both ToAs and locations of receivers, are used for estimating radio maps, assuming that locations are known. Due to the changes in propagation characteristics, a radio map model learned in a specific wireless environment cannot be directly employed in a new wireless environment. To address this issue, a data-driven transfer learning method is designed that transfers and fine-tunes a deep neural network model learned for a radio map from a source wireless environment to other distinct (target) wireless environments. Our proposed method predicts the training data required in the new wireless environments using a data-driven similarity measure. Our results demonstrate that using ToA (location-free) features results in superior performance for estimating radio maps in terms of the necessary number of sensor measurements for estimating radio maps with good accuracy, as compared to a location-based approach, where it may be difficult to have accurate location estimations. It leads to a saving of 70-90% of the necessary sensor measurement data for a mean square error (MSE) of 0.004.**

C.1 Introduction

Estimating accurate radio maps is crucial for numerous applications in wireless communications, such as network planning, spectrum management, channel modelling, and resource allocation, to name a few. Radio maps provide typically the value of received power at every spatial location over a certain frequency band of interest in

any geographical area. The received powers at distinct spatial locations vary due to distinct factors, for instance, reflections, diffractions, propagation loss, and objects that block the line of sight between a transmitter (Tx) and a receiver (Rx).

Most of the algorithms to estimate radio maps are based on the knowledge of the Tx and Rx locations. This is known as the location-based radio map estimation [34]. However, accurate knowledge of sensor locations is difficult to obtain specially in environments where high wireless multipath fading is present. Alternatively, one can employ the features based on the received signals, such as the time of arrival[1] (ToA) for estimating the radio maps. This is known as the location-free radio map estimation [6]. Both the location-based and the location-free features are combined to estimate the channel gain maps in [80].

Because of the differences in propagation characteristics, a radio map model developed in a specific wireless environment, may not work appropriately in the new wireless environments. Training of machine learning (ML) methods, as a consequence, require substantial data for new wireless environments. Thus, one needs to have a large number of samples and training epochs for training the deep neural network (DNN)-based methods. Moreover, adequate computational time and data acquisition cost are needed while training the DNN for each new wireless environment. For alleviating these costs, one can develop a radio map model in one wireless environment (source environment) and then smartly use this model in a similar wireless environment (target environment), by exploiting the concept of transfer learning (TL) [19].

Model-based transfer learning [19] is the process of exploiting acquired knowledge from a certain learning task to another target task. TL handles the issue of data scarcity in the target environment in terms of reducing the amount of necessary data for efficient learning. In the scenario of estimating radio maps, one may have learned a radio map model in one indoor wireless environment but needs to estimate radio maps in other similar indoor wireless environments. Using TL, one does not require to learn a solution from the very beginning in the target environment, which requires in general a large amount of data, instead, the knowledge from the source model can be exploited.

Few works consider exploiting TL for wireless communications. A TL-based approach has been employed from the perfect channel state information (CSI) scenario to the imperfect CSI scenario in underlay D2D communications for jointly allocating channel and power [36]. An estimation of radio maps for indoor wireless environments employing the locations of Txs and Rxs, using TL, is investigated in [34]. In [20], a model-based TL is employed for capturing the diversity of cellular traffic patterns of different cities. For improving the robustness of the spectrum sensing algorithm developed for cognitive radio, TL is also applied in [21]. For tackling the non-convex resource allocation problem in [32], TL is employed to transfer from one solver to another solver. In [23], the optimal transport-based TL approach is designed which minimises the Wasserstein distance [24] for wireless fingerprinting

[1]Time of arrival (ToA) is the time at which the radio signals arrive at the Rx from the Tx.

localization.

This work employs TL for estimating radio maps in indoor wireless communications environments when the wireless environment changes. The following are the key contributions of this paper:

- Design of an efficient DNN-based model that learns a radio map for an indoor wireless environment. Additionally, the design of a data-driven TL method that transfers a DNN model for a radio map learned from a *source* baseline indoor wireless environment to another distinct *target* indoor wireless environment and further fine-tunes that DNN model.

- Design of a data-driven similarity measure model that maps the images of indoor wireless environments to the mean square error (MSE) that will be achieved for the estimated radio map in a new indoor wireless environment when performing the TL operation from a baseline (source) environment to a new (target) environment.

- Estimation of the amount of training data required for training in the new wireless environment, depending on a certain criterion of the MSE and the training epochs thresholds for estimating the radio maps, while executing the TL operation, employing the data-driven similarity measure that we design.

- Extensive testing of our algorithms using simulated data from the Remcom simulator [33]. Our simulation results demonstrate that employing only ToA (location-free) feature is better in estimating accurate radio maps when the location information is not accurately known.

- Numerically, we show that the proposed TL method employing only ToA (location-free) feature requires less amount of training data as compared to the location-based method [34]. It is also shown experimentally that a similarity measure based on the Wasserstein distance (WD), which is widely used in TL, is not applicable to our radio map application.

The remaining paper is organized as follows: Section C.2 explains the generation of distinct wireless environments for TL. The proposed radio map estimation method is introduced in Section C.3. Section C.4 discusses the numerical results. Section C.5 presents the conclusion.

C.2 Generation of Wireless Environments

The highly accurate and standard ray-tracing 3D ray model [41], calculated using Remcom [33], is employed to obtain power measurements for real indoor wireless environments. Along this line, Figure C.1 depicts an indoor environment which consists of a single floor with two rooms. For obtaining power measurements $\{P(x_i^r)\}$ at every Rx location $\{x_i^r\}$, a transmitter "Tx" is placed at one fixed location $\{x^t\}$, where i is the receiver location index. Also, several receivers are uniformly located.

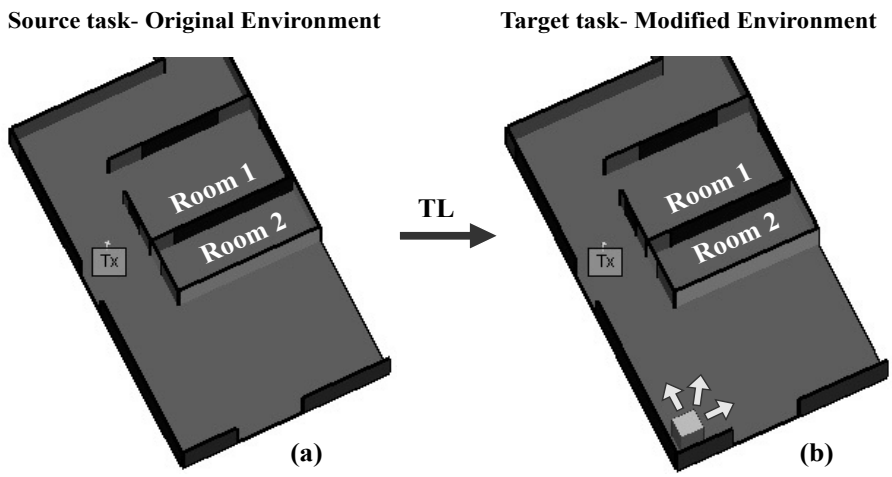


Figure C.1: TL environments: (a) original (source) environment, and (b) modified (target) environments.

Table C.1: Remcom parameters to generate data.

Waveform	Narrow-band Sinusoidal
Frequency of carrier with bandwidth	900 MHz (1 MHz)
Type of antenna	Omni-directional
Location of Tx	(1.5, 10, 1.3) m
Height of Tx	1.3 m
Power of Tx	27.73 dBm
Threshold of Receiver	-250 dBm
Voltage standing wave ratio (VSWR)	1.0
Loss of Transmission line	0 dB
Two Rx points separation	15 cm
Two Rx routes separation	15 cm
Noise figure	3 dB
Ray spacing	0.2°
No. of reflections, transmissions, and diffractions	3, 2, 0
Object volume (single cube)	1 m^3
Number of Rxs for locations and ToAs	6678
Dimension of floor (width x length x height)	(9.5 x 20.6 x 2.88) m
Total indoor environments generated	250
Floor image size (width x height)	(160 x 275) pixels

Similarly, for each Rx location $\{x_i^r\}$, we acquire the ToA estimated also using the model in [33], denoted as $\{\tau_i^r\}$, and we associate it with the corresponding received power at that location, which leads to a mapping of the form $\{P(\tau_i^r)\}$. The dimension of the floor, the sensor placement, and other parameters employed in Remcom are presented in Table C.1. A total of 6678 ToA features corresponding to every Rx location and power measurement are obtained.

Next, for creating distinct indoor wireless environments conceptually, an object, that is, a single solid cube block having a volume of $1m^3$ and made of metal is incorporated inside the original indoor environment. With an incremental spacing of 15cm, the location of the object changes in each direction, as depicted by arrows in Figure C.1(b). Each time new indoor environment generates whenever this object arrives at a new location. At the same time, the power measurements are obtained for this new environment by the ray-tracing model. This process results in 250 distinct indoor wireless environments. In addition, we save the images of each environment having an image size of 160 x 275 pixels. These images are later exploited for investigating different similarity measures for our TL problem, such as the WD and the data-driven similarity measure among distinct indoor wireless environments.

C.3 Radio Map Estimation Method

To estimate a radio map from a source to a target environment, the proposed TL scheme is depicted in Figure C.1. Defining similarity among source and target environments is important to perform TL operation. Along this line, a baseline DNN model (see Figure C.2) is developed in the source environment and then the target environment exploits this baseline model and fine-tunes it with additional data available from the target environment for establishing the similarity between both environments. In other words, the test MSE and the number of training epochs that are achieved after transferring the baseline DNN and re-training it in the new environments, are used as inputs to our data-driven similarity measure which is designed using the convolutional neural network (CNN) (see Figure C.3).

C.3.1 Baseline Model and TL Operation

Firstly, a DNN model is developed which approximates the power values for the original indoor environment (see Figure C.1(a)). Here we propose two methods. Method 1 uses both ToAs and locations of Rxs, and Method 2 uses only ToA (location-free) features. Along these lines, fully-connected DNN models are trained for each method and are referred to as respective "Baseline models", as shown in Figure C.2.

Next, each baseline model is transferred to each new indoor environment (a total of 250 environments) and fine-tuned separately. Our hypothesis is: for training a DNN under the new environment, no ample amount of training data is available. Thus, to estimate radio maps in the new environment, one can exploit the TL operation for training a DNN model in the new environment with only a small amount of training data.

The data generated from Remcom (see Section C.2) is normalized to be between 0 and 1 to favour numerical stability in the training of DNN [34]. Let $\mathbf{P}(\tau_{\mathbf{i}}, \mathbf{x_i})$ and $\hat{\mathbf{P}}(\tau_{\mathbf{i}}, \mathbf{x_i})$ be the actual and the estimated power values, respectively, the mean

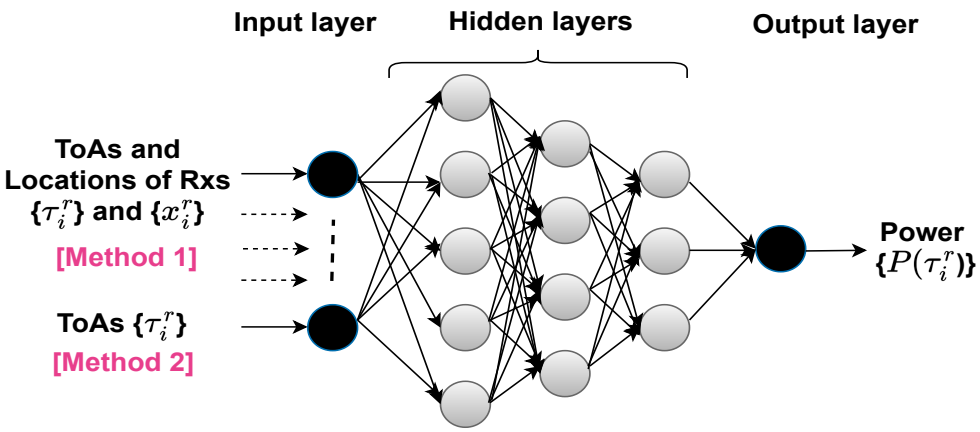


Figure C.2: DNN model for radio map estimation.

square error (MSE) [64] which is used as the loss function is calculated as, MSE = $\frac{\sum_{i=1}^{r}(\mathbf{P}(\tau_\mathbf{i},\mathbf{x_i})-\hat{\mathbf{P}}(\tau_\mathbf{i},\mathbf{x_i}))^\mathbf{2}}{r}$, where r is the total number of Rx locations.

Generally, the amount of training data needed for performing the TL operation relies on its similarity measure. For example, less amount of training data is needed between the two environments, if both environments are similar, and vice-versa. Thus, a similarity measure is needed to determine whether the TL operation is effective.

C.3.2 Similarity Measure based on the Wasserstein Distance

The Wasserstein distance (WD) [23] is one of the widely used similarity measures employed in the TL for establishing the similarity between a source and a target task. The distance between the two distributions is computed using WD. In the proposed TL scheme, the WD among the two images corresponding to the two different indoor wireless environments is computed. In practice, one computes cumulative distribution functions (CDFs) (see Equation C.1) empirically from the corresponding histograms with a sufficient amount of data. Let us consider two random variables S and T. In our case, S and T represent the image values corresponding to an environment in which the cube is located at the left bottom corner (see Figure C.1), and image values corresponding to another environment in which the cube is located at a distinct location, respectively. Let $F_S(s)$ and $F_T(t)$ be the respective CDFs. $\mathcal{F}_{\mathcal{ST}}$ and $\mathbb{E}_{F_{ST}}[\cdot]$ be the joint CDFs, and the expectation with respect to the joint CDFs, respectively. The WD between both environments is calculated as [51]:

$$d(S,T) = \inf_{F_{ST}\in\mathcal{F}_{\mathcal{ST}}} \mathbb{E}_{F_{ST}}[|S-T|] \tag{C.1}$$

C.3.3 Data-driven Similarity Measure

As shown later in Section C.4.2, it can be shown experimentally that typical similarity measures, such as the widely used WD, are not effective in our application of radio map estimation (as compared to other applications), since it is not able to track the variations of the radio maps as the indoor wireless environment varies.

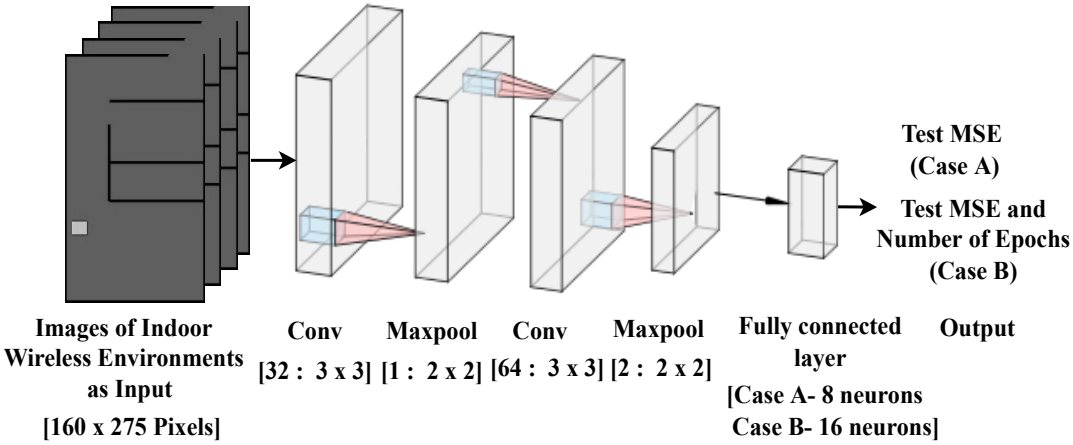


Figure C.3: CNN model for similarity measure.

This motivates the design of a data-driven similarity measure (DDS) for our TL problem that is able to understand the structure of the data in radio maps, and how it varies as the indoor wireless environment changes.

To determine the substantial reduction of training data during the TL operation using our data-driven similarity measure, the training data in the target indoor wireless environments varies from 5% to 40%. For establishing the similarity among the two environments, the test MSE corresponding to TL and the number of epochs employed in DNN training are stored, once the DNN is trained in each new (target) environment. Next, to design a DDS (see Figure C.3), a CNN regression model is trained within two distinct cases for each of the methods using both ToAs and locations of Rxs features, and using only ToA (location-free) features, respectively and then a threshold is set empirically for its comparison with the original one. The CNN networks used in both cases are shown in Figure C.3.

C.3.3.1 Case A

The images of distinct indoor wireless environments (a total of 250 environments) are the input of the CNN model. The test MSE obtained after performing the TL operation over each environment is the output of the CNN. Each coloured floor image has a size of 160 x 275 pixels. Before injecting these images into the CNN, they are converted into grayscale images using the OpenCV [81] library.

Next, for recommending the TL operation among two different environments, an empirical threshold for the test MSE corresponding to TL $\beta_{TestMSE}$ is set as 0.01. Two environments are recognized as sufficiently similar, if the test MSE obtained after performing the TL operation is less than this threshold, else, the TL operation is not recommended.

C.3.3.2 Case B

The same input (as in case A) is provided to the CNN model. The test MSE and the number of training epochs obtained after performing the TL operation over each environment are the output of the CNN model. The training epochs provide

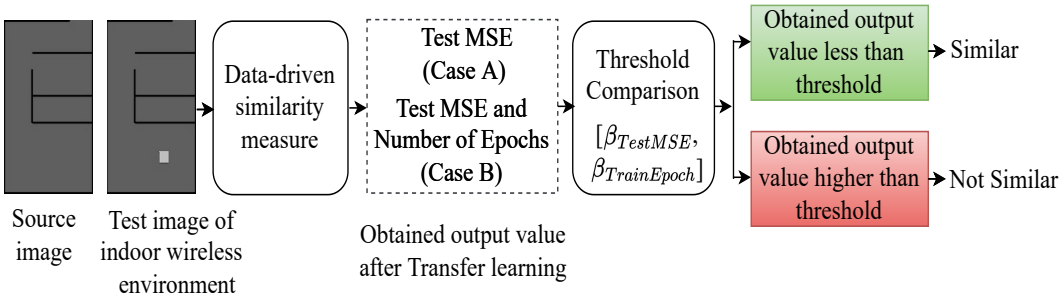


Figure C.4: Similarity decision between two wireless environments.

information about the time needed to train the model for a particular environment. Note that, a larger CNN model is required for this case, resulting in a need for larger feature sets. Hence, it becomes computationally demanding.

Next, for recommending the TL operation among two different environments, an empirical threshold for the test MSE corresponding to TL $\beta_{TestMSE}$, and the number of training epochs $\beta_{TrainEpoch}$ is set as 0.01 and 20, respectively. Two environments are recognized as sufficiently similar, if both values are less than these thresholds, else, the TL operation is not recommended. Figure C.4 illustrates the similarity decision taken between two indoor wireless environments under each case of the proposed TL method.

C.4 Results and Discussions

This section discusses the numerical results obtained in each method. Here we compare the performance of the proposed Method 1 and Method 2 with the TL method in [34] (Method 3) which employs only locations of Rxs for estimating radio maps in new indoor wireless environments.

C.4.1 Baseline DNN Model

Table C.2 presents the structures of the baseline DNN models that are used for each method (see Figure C.2). It also presents the structure of the baseline DNN model used in [34] which employs only locations of Rxs for estimating radio maps. We refer to the method used in [34] as *Method 3* for its comparison with Method 1 and Method 2. It can be seen from Table C.2 that under each method, the test MSE is comparable and is slightly larger than the train MSE which reflects no over-fitting in each baseline DNN model, resulting in successful learning of each baseline model.

Next, observe the test MSE obtained in Method 1 and Method 2 with the test MSE obtained in Method 3 in Table C.2, we notice that Method 1 and Method 2 have lesser test MSE values as compared to Method 3. Specifically, there is a reduction of around 69% and 55% in the test MSE values using both ToAs and Rx locations features (Method 1) and only ToA features (Method 2), respectively, as compared to using only locations of Rxs features (Method 3). This shows that Method 1 and Method 2 are having better baseline models as compared to Method 3 for the TL operation. Moreover, the test MSE of the baseline model under Method 2 (location-free) is almost the same as Method 1. Notice that the structure of each

Table C.2: The Baseline model learning

Parameters used	Both ToAs and locations of Rxs (Method 1)	Only ToAs (Method 2)	Only locations of Rxs [34] (Method 3)
Input layer neurons	4	2	2
Total hidden layers	6	3	2
Hidden layer neurons	512, 256, 128, 64, 32, 16	64, 32, 16	16, 8
Activation function in hidden layers	ReLU	ReLU	ReLU
Dropout after each hidden layer	-	-	0.20
Output layer neurons	1	1	1
Activation function in output layer	ReLU	ReLU	ReLU
Optimizer employed	Adam	Adam	Adam
Learning rate	0.0001	0.0001	0.001
Loss function	MSE	MSE	MSE
Size of mini-batch	16	16	32
Data split ratio for training-testing	70:30	70:30	80:20
Train MSE	0.00282	0.00459	0.0101
Test MSE	0.00313	0.00463	0.0102

baseline model in Table C.2 is optimal and is obtained using the grid search with its evaluation through 5-fold cross-validation.

C.4.2 Wasserstein Distance as a Similarity Measure

To test whether WD is suitable for our TL application, the WD between the floor images of indoor environments and the WD between their associated sampled radio maps (power values) when the cube is located at the left bottom corner (see Figure C.1(b)), and the rest of the indoor environments (a total of 250 environments) when the cube is located at other distinct locations, are calculated. Next, we compute Pearson's correlation coefficient (PCC) [34], which is one of the popular metrics for calculating correlations [20] between both kinds of WDs (floor images and associated sampled radio maps) across the several considered environments, then the correlation obtained is 0.08 that reflects a very poor correlation. This indicates that WD cannot be employed as a similarity measure for the TL-based radio map estimation in indoor wireless communications. The reason is that the WD is not able to capture effectively the changes in the radio maps caused by the variations in the locations of the cube. Hence, our designed data-driven similarity measure is exploited further for the TL-based radio map estimation problem.

Table C.3: The CNN model under Case A of Method 1.

Number of available indoor environments	250
Floor image size (width x height)	160 x 275
Total convolutional layers	2
Filters in 1st and 2nd convolutional layer	32, 64
Size of filters	3 x 3
Max pooling layers	2
Filter/pool size	2 x 2
Strides under 1st and 2nd convolutional layer	2
Convolutional layer activation function	ReLU
Dropout after 2nd max pooling layer	0.3
Output layer neurons of neural network	8
Output layer activation function	ReLU
Optimizer with learning rate	Adam (0.001)
Loss function	MSE
Size of mini-batch	16
Data split ratio for training-testing	80:20
Train MSE	0.01768
Test MSE	0.02790

Table C.4: The CNN model under Case B of Method 1.

Total output layer neurons of neural network	16
Remaining configurations are the same as in Table C.3.	
Train MSE	0.01256
Test MSE	0.02047

C.4.3 Data-driven Similarity Measure

C.4.3.1 Method 1

The model learnings of the CNN under case A and case B for the Method 1, respectively, are presented in Tables C.3 and C.4. Note that, the configurations to train the CNN in case B are the same as in case A, except for the output layer neurons which have now become 16 due to the combination of both the test MSE and the number of training epochs corresponding to TL. Both Tables reflect a proper training of CNN under each case, that is, the testing MSE is comparable and slightly larger than the training MSE.

C.4.3.2 Method 2

The model learnings of the CNN under case A and case B for the Method 2, respectively, are presented in Tables C.5 and C.6. Both Tables reflect the proper training of CNN under each case.

Next, as explained in Section C.3.3, the decisions of recommendation of TL oper-

Table C.5: The CNN model under Case A of Method 2.

All parameters and hyper-parameters are same as in Table C.3.	
Train MSE	0.02134
Test MSE	0.03223

Table C.6: The CNN model under Case B of Method 2.

Total output layer neurons of neural network	16
Remaining configurations are the same as in Table C.3.	
Train MSE	0.01420
Test MSE	0.02276

Table C.7: Performance of proposed TL method.

	ToAs and locations of Rxs (Method 1)		Only ToAs (Method 2)		Only locations of Rxs [34] (Method 3)	
	Case A (Only MSE)	Case B (MSE and Epochs)	Case A (Only MSE)	Case B (MSE and Epochs)	Case A (Only MSE)	Case B (MSE and Epochs)
Training data needed after TL	No. of Envs.	No. of Envs.	No. of Envs.	No. of Envs.	No. of Envs.	No. of Envs.
5%	250	184	208	78	-	-
10%	-	5	35	49	-	-
15%	-	11	7	11	-	-
20%	-	5	-	19	-	3
25%	-	8	-	6	-	31
30%	-	3	-	10	17	152
35%	-	1	-	3	206	31
40%	-	5	-	1	23	-
% of training data after TL	5%	5-40%	5-15%	5-40%	30-40%	20-35%

ations for both case A and case B of each Method are determined and are presented in Table C.7. It shows the number of indoor wireless environments that follow our proposed TL strategy and the corresponding amount of training data required after performing the TL operations. Table C.7 illustrates that under case A of Method 1, all 250 distinct environments achieve the threshold criteria and are therefore recommended for TL, with a need of only 5% training data. This accounts for 100% TL recommendation rate. In the same manner, under case B using Method 1, 222 out of 250 distinct environments achieve the threshold criteria and are therefore recommended for TL, with a need of only 5-40% training data. This accounts for 88.8% TL recommendation rate. On the other hand, under case A using Method 2, all 250 distinct environments achieve the threshold criteria and are therefore recommended

for TL, with a need of only 5-15% training data. This accounts for 100% TL recommendation rate. In the same manner, under case B of Method 2, 177 out of 250 distinct environments achieve the threshold criteria and are therefore recommended for TL, with a need of only 5-40% training data. This accounts for 70.8% TL recommendation rate. The reason for having a lower TL recommendation rate under case B for both Method 1 and Method 2 is due to not having a better correlation between the training epochs and the test MSE obtained after TL operation for the similarity measure.

Furthermore, in Table C.7, we also compare the requirement of the percentage of training data after TL under each case of Method 1 and Method 2 with Method 3, we notice that the wireless environments under case A of Method 1 require only 5% training data as compared to 30-40% in Method 3, saving around 85% training data. Similarly, most of the wireless environments under case B of Method 1 require only 5% training data and the remaining few wireless environments require only 10-40% training data. This illustrates the saving of a large amount of training data as compared to Method 3, where the wireless environments mostly require 20-35% training data. In the same manner, the wireless environments under case A of Method 2 (location-free) require only 5-15% training data as compared to 30-40% in Method 3, saving around 65% training data. Similarly, most of the wireless environments under case B of Method 2 (location-free) require only 5-10% training data and the remaining wireless environments require only 15-40% training data. This again demonstrates performance improvement in terms of saving a large amount of training data as compared to Method 3. In addition, observe that the percentage of training data required after TL under each case of Method 2 (location-free) is almost the same as Method 1. The above observation highlights that the presence of a small number of sensor measurements in the new (target) indoor wireless environments is sufficient to effectively estimate radio maps for Method 1 and Method 2.

C.4.4 Comparison of Reliability

To understand the reliability of the proposed TL method, transferability, F1-score and accuracy [69] are calculated for both cases under each Method. The effectiveness of the proposed method in recommending a correct TL is characterised by transferability. The model test accuracy is characterised by F1-score. All these measures range from 0 to 1. The higher values of these measures suggest that the developed model is better in respective performance.

To this line, Figure C.5 presents the transferability, F1-score and accuracy [69] for each case under each Method along with Method 3 for comparison. Notice that the transferability, F1-score and accuracy under case A of both Method 1 and Method 2 are higher than its corresponding case B. In addition, 100% accuracy is achieved by Method 1 and Method 2 under case A, which is higher than the corresponding value under case B. The reason for having lower accuracy under case B of both Method 1 and Method 2 is due to the small correlation between the training epochs and the test MSE obtained after TL operation for the similarity measure. Therefore,

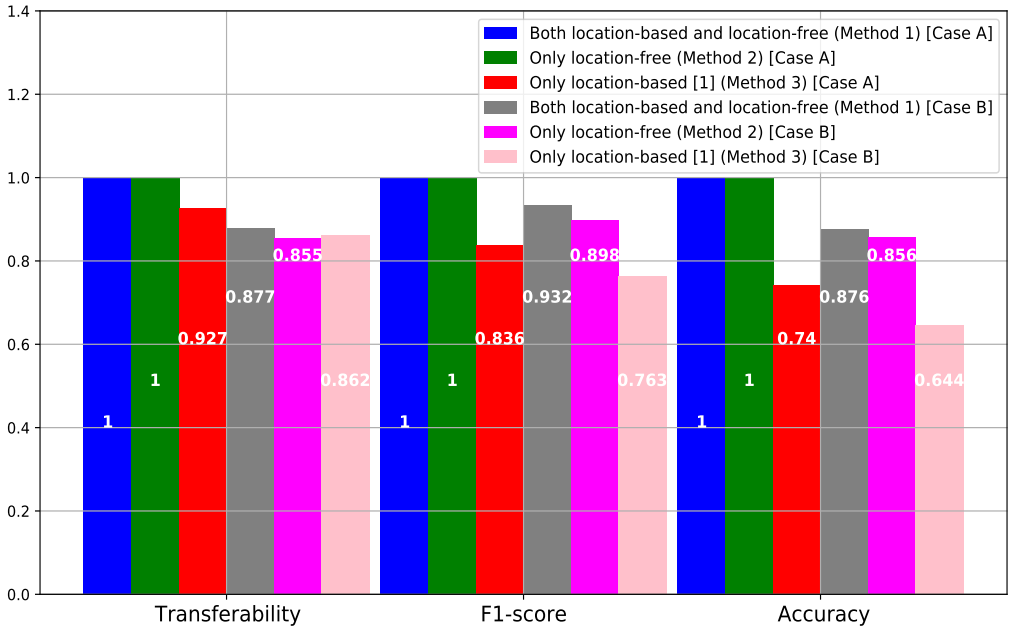


Figure C.5: Reliability comparison of different methods.

case A under both Method 1 and Method 2 performs better. On the other hand, the transferability, F1-score and accuracy under case B of Method 2 are slightly lower than the corresponding values under case B of Method 1.

Moreover, while comparing the transferability, F1-score and accuracy, in Figure C.5, under each case of Method 1 and Method 2 with Method 3, we notice that both cases under Method 1 and Method 2 are performing better than the corresponding cases under Method 3. In particular, there is an improvement of around 26% in accuracy under each case of Method 1 as compared to Method 3. Similarly, there is an improvement of around 24.5% in accuracy under each case of Method 2 (location-free) as compared to Method 3. In addition, the values of all measures under each case of Method 2 (location-free) are almost the same as Method 1.

Therefore, we can conclude from the results that the incorporation of ToA (location-free) features alone is satisfactory for estimating accurate radio maps in the new indoor wireless environments, which motivates using this method in scenarios with high wireless multi-path, where it may be difficult to have accurate location estimations.

C.4.5 Illustration of Radio Maps

We consider an indoor environment where the cube is present near the Tx. Figure C.6(b) shows the radio map for this environment without performing the TL operation. However, Figure C.6(c) and Figure C.6(d), respectively, illustrate the radio map for this same environment where only 40% of training data is needed while performing the TL operation using both ToA features and locations of Rxs (Method 1) and using only ToA features (Method 2). The solid black and dashed red colour, respectively, represent the cube and the walls of the rooms in the radio maps. It is clearly visible that the radio map estimated using only ToA features

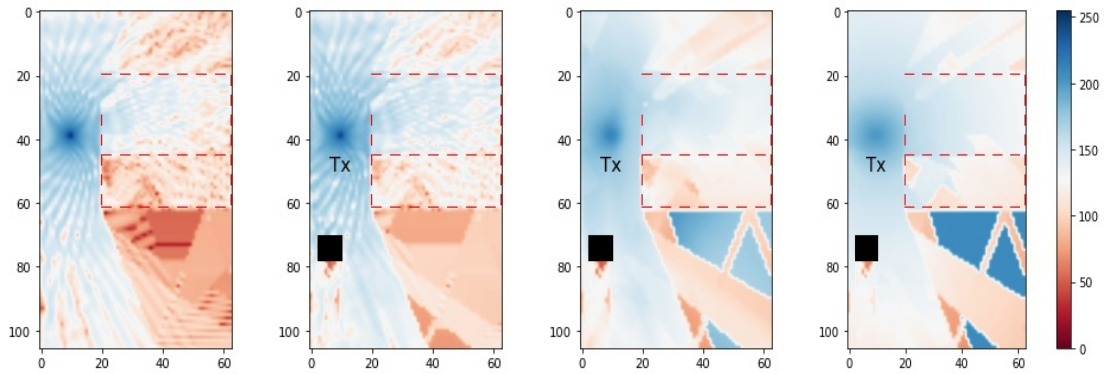


Figure C.6: Radio map for (a) original indoor wireless environment, (b) a modified indoor wireless environment where the cube is present near the Tx, (c) same environment of (b) after executing the operation of TL using both ToA features and locations of Rxs (Method 1), where only 40% of training data is used for training the DNN resulting in a test MSE after TL as 0.00203, and (d) same environment of (b) after executing the operation of TL using only ToA features (location-free) (Method 2), where only 40% of training data is used for training the DNN resulting in a test MSE after TL as 0.00368.

(location-free) (Method 2) is smoother and satisfactory. In fact, the corresponding test MSE is very similar to the one obtained when using both ToA features and locations of Rxs (Method 1).

C.5 Conclusion

A TL-based radio map estimation method for indoor wireless networks is presented in this paper. We first design a method (Method 1) where both ToA features and locations of Rxs are used, assuming that the exact locations of receivers are known. Then, we consider a method (Method 2) based only on the ToA features (location-free). Moreover, for establishing the similarity between two wireless environments, a data-driven similarity measure is developed. This similarity measure is later employed for making a decision regarding the recommendation to perform TL operation, given only the image of the new wireless environment. Additionally, the performance of Method 1 and Method 2 is also compared with Method 3 [34], which employs only the locations of receivers for estimating radio maps. Satisfactory performance is observed while employing only ToA features (location-free), which motivates using this method in scenarios with high wireless multi-path where it may be difficult to have accurate location estimations. The proposed TL method employing only ToA features (location-free) also significantly reduces the number of training samples as compared to the location-based method [34] in similar indoor wireless environments and hence, a large amount of sensor measurements are saved.

Appendix D

PAPER D

Title: Leveraging Transfer Learning for Radio Map Estimation via Mixture of Experts

Authors: **Rahul Jaiswal**, Mohamed Elnourani, Siddharth Deshmukh, Baltasar Beferull-Lozano

Journal: IEEE Transactions on Machine Learning in Communications and Networking (TMLCN) 2024 (Submitted)

Leveraging Transfer Learning for Radio Map Estimation via Mixture of Experts

R. Jaiswal, M. Elnourani, S. Deshmukh, B. Baltasar Lozano

Abstract: This paper leverages transfer learning (TL) on a mixture of experts (MoE) model for indoor radio map estimation. The proposed MoE combines location-based and location-free experts through a gating network exploiting their complementary benefits. To estimate the radio map in a new wireless environment, the learned model of another sufficiently similar environment is transferred and fine-tuned with additional data from the new environment. The proposed data-driven similarity measure predicts the amount of data needed for TL. Results demonstrate that the proposed method only requires 15-40% of measurements to adapt to several varying environments, and as expected, the proposed MoE method outperforms both experts.

D.1 Introduction

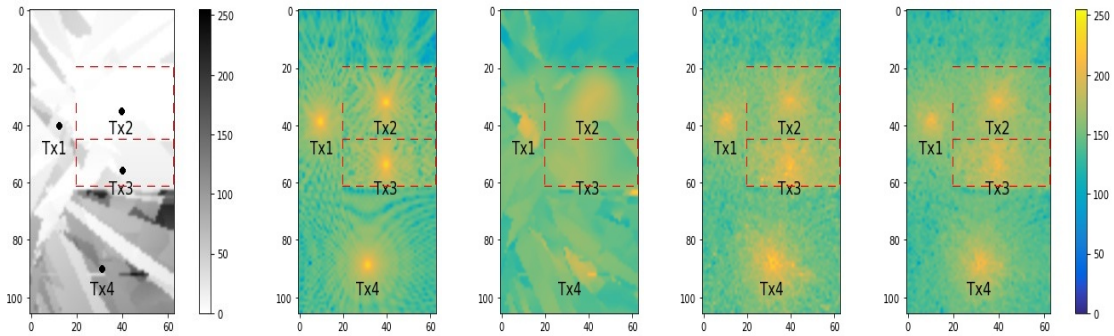
To improve the performance of future wireless communication systems, accurate estimation of radio maps is important. A radio map portrays an estimate of the power spectral density (PSD) over a required geographical area as a function of location, time, and frequency. It also depicts the various propagation characteristics of the wireless environment over a certain region, which is governed by various factors, such as reflections, diffractions, and scattering [38, 82]. In the case of multi-transmitter `Tx` environments, there are two types of radio maps. One that uses the aggregate power from all the different `Txs` which can be used for applications, such as network designing and planning. The other type uses the individual power of each `Tx` and can be used for applications, such as localization, and resource allocation. Additionally, this work focuses on data-driven radio map estimation for indoor wireless environments. However, the same method can also be applied to outdoor wireless environments. There have been some works related to radio map estimation in the literature. For example, a RadioUNet [3] model which is a modified UNet architecture [14] (originally designed for biomedical image processing) is adapted for estimating radio maps in urban environments. To estimate outdoor radio maps, a two-phase learning framework integrating the radio propagation model and designing a conditional generative adversarial network is proposed in [16]. A

maximum likelihood estimation-based indoor radio map estimation framework under a Gaussian quantizer is proposed in [17]. However, these works require training the model from the very beginning for each new wireless environment, resulting in the need for a large amount of sensor measurements.

In addition, the propagation characteristics of wireless environments may change over space, time, and frequency. Therefore, a radio map model developed under a given wireless environment (source environment), can not be deployed directly in a new wireless environment (target environment). Moreover, developing a new radio map model for each new wireless environment needs a large amount of sensing data. To tackle these issues, one can exploit transfer learning (TL) [19] where the knowledge acquired in the source environment for the learning task (in our case, learning a radio map model) can be used in the target environment when both the source and target environments are sufficiently similar. Some previous works on the exploitation of TL for wireless networks include [21, 23, 36]. TL has been employed to improve the robustness of deep neural network (DNN)-based spectrum sensing in cognitive radios [21]. A TL method is proposed for the joint resource allocation problem in [36]. Similarly, TL is also used for wireless fingerprinting localization in [23]. None of these works consider radio map estimation using TL.

To estimate radio maps, location-based algorithms [2,34] employ exact knowledge of receiver (`Rx`) locations. However, in practice, it is difficult to obtain accurate sensor locations, particularly, in environments having a large number of wireless multipaths. It is possible to estimate the approximate sensor locations from the time of arrival (ToA) features of radio signals using different types of algorithms, such as the trilateration localization [1]. This algorithm determines locations of `Rxs` based on one ToA value from each of the several `Txs` placed at known locations. On the contrary, one can also directly use ToA features to estimate radio maps without estimating the sensor locations, which is known as location-free algorithms [8]. It is also possible to incorporate both location-based and location-free algorithms, by using a mixture of experts (MoE) architecture. MoE [66] exploits the complementary advantages of each expert. MoE is exploited in [80] to estimate channel gain maps using kernel-based methods.

In this paper, in the context of radio map estimation, for a fixed time and frequency band, we consider a MoE that comprises two DNN-based experts. Expert 1 is location-based, trained with the estimated locations of `Rxs`. Expert 2 is location-free, trained with ToA features. The training of Expert 2 involves higher complexity due to the fact that it requires learning a higher dimension function. The gating network, which is also a DNN, combines the output of both experts to obtain the final radio map. To demonstrate the underlying motivation behind the use of the MoE for radio map estimation, we show in Figure D.1(a) the location-error (in meters) $\tilde{e}_{x_j^r} = ||x_j^r - \hat{x}_j^r||$ map, where x_j^r is the actual locations of `Rxs` and $\hat{x}_j^r$ is the estimated locations of `Rxs`, and the corresponding wireless environment radio map shown in Figure D.1(b). The location-error map is a gray-scale image that depicts the normalized error values of the sensor locations between 0 and 1. It is obtained by calculating, at each location, the Euclidean distance between the true and the



(a) (b) (c) (d) (e)

Figure D.1: (a) Location-error map for the original wireless environment (63×106 points with a spacing of 15 cm); Radio map for: (b) the original wireless environment with 4 `Txs`, (c) the same environment as in (b) estimated using Expert 1 (test MSE = 0.0051), (d) the same environment as in (b) estimated using Expert 2 (test MSE = 0.0050), and (e) the same environment as in (b) estimated using the MoE (test MSE = 0.0039).

estimated locations of `Rxs`. In this particular case, the estimated locations are obtained using the trilateration localization algorithm [1]. The walls of the rooms are represented in dashed red colour.

Additionally, Figure D.1 shows three radio maps. Figure D.1(c), Figure D.1(d), and Figure D.1(e) are the radio maps of the original wireless environment as shown in Figure D.1(b), estimated using Expert 1, Expert 2, and a MoE using a gating network, respectively. These radio maps are obtained by either using the power value outputs (Figure D.1(b)) or by training the respective Experts (Figure D.1(c)-(e)). The location-error map (Figure D.1(a)) depicts a significant amount of location errors in some areas, which can affect the accurate radio map estimation. In general, the performance of Experts 1 and 2 varies with the locations of `Rxs`. As we describe later in Section D.2, it is observed that the MoE based on the gating network, provides an improvement in the accuracy of the radio map (see Figure D.1(e)), as it incorporates both location-based measurements and location-free features, along with the location-error information. This motivates us to design a data-driven MoE model, which can adapt to location errors and experts' inaccuracies in order to estimate accurate radio maps. The main contributions of this paper are:

- Design of an effective data-driven transfer learning method based on a MoE-based radio map model learned from an original wireless environment that is transferred and fine-tuned to other similar wireless environments.

- Adapting our data-driven similarity measure, proposed in [2, 34], which is then used to decide whether TL is performed or not.

- Prediction of the amount of training data needed in new wireless environments when performing the TL operation.

- Extensive testing of the proposed TL scheme using simulated data from the Remcom simulator [33]. Results show that the proposed scheme performs an effective TL for estimating radio maps in most wireless environments while using a relatively small amount of training data. Numerically, the proposed scheme outperforms the individual location-based and the location-free experts [2, 8].

D.2 Problem Formulation

In this work, we tackle three problems, namely, the design of a MoE, the TL operation, and the TL similarity measure. Firstly, consider two separate experts. Expert 1 is a location-based having the estimated locations of `Rxs` $\{\hat{x}_j^r\}$ as input (where j is the index of the `Rx`), and $F_1(\hat{x}_j^r) = \hat{P}(\hat{x}_j^r)$ as output, the corresponding aggregate power[1] (in dBm), or a vector of all the individually received powers from each `Tx`. Similarly, Expert 2 is location-free having the estimated ToAs $\{\hat{\tau}_j^r\}$ as input, and $F_2(\hat{\tau}_j^r) = \hat{P}(\hat{\tau}_j^r)$ as output, the corresponding aggregate power or a vector of all the individually received powers from each `Tx`. Furthermore, we consider using the MoE principle [66], we can combine both experts using a gating network (G). The input to the gating network is $\{x_{GN_j}\} = \{\hat{P}(\hat{x}_j^r), \hat{P}(\hat{\tau}_j^r), \hat{\tau}_j^r, \hat{x}_j^r, e_{x_j^r}\}$, where $\hat{P}(\hat{x}_j^r)$, and $\hat{P}(\hat{\tau}_j^r)$ are the predicted receiver powers from Expert 1 and Expert 2, respectively, and $e_{x_j^r}$ is the location estimation error, assumed to be known. In practice, $e_{x_j^r}$ can be either estimated by performing location estimation measurements in advance, e.g., varying the wireless environment and estimating (offline) $\tilde{e}_{x_j^r}$ at each position of the potential receiver and calculating the average of these errors across environments $e_{x_j^r} = \mathbb{E}_{\text{environments}}\tilde{e}_{x_j^r}$, or through a localization algorithm that is able to estimate this error. This error generally defines an uncertainty region around the estimated

[1]In our experiment, we use aggregate power. However, the same formulation holds for both, that is, aggregate power and individual power radio maps.

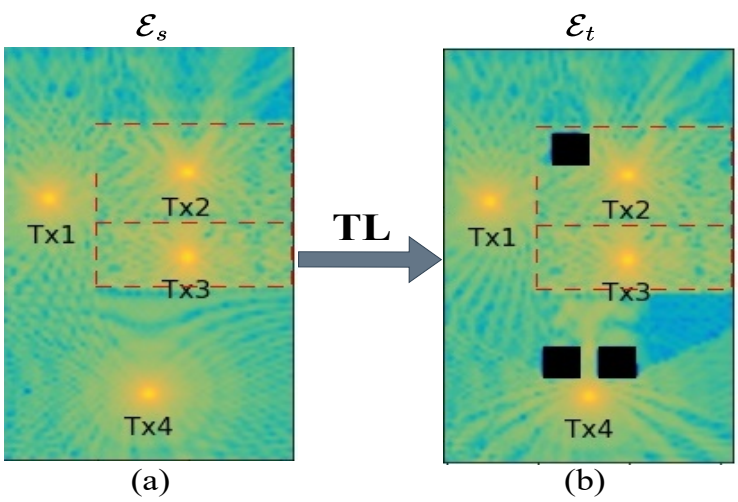


Figure D.2: Illustration of radio maps: (a) original wireless environment, and (b) target wireless environment with `4 Txs` and 3 cube objects represented by 3 black squares.

location where the actual location resides. The output of the gating network is $G(x_{GN_j}) = \hat{P}(x_{GN_j})$, that is, the aggregate received power at $\texttt{Rx}_j$. Our first problem is how to find the functions $F_1(\cdot)$, $F_2(\cdot)$, and $G(\cdot)$.

To understand the second problem, let $\mathcal{E}_s$, and $\mathcal{E}_t$ denote the source and target environments, respectively, as shown in Figure D.2. Next, let us assume that only a small amount of power values are available in $\mathcal{E}_t$, as compared to $\mathcal{E}_s$.

For the source environment $\mathcal{E}_s$, we first learn the DNN-based MoE $G_s(\cdot)$ with weight parameters $\theta_s^\star$ through minimizing the loss function mean square error (MSE), $\text{MSE} = \frac{\sum_{i=1}^{N_s}[P(x_i)-G_s(\cdot)]^2}{N_s}$, where N_s is number of $\texttt{Rxs}$. Let us also assume that $\mathcal{E}_s$ and $\mathcal{E}_t$ have some similarity in terms of wireless propagation characteristics. We perform TL by transferring the baseline MoE model learnt in $\mathcal{E}_s$ to $\mathcal{E}_t$, and initialize its parameters by the values obtained from the baseline MoE model of $\mathcal{E}_s$. Then, we can fine-tune this model for minimising the same loss function but in the target environment $\mathcal{E}_t$, by using only fewer additional measurement samples. The detailed mathematical formulation of TL can be seen in [2].

Notice that one should have a similarity measure between the source $\mathcal{E}_s$ and target $\mathcal{E}_t$ environments because it affects the TL performance. This is our third problem. The more similar $\mathcal{E}_s$ and $\mathcal{E}_t$ are, the more effective is expected TL to be. We have also tested classical similarity measures [2], such as scale-invariant feature transform[2] (SIFT) [56], and widely used Wasserstein distance[3] (WD) [51], and found them inadequate for the application of TL for radio map estimation [2]. The reason is that these similarity measures can be applied to any type of image representing the source environment $\mathcal{E}_s$ and they do not consider the radio environment propagation properties. Thus, we design our data-driven similarity measure (DDS) that incorporates and learns the wireless propagation characteristics from the data.

D.3 Proposed Radio Map Estimation Model

The proposed model comprises two phases, namely; the training phase and the execution phase, as shown in Figure D.3 and Figure D.4. The training phase consists of two stages: (i) the design of the baseline MoE model in source environment $\mathcal{E}_s$, followed by the possibility of transferring it to target environment $\mathcal{E}_t$, and (ii) the establishment of a similarity measure between $\mathcal{E}_s$ and $\mathcal{E}_t$. During the execution phase, the image of a test wireless environment is given as input to the DDS. The DDS predicts different MSE values and the number of training epochs for the TL operation at different data splits (number of training samples). Next, in the TL

[2]SIFT [56] is a scale, rotation and affine invariant 2D feature detector algorithm based on the difference-of-Gaussians (DoG) operator [83]. It detects feature points by searching the local maxima at various scales of the subject image. Feature matching is then done using $L1$-norm and setting a threshold.

[3]WD [51] calculates the distance between two probability distributions. We calculate the WD between the floor image corresponding to the $\mathcal{E}_s$ when the cube is positioned at one location, and the different floor images corresponding to the remaining $\mathcal{E}_t$ when the cube is positioned at different locations.

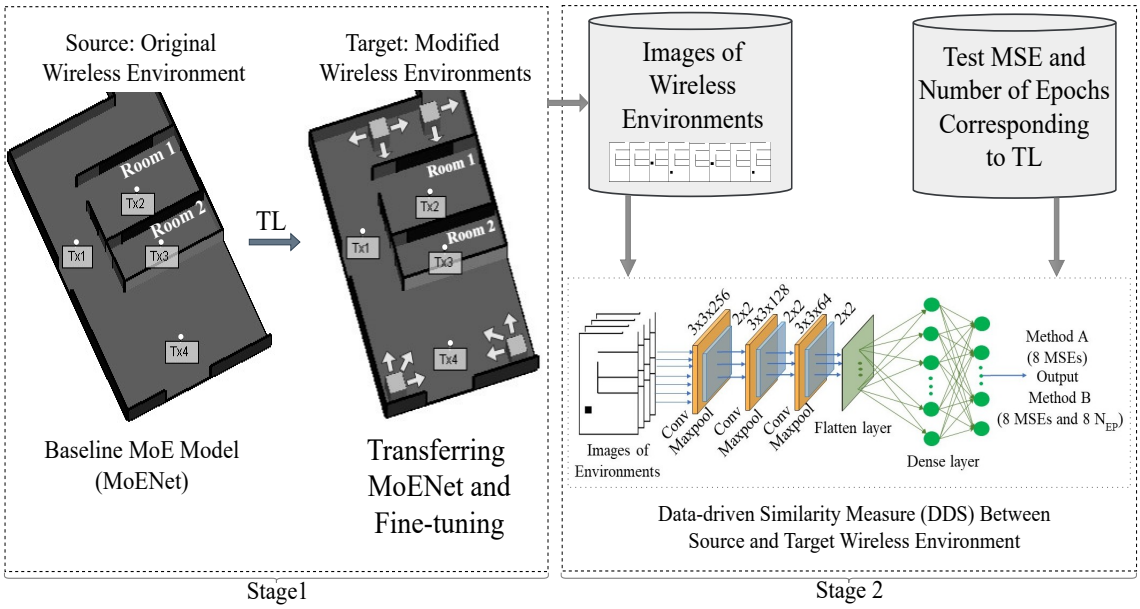


Figure D.3: Data-driven TL-based radio map estimation model using MoE (training phase).

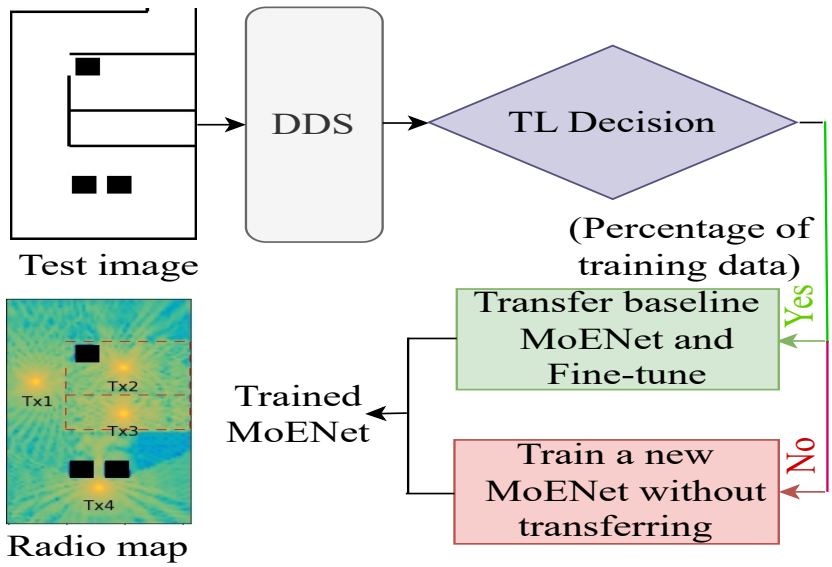


Figure D.4: Data-driven TL-based radio map estimation model using MoE (execution phase).

decision block, these values are compared with the set MSE and training epochs thresholds to decide further whether to perform TL operation with a specific data split. If the thresholds are satisfied, then the TL can be performed with that specific data split, else, TL will not be performed and *MoENet* needs to be trained from the very beginning.

D.3.1 Baseline MoE Model and Transfer Learning Approach

We first design a MoE model which comprises two experts and a gating network, as described in Section D.2. Both experts and the gating network are trained jointly as

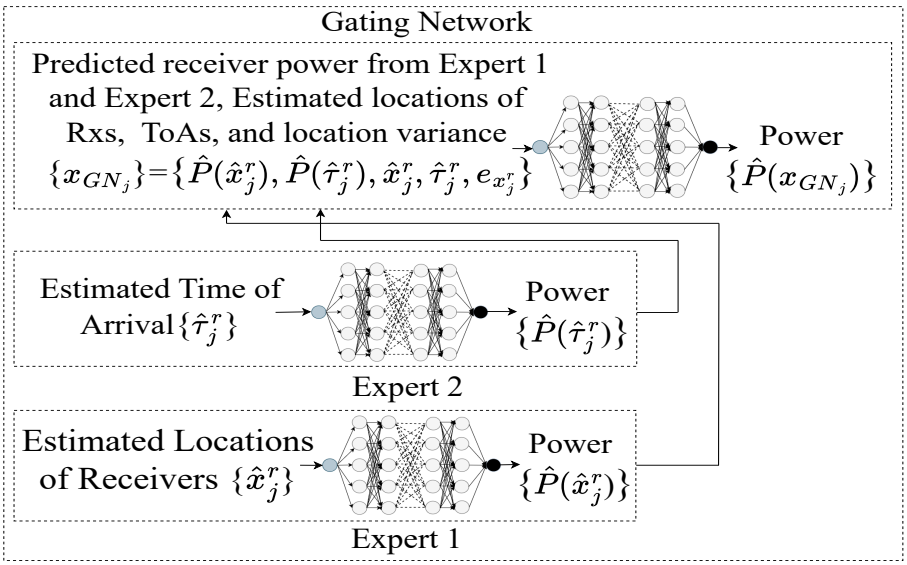


Figure D.5: MoENet: Baseline MoE model.

a feed-forward DNN for the original wireless environment (see Figure D.2(a)). We refer to the baseline MoE model (shown in Figure D.5) as *MoENet*.

A high-level description of the *MoENet* in the form of pseudo-code is presented in **Algorithm 8**.

Next, given the baseline MoE model *MoENet*, we first decide whether to perform the TL operation on the basis of the similarity measure. However, since our similarity measure is data-driven, a sufficient amount of data from a number of environments is necessary to learn this similarity measure. To this end, we transfer the *MoENet* to each target environment $\mathcal{E}_t$ individually, followed up by fine-tuning it using a small amount of additional training data from $\mathcal{E}_t$. The required training data for the target environments is varied from 5% to 40% of the amount of training data that is necessary for learning the *MoENet* from $\mathcal{E}_s$. After training and fine-tuning the transferred *MoENet* for each new environment $\mathcal{E}_t$, we store the test MSE($\mathcal{E}_t$) and the number of training epochs $N_{EP}(\mathcal{E}_t)$. The training is stopped after satisfying an early stopping [74] convergence criterion, that is, when the MSE is approximately stable for a certain number of consecutive iterations.

A high-level description of the TL method in the form of pseudo-code is presented in **Algorithm 9**.

D.3.2 Similarity Measure

Similar to the work in [2, 34], the DDS incorporates explicitly the TL effect in terms of MSE and the number of training epochs, along with the images of the environments. These are required to achieve a certain accuracy (MSE) of the radio map estimation in $\mathcal{E}_t$ and a certain number of training epochs (time complexity). For this, we train a convolutional neural network (CNN) [50] regression model under two distinct loss functions, one for accuracy and one for both accuracy and time complexity. Loss function A: $L_A = |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t|^2$ is the squared difference

Algorithm 8: Baseline DNN model *MoENet* with three jointly trained DNNs

Input: $D := \{\hat{x}_i, \hat{\tau}_i, e_{x_i}, P_i\}$: dataset of n estimated locations of `Rxs` $\hat{x}_i$ and n estimated ToAs $\hat{\tau}_i$, location estimation error at same `Rxs` e_{x_i} , and power values P_i

B: Batch size, α: Learning rate, N_{EP}: Number of epochs

Output: $\theta\mathbf{1}$: Trained DNN1 model parameters, $\theta\mathbf{2}$: Trained DNN2 model parameters, $\theta\mathbf{3}$: Trained DNN3 model parameters, MSE1: MSE of trained DNN1 based on test data, MSE2: MSE of trained DNN2 based on test data, MSE3: MSE of trained DNN3 based on test data

Data-Splitting: Splitting data into D_{train} and D_{test}

Data-division for three different DNN networks: $D1 := \{\hat{x}_i, P_i\}$, $D2 := \{\hat{\tau}_i, P_i\}$, $D3 = \{x_{GN_i}, P_i\} := \{\hat{P}(x_i), \hat{P}(\tau_i), \hat{x}_i, \hat{\tau}_i, e_{x_i}, P_i\}$, where $\hat{P}(x_i), \hat{P}(\tau_i)$ are the output of DNN1 and DNN2, respectively

`Training stage`
Randomly initialize the network parameters
for k = 1,...,N_{EP} **do**
  for j = 1,...,N_B = ceil($|D1_{\text{train}}, D2_{\text{train}}, D3_{\text{train}}|/B$) **do**
    Randomly select B training samples from training data $D1_{\text{train}}, D2_{\text{train}}, D3_{\text{train}}$ as the training batch
    Update DNN1, DNN2, & DNN3 parameters $\theta1, \theta2, \theta3$ with learning rate α to minimize $Losses = \text{MSE1} = \frac{\sum[P_i - \hat{P}^{\theta1}(x_i)]^2}{|B|}$, $\text{MSE2} = \frac{\sum[P_i - \hat{P}^{\theta2}(\tau_i)]^2}{|B|}$, $\text{MSE3} = \frac{\sum[P_i - \hat{P}^{\theta3}(x_{GN_i})]^2}{|B|}$
  end for
end for
`Testing stage`
Initialize MSE1, MSE2, MSE3: $\text{MSE1} \leftarrow 0, \text{MSE2} \leftarrow 0, \text{MSE3} \leftarrow 0$
Use test data $D1_{\text{test}}, D2_{\text{test}}, D3_{\text{test}}$
Predict power values $\hat{P}(x_i)$ on given $\{x_i\}$, $\hat{P}(\tau_i)$ on given $\{\tau_i\}$, and $\hat{P}(x_{GN_i})$ on given $\{x_{GN_i}\}$
Calculate $\text{MSE1} = \frac{\sum[P_i - \hat{P}^{\theta1}(x_i)]^2}{|D1_{\text{test}}|}$, $\text{MSE2} = \frac{\sum[P_i - \hat{P}^{\theta2}(\tau_i)]^2}{|D2_{\text{test}}|}$, $\text{MSE3} = \frac{\sum[P_i - \hat{P}^{\theta3}(x_{GN_i})]^2}{|D3_{\text{test}}|}$
Save trained DNN1, DNN2, and DNN3 models parameters $\theta\mathbf{1}, \theta\mathbf{2}, \theta\mathbf{3}$.

between the actual test MSE obtained from the TL, denoted by $\text{MSE}(\mathcal{E}_t)$, and the predicted test MSE, denoted by $\widehat{\text{MSE}}_t$, which is obtained when performing the TL. Loss function B: $L_B = ||[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t, N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]||^2$ contains both the test MSE and the number of training epochs which specify the power-value errors and the need for training time to perform TL, respectively. Note that $N_{EP}(\mathcal{E}_t)$, and $\widehat{N_{EP_t}}$ are the actual and predicted TL training epochs, respectively. Based on these loss functions, we propose two methods.

Method A: This method comprises CNN (see Figure D.6) having the images of distinct target environments $\mathcal{E}_t$ as input, and eight sets of test MSEs, i.e., $MSE(\mathcal{E}_t)$

Algorithm 9: TL Method

Input: D_t: data of target environment $\mathcal{E}_t$, $MoENet$: baseline MoE model $\theta1^\star_s, \theta2^\star_s, \theta3^\star_s$,
α: Learning rate, B: Batch size, r: Splitting ratio, N_{EP}: Number of epochs

Output: $MoENet$: DNN model $\theta1^\star_t, \theta2^\star_t, \theta3^\star_t$ for target environment $\mathcal{E}_t$, MSE1($\mathcal{E}_t$): MSE of transferred DNN1 based on the test data, MSE2($\mathcal{E}_t$): MSE of transferred DNN2 based on the test data, MSE3($\mathcal{E}_t$): MSE of transferred DNN3 based on the test data, $N_{EP}(\mathcal{E}_t)$: Number of training epochs for the jointly transferred DNN models

Data-Splitting: Splitting data into training $D1^t_{\text{train}}, D2^t_{\text{train}}, D3^t_{\text{train}}$ and test $D1^t_{\text{test}}, D2^t_{\text{test}}, D3^t_{\text{test}}$ with splitting ratio r, as shown in **Algorithm 8**

1: `Fine-Tuning and Testing stage`
2: Initialize MSE1($\mathcal{E}_t$), MSE2($\mathcal{E}_t$), MSE3($\mathcal{E}_t$):
MSE1($\mathcal{E}_t$) $\leftarrow 0$, MSE2($\mathcal{E}_t$) $\leftarrow 0$, MSE3($\mathcal{E}_t$) $\leftarrow 0$
3: Initialize $N_{EP}(\mathcal{E}_t)$: $N_{EP}(\mathcal{E}_t) \leftarrow 0$
4: `Fine-Tuning stage`
5: **for** k = 1,...,N_{EP} **do**
6: **for** j = 1,...,N_B = ceil($|D1^t_{\text{train}}, D2^t_{\text{train}}, D3^t_{\text{train}}|/B$) **do**
7: Randomly select B training samples from training data $D1^t_{\text{train}}, D2^t_{\text{train}}, D3^t_{\text{train}}$ as the training batch
8: Update DNN parameters $\theta1_t$, $\theta2_t$, and $\theta3_t$ with α to minimize $Losses = \text{MSE1} = \frac{\sum[P_i - \hat{P}^{\theta1}(x_i)]^2}{|B|}, \text{MSE2} = \frac{\sum[P_i - \hat{P}^{\theta2}(\tau_i)]^2}{|B|}, \text{MSE3} = \frac{\sum[P_i - \hat{P}^{\theta3}(x_{GN_i})]^2}{|B|}$
9: **end for**
10: Check early stopping criterion and stop when satisfied
11: **end for**
12: Save number of training epochs $N_{EP}(\mathcal{E}_t)$
13: `Testing stage`
14: Predict power values $\hat{P}(x_i)$, $\hat{P}(\tau_i)$, and $\hat{P}(x_{GN_i})$ on given $\{x_i\}$, $\{\tau_i\}$, and $\{x_{GN_i}\}$, respectively
15: Calculate MSE1($\mathcal{E}_t$) $= \frac{\sum[P_i - \hat{P}^{\theta1}(x_i)]^2}{|D1^t_{\text{test}}|}$, MSE2($\mathcal{E}_t$) $= \frac{\sum[P_i - \hat{P}^{\theta2}(\tau_i)]^2}{|D2^t_{\text{test}}|}$, MSE3($\mathcal{E}_t$) $= \frac{\sum[P_i - P^{\theta3}(x_{GN_i})]^2}{|D3^t_{\text{test}}|}$

obtained after performing TL with different data splits (5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%), as output. It uses the Loss function L_A and the corresponding learned CNN model is henceforth referred to as $SimNet_A$. Each floor image representing an environment is assumed to be a three-level image in which free space is represented as white (gray value 255), walls are assumed a gray value of 128, and objects (e.g., cubes) are black (gray value 0).

Method B: This method comprises CNN with the same input but whose output is now the combination of eight sets of test MSEs, i.e., $MSE(\mathcal{E}_t)$ obtained after performing TL with different data splits (5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%),

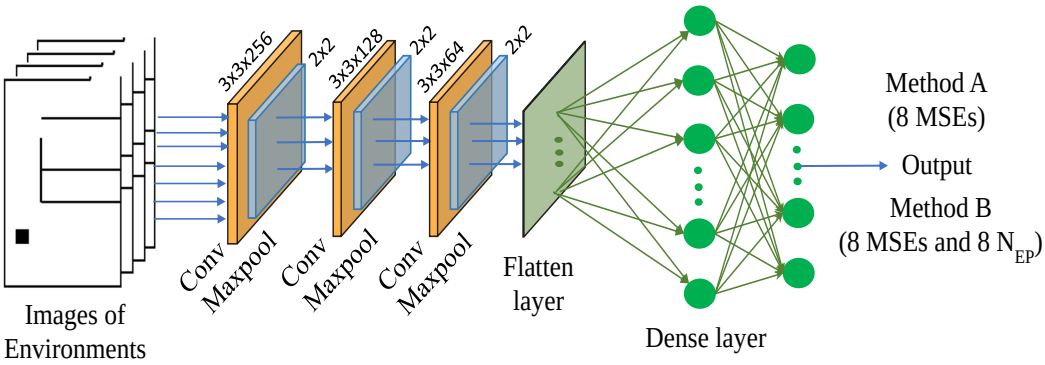


Figure D.6: CNN-based neural network structure for the similarity measure.

and eight sets of $N_{EP}(\mathcal{E}_t)$ of training epochs required for each target environment $\mathcal{E}_t$ when *MoENet* is transferred and re-trained. It uses the Loss function L_B and the corresponding learned CNN model is henceforth referred to as $SimNet_B$.

For any similarity measure, the goal is to decide whether TL is beneficial (in our case, small MSE($\mathcal{E}_t$), and small $N_{EP}(\mathcal{E}_t)$ (Method B)) or not between the two environments, based on certain thresholds. To select the value of these thresholds, we have to trade-off between the radio map quality, convergence time (Method B), and the recommendation rate[4] of the TL. In the case of our DDS, under Method A, we set a threshold for the test MSE (MSE$_{thr}$) and compare it with the predicted test MSE ($\widehat{MSE}(\mathcal{E}_t)$) obtained after TL. Similarly, under Method B, we set thresholds for both the test MSE (MSE$_{thr}$) and the number of training epochs ($N_{EP_{thr}}$). If both predicted values are less than these thresholds, then two radio maps are recognized as similar and TL can be performed effectively, otherwise, we assume that TL is ineffective. Note that, as shown in Figure D.6 of DSS, we give an image of the floor plan of a new target wireless environment as input and then map it to the MSE of the TL operation (MSE and N_{EP} in Method B).

A high-level description of DDS in the form of pseudo-code is presented in **Algorithm 10**.

D.4 Generation of Indoor Wireless Environments

To obtain power measurements at multiple `Rx` locations for distinct types of environments, we use the high accuracy ray-tracing X3D ray model [41], computed using Remcom [33]. We first consider an environment of a single floor with two rooms. We place $N_t = 4$ transmitters `Txs` at distinct locations $\{x_i^t\}_{i=1}^{N_t}$ and then `Rxs` at uniformly spaced locations $\{x_j^r\}$ with a 15 cm spacing vertically and horizontally to obtain the aggregate power values $\{P(x_j^r)\}$ at each `Rx` location. Additionally, we measure the first two ToAs of each `Tx`'s signals at each `Rx` location. We add Gaussian noise with zero mean and a variance of 1 dB to these ToAs. This results in (63×106) 6678 `Rx` locations and corresponding aggregate power values. Table D.1 summarizes

[4]The recommendation rate of TL is computed as the number of environments for which TL is recommended divided by total number of environments.

Algorithm 10: Data-driven Similarity Measure *SimNet*

Input: $D := \{\text{Im}_t, \text{MSE1}_t, \text{MSE2}_t, \text{MSE3}_t, N_{EP_t}$ (only for Method B)$\}$: dataset comprising three-level images of target environments, 8 sets of MSEs (MSE1, MSE2, MSE3) of the transferred model from **Algorithm 9** for different data splits $r_t = \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%\}$, Training epochs of the transferred model from **Algorithm 9** for all 8 sets of data splits, for each environment, respectively, MSE_{thr}: MSE threshold, $N_{EP_{thr}}$: Epoch threshold, r: Split ratio, N_{EP}^{cnn}: Training epochs of *SimNet*

Output: $SimNet(\theta_{sim})$, MSE^{cnn}: MSE of the trained CNN model, TL decision: Yes/No

Data-Splitting: Split into training D_{train} & test D_{test} with split ratio r

1: `Training stage`
2: Use data from D_{train}
3: Normalize image values Im_t, MSE1_t, MSE2_t, MSE3_t and N_{EP_t} to $[0, 1]$
4: Randomly initialize θ_{sim}
5: **for** k = 1,...,N_{EP}^{cnn} **do**
6: **for** j = 1,...,$N_B = \text{ceil}(|D_{\text{train}}|/B)$ **do**
7: Randomly select B training samples from D_{train} as the training batch
8: Update θ_{sim} with learning rate α to minimize *loss* given by the

$$\begin{aligned} \text{MSE}^{cnn} &= L_A = |\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t|^2 \quad \text{(Method A)} \\ &= L_B = |[\text{MSE}(\mathcal{E}_t) - \widehat{\text{MSE}}_t, \\ &\qquad N_{EP}(\mathcal{E}_t) - \widehat{N_{EP_t}}]|^2 \quad \text{(Method B)} \end{aligned} \tag{D.1}$$

9: **end for**
10: **end for**
11: `Testing stage`
12: Initialize MSE^{cnn}: $\text{MSE}^{cnn} \leftarrow 0$
13: Use test data from D_{test}
14: Predict MSE1_t, MSE2_t, MSE3_t and N_{EP_t} on a given image of Im_t
15: Calculate MSE^{cnn} using (D.1)
16: `TL decision`
17: **for** q = 1,...,$|D|$ **do**
18: **for** each r_t **do**
19: **if** max $(\widehat{\text{MSE1}}_t, \widehat{\text{MSE2}}_t, \widehat{\text{MSE3}}_t) \leq \text{MSE}_{thr}$ **then**
20: TL decision $\leftarrow$ Yes and output r_t, **else**, TL decision $\leftarrow$ No
21: **if** max $(\widehat{\text{MSE1}}_t, \widehat{\text{MSE2}}_t, \widehat{\text{MSE3}}_t) \leq \text{MSE}_{thr}$ & $\widehat{N}_{EP_t} \leq N_{EP_{thr}}$ (for Method B) **then**
22: TL decision $\leftarrow$ Yes and output r_t, **else**, TL decision $\leftarrow$ No
23: **end if**
24: **end if**
25: **end for**
26: **end for**

Table D.1: Parameters used in Remcom for generating the data

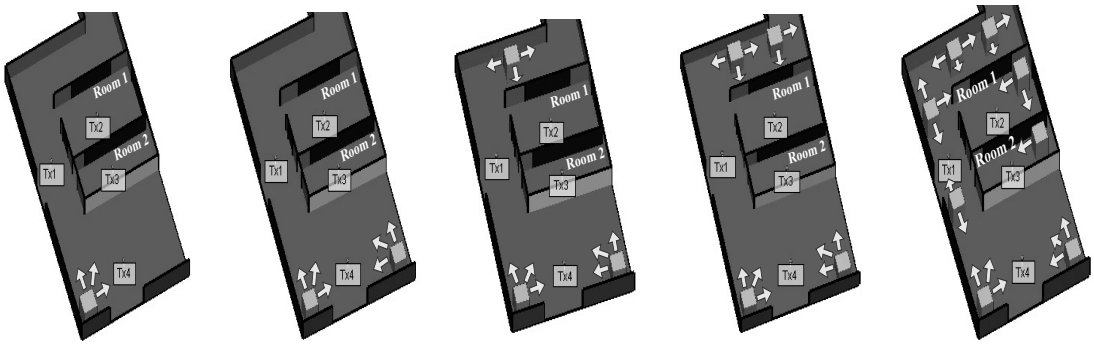
Waveform	Narrow-band Sinusoidal
Carrier frequency with Bandwidth	900 MHz (1.0 MHz)
Antenna type	Omni-directional
`Tx1` location `Tx2` location `Tx3` location `Tx4` location	(1.5, 10.0, 1.3) m (6.0, 11.0, 1.3) m (6.0, 7.75, 1.3) m (4.75, 2.5, 1.3) m
`Tx` height, and `Tx` power	1.3 m, and 27.73 dBm
Space between two `Rx` points and `Rx` routes	15 cm each
Ray tracing model, and Ray spacing	X3D ray model, and 0.2°
Reflections, transmissions, and diffractions	3, 2, 0
Floor dimension (width x length x height)	(9.5 x 20.6 x 2.88) m
Image size of floor (width x height)	(160 x 275) pixels

Table D.2: Summary of wireless environment $\mathcal{E}$.

Type of Environment $\mathcal{E}$	Description of $\mathcal{E}$	Number
Type I	`4 Txs`, 1 Cube	250
Type II	`4Txs`, 2 Cubes	252
Type III	`4 Txs`, 3 Cubes	248
Type IV	`4 Txs`, 4 Cubes	254
Type V	`4 Txs`, 8 Cubes	42
Total number of environments	1046	

the selected Remcom parameters with the radio map shown in Figure D.2 (a).

Next, we place physical objects, represented by cubes, at different locations to create distinct types of environments with 4 `Txs`. For the sake of simplicity, we consider objects represented by a single 1 m^3 metallic solid cube block, placed in the environment. We create 5 types of environments; *Type I*, *Type II*, *Type III*, *Type IV* and *Type V* environments where 1, 2, 3, 4 or 8 cubes are placed in the original wireless environment at various locations respectively. This results in 250, 252, 248, 254, and 42 environments of each type respectively, and a total of 1046 environments. Note that the location of each cube is changed horizontally, vertically, and diagonally with an incremental spacing of 15 cm to generate these environments. The image of each environment is saved as a 160 × 275 three-level image. These images are later used to measure the similarity between environments, as discussed in Section D.3.2. In each of these environments, the same Remcom parameters were used to generate the aggregate power at each `Rx` location. The summary of different types of wireless environments with the respective number of environments is presented in Table D.2 and shown in Figure D.7. Additionally, we generate 130 different wireless environments by making the following changes: (a) changing the carrier frequency of waveform from 900 to 950 MHz, (b) changing the location of



(a) (b) (c) (d) (e)

Figure D.7: Illustration of different types of wireless environments $\mathcal{E}$ (a) Type I $\mathcal{E}$ (`4 Txs`, 1 cube), (b) Type II $\mathcal{E}$ (`4 Txs`, 2 cubes), (c) Type III $\mathcal{E}$ (`4 Txs`, 3 cubes), (d) Type IV $\mathcal{E}$ (`4 Txs`, 4 cubes), and (e) Type V $\mathcal{E}$ (`4 Txs`, 8 cubes).

each `Tx` by 1 meter, (c) adding a wall below to second room, and (d) adding two objects in the floor plan and changing its locations horizontally, and vertically with a spacing of 15 cm.

D.5 Results and Discussions

This section discusses the results of the proposed model.

D.5.1 Numerical Analysis with Noiseless ToAs

In this experiment, we use the exact ToA features (noiseless). To optimize the baseline MoE model $MoENet$ (see Figure D.5) that we need before performing TL in the new target environment $\mathcal{E}_t$, grid search is performed, for the dataset described in Section D.4, with a variable number of hidden layers (5-10), neurons for each layer (8-1024), and learning rate range (0.0001-0.01). The average localization error $e_{x_j^r}$ is obtained numerically by using the trilateration algorithm [1] for all wireless environments and then calculating the average at each `Rx` location. The histogram of this error (in meters) is shown in Figure D.8.

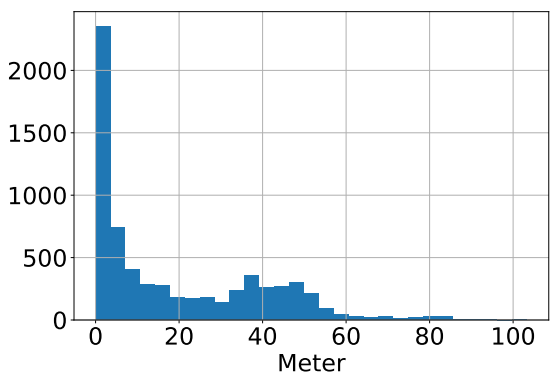


Figure D.8: Histogram of average location estimation error from all environments.

Table D.3: Parameters and hyper-parameters for $MoENet$

	Expert 1	Expert 2	Gating Network
Neurons in input layer	2	8	13
Number of hidden layers	7	5	3
Number of neurons in each hidden layer	1024, 512, 256, 128, 64, 32, 16	1024, 512, 256, 128, 32	256, 128, 64
Batch normalization	1 to 5 layer	-	-
Batch normalization	-	1 to 4 layer	-
Batch normalization	-	-	1 and 2 layer
Dropout after 3rd layer	-	-	0.2
Hidden and output layers activation function	ReLU		
Neurons in output layer	1		
Optimizer with learning rate	Adam (0.0001) [77]		
Loss function & Batch size	MSE, 64		
Training-Testing split ratio	80:20		
Iterations for early stopping	5		

With the parameters as in Table D.3, the best jointly trained $MoENet$ results in a training and a test MSE of 0.0048 and 0.0051 for Expert 1, 0.0031 and 0.0034 for Expert 2, and 0.0022 and 0.0027 for $MoENet$ gating network, respectively. This reflects that the $MoENet$ is indeed an accurate model for training data and also generalizes well for the test data. Based on our experiment, we have observed that the gating network only requires 3 hidden layers.

TL Performance Analysis: While training the CNN model $SimNet_A$ under Method A (see Figure D.6 and Table D.4), a training MSE and a test MSE of 0.0276, and 0.0315, respectively, are obtained. Similarly, while training the CNN model $SimNet_B$ under Method B, a training MSE and a test MSE of 0.0206, and 0.0234, respectively, are obtained. This reflects that both $SimNet_A$ and $SimNet_B$ are trained appropriately.

Table D.5 presents the TL decisions for both methods. Under Method A, 1031 out of 1046 (98.56%), and under Method B, 1034 out of 1046 (98.85%) different environments, respectively, satisfy our criteria and are recommended for TL, with a training data need in the range of only 15-40%.

SimNet Model Accuracy: To investigate the reliability of the proposed scheme, Table D.6 presents the transferability, F1-score, and accuracy [2]. The transferability represents how good the considered model is in recommending a correct TL for a new $\mathcal{E}_t$. The F1-score is a measure of the test accuracy of the model. Accuracy measures the degree of veracity of the model. Given true positive (TP), true negative (TN), false positive (FP), and false negative (FN), all these measures are defined as [69]:

$$\text{Transferability} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \ \text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{D.2}$$

Table D.4: Parameters and hyper-parameters for $SimNet_A$ and $SimNet_B$

	$SimNet_A$	$SimNet_B$
Number of convolutional layers	3	
Number of filters in each convolutional layer	256, 128, 64	
Filter size in each convolutional layer	3 x 3	
Number of max pooling layers	3	
Filter/pool size in each max pooling layer	2 x 2	
Dropout after each max pooling layer	0.2	
Number of strides in each convolutional layer	2	
Activation function in each convolutional layer	ReLU	
Neurons in 1st flatten layer of neural network	32	
Activation function in neural network	ReLU	
Dropout after 1st flatten layer of neural network	0.2	
Neurons in output layer of neural network	24	32
Activation fn. in output layer of neural network	ReLU	
Optimizer with learning rate	Adam (0.001)	
Loss function and Batch size	MSE, 32	
Training-Testing split ratio	80:20	
Threshold MSE_{thr} for MSE	0.01	
Threshold $N_{EP_{thr}}$ for number of training epochs	-	11

Table D.5: TL performance

	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
Necessary training data	Number of wireless environments	Number of wireless environments
15%	1	-
20%	55	30
25%	21	39
30%	515	381
35%	289	483
40%	150	101
TL Recommendation rate (in %)	1031/1046 (98.56)	1034/1046 (98.85)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{D.3}$$

Table D.6 reflects that the transferability under each method is the same, while the F1-score is slightly better in Method A. The accuracy under Method A, and Method B is 79.2% and 78.3%, respectively. The reason for observing a smaller accuracy under Method B, is because Method B is more strict towards satisfying both the test MSE and the number of training epochs threshold criteria, as compared

Table D.6: Transferability, F1-score, and Accuracy

Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	808	798
TN	21	22
FP	203	213
FN	14	13
Transferability	0.982	0.983
F1-score	0.881	0.875
Accuracy	0.792	0.783

to Method A, where only the test MSE criterion needs to be satisfied. This suggests that our proposed model can be used effectively to estimate radio maps in new wireless environments with fewer sensor measurements, as compared to the case of training a raw model without TL.

D.5.2 Numerical Analysis with Noisy ToAs

In this experiment, we add Gaussian noise with zero mean and a variance of 1 dB to the ToA features (noisy). With the parameters as in Table D.3, the best jointly trained $MoENet$ results in a training and a test MSE of 0.0048 and 0.0051 for Expert 1, 0.0028 and 0.005 for Expert 2, and 0.0031 and 0.0039 for $MoENet$ gating network, respectively.

TL Performance Analysis: While training the CNN model $SimNet_A$ under Method A (see Figure D.6 and Table D.4), a training MSE and a test MSE of 0.0201, and 0.0255, respectively, are obtained. Similarly, while training the CNN model $SimNet_B$ under Method B, a training MSE and a test MSE of 0.0277, and 0.032, respectively, are obtained. This reflects that both $SimNet_A$ and $SimNet_B$ are trained appropriately.

Table D.7: TL performance

	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
Necessary training data	Number of wireless environments	Number of wireless environments
15%	-	13
20%	27	22
25%	-	47
30%	128	116
35%	541	561
40%	297	263
TL Recommendation rate (in %)	933/1046 (94.93)	1022/1046 (97.7)

Table D.8: Transferability, F1-score, and Accuracy

Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	748	758
TN	53	24
FP	212	241
FN	33	23
Transferability	0.957	0.970
F1-score	0.859	0.851
Accuracy	0.765	0.747

Table D.7 presents the TL decisions for both methods. Under Method A, 933 out of 1046 (94.93%), and under Method B, 1022 out of 1046 (97.7%) different environments, respectively, satisfy our criteria and are recommended for TL, with a training data need in the range of only 15-40%.

SimNet Model Accuracy: Table D.8 reflects that the F1-score under each method is the same, while the transferability is slightly better in Method B. Method B has slightly lower accuracy than Method A. The accuracy under Method A is 76.5% and the accuracy under Method B is 74.7%. This again suggests that our proposed model can be used effectively to estimate radio maps in new wireless environments with fewer sensor measurements, as compared to the case of training a raw model without TL, even in the presence of noise in the ToA features.

D.5.3 Numerical Analysis with Noisy ToAs and Additional Complex Changes in Wireless Environments

As discussed in Section D.4, 130 different wireless environments are generated. These environments are combined with the 1046 environments, resulting in a total of 1176 target wireless environments. The TL operation is performed for all these environ-

Table D.9: TL performance

	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
Necessary training data	Number of wireless environments	Number of wireless environments
20%	55	61
25%	315	86
30%	271	429
35%	404	496
40%	105	101
TL Recommendation rate (in %)	1150/1176 (97.78)	1173/1176 (99.74)

Table D.10: Transferability, F1-score, and Accuracy

Performance measures	Method A Only MSE($\mathcal{E}_t$)	Method B Both MSE($\mathcal{E}_t$) and $N_{EP}(\mathcal{E}_t)$
TP	833	808
TN	20	8
FP	305	357
FN	18	3
Transferability	0.978	0.996
F1-score	0.837	0.817
Accuracy	0.725	0.693

ments while keeping both MSE and training epoch thresholds the same.

TL Performance Analysis: Table D.9 presents the TL decisions for both methods. Under Method A, 1150 out of 1176 (97.78%), and under Method B, 1173 out of 1176 (99.74%) different environments, respectively, satisfy our criteria and are recommended for TL, with a training data need in the range of only 20-40%.

SimNet Model Accuracy: Table D.10 reflects that the transferability is slightly better in Method B. However, the F1-score and accuracy under Method A are slightly better than Method B. The accuracy under Method B is 69.3% and the accuracy under Method A is 72.5%.

D.5.4 Performance Comparison

While comparing the performance of the proposed TL model with the noiseless and noisy ToA features (see Table D.6 and Table D.8), it can be observed that the overall model accuracy slightly decreases with the presence of noise (79.2% with noiseless and 76.5% with noisy ToAs). However, both scenarios have satisfactory performance.

Similarly, while comparing the performance of the proposed TL model with noisy ToAs and with and without changes in the wireless environments (see Table D.8 and Table D.10), it can be observed that the TL Recommendation rates under both scenarios are almost the same. However, the overall model accuracy slightly decreases when making complex changes in the wireless environments (76.5% without change and 72.5% with change in environment). This signifies that our proposed TL model is adaptable to more complex changes (e.g., changes in carrier frequency, the height of antennas, adding an extra wall, etc.) in wireless environments.

D.5.5 Choice of Thresholds

We have estimated experimentally the minimum test MSE threshold (MSE_{thr}) that allows a particular expert to perform TL for estimating accurate radio maps of new environments. A very high threshold value will result in a less accurate radio map estimation. However, if we choose a very low threshold value, then it will result

in a better radio map estimation, but a lower TL recommendation rate. Moreover, it should be noticed that the choice of these thresholds will also depend on the application scenario. Additionally, the amount of computation time required for executing a MoE model, which incorporates three DNNs inside it, will be always higher than the computation time required for executing an individual expert, such as Expert 1 or Expert 2, which incorporates only a single DNN. To this end, we experimentally selected a MSE_{thr} of 0.0098 for Expert 1, 0.0079 for Expert 2, and 0.0075 for the MoE gating network, respectively. Notice that our overall proposed model requires a MSE_{thr} of 0.01 to perform effective TL operation.

D.5.6 Estimated Radio Maps

To visualize the radio maps, Figure D.9 and Figure D.10 depict five different radio maps of different wireless environments with and without TL operation. The cubes and the walls of the rooms are represented in black and dashed red colours, respectively. The differences in the radio maps among the five distinct scenarios can be easily visualized.

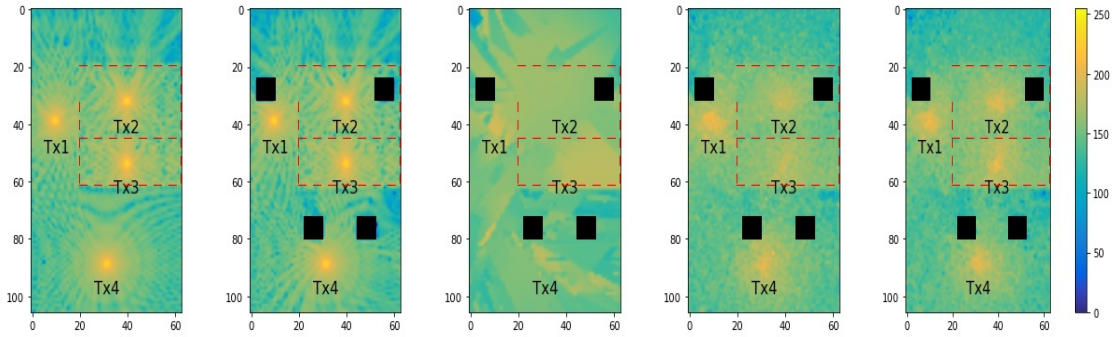


Figure D.9: Radio map for (a) original wireless environment with `4 Txs`, (b) an environment from Type IV, (c) environment (b) after performing TL using Expert 1 with 40% training data ($\text{MSE}(\mathcal{E}_t) = 0.0078$), (d) environment (b) after performing TL using Expert 2 with 40% training data ($\text{MSE}(\mathcal{E}_t) = 0.006$), and (e) environment (b) after performing TL using *MoENet* with 40% training data ($\text{MSE}(\mathcal{E}_t) = 0.0054$).

D.6 Conclusions

This paper presents a mixture-of-expert (MoE) based transfer learning method for estimating radio maps in wireless networks. The MoE comprises a location-based and a location-free expert. A gating network supervises these two experts. To quantify the similarity between two wireless environments, a data-driven similarity measure is adapted and later used to perform TL operation for a given image of the wireless environment. Results demonstrate that the proposed MoE-based TL method performs efficiently in estimating radio maps for the new wireless environments while requiring fewer amount of training samples, and outperforms both ex-

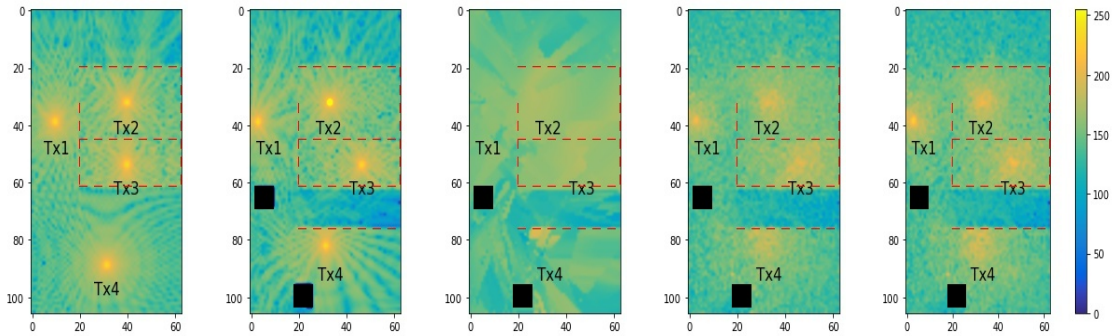


Figure D.10: Radio map for (a) original wireless environment with `4 Txs`, (b) an environment with additional wall and changed `Tx` locations, (c) environment (b) after performing TL using Expert 1 with 30% training data (MSE($\mathcal{E}_t$) = 0.0096), (d) environment (b) after performing TL using Expert 2 with 30% training data (MSE($\mathcal{E}_t$) = 0.0076), and (e) environment (b) after performing TL using *MoENet* with 30% training data (MSE($\mathcal{E}_t$) = 0.0071).

perts. Also, our proposed method performs well in the presence of noise in the time of arrival features and when making complex changes in the wireless environments.

Appendix E

PAPER E

Title: Transfer Learning Based Joint Resource Allocation for Underlay D2D Communications

Authors: **Rahul Jaiswal**, Siddharth Deshmukh, Mohamed Elnourani, Baltasar Beferull-Lozano

Conference: IEEE WCNC 2022

Transfer Learning Based Joint Resource Allocation for Underlay D2D Communications

R. Jaiswal, S. Deshmukh, M. Elnourani, B. Baltasar Lozano

Abstract: In this paper, we investigate the application of transfer learning to train a Deep Neural Network (DNN) model for joint channel and power allocation in underlay D2D communication. Based on the traditional optimization solutions, generating training datasets for scenarios with perfect channel state information (CSI) is not computationally demanding, compared to scenarios with imperfect CSI. Thus, a transfer learning-based approach can be exploited to transfer the DNN model trained for the perfect CSI scenarios to the imperfect CSI scenarios. We also consider the issue of defining the similarity between two types of resource allocation tasks. For this, we first determine the value of outage probability for which two resource allocation tasks are the same, that is, for which our numerical results illustrate the minimal need for relearning from the transferred DNN model. For other values of outage probability, there is a mismatch between the two tasks and our results illustrate a more efficient relearning of the transferred DNN model. Our results show that the learning dataset required for relearning of the transferred DNN model is significantly smaller than the required training dataset for a DNN model without transfer learning.

E.1 Introduction

In the last decade, research efforts in underlay D2D communication have categorically focused on devising judicious resource allocation algorithms, as it is fundamental for the efficient performance of both cellular and D2D networks [84–86]. Judicious resource allocation in the form of prudent power control and channel assignment to D2D pairs is vital to limit interference. Its main goal is to maximize performance metrics such as sum rate, energy efficiency, spectral efficiency etc. while satisfying the desired quality of service (QoS) requirements [84, 86]. Unfortunately, most formulations on resource allocation lead to either mixed-integer nonlinear problems or highly non-convex problems, which in general involve combinatorial complexity for obtaining the optimal solution. Thus, convex relaxation approaches can be exploited to obtain the sub-optimal solution with consideration of (i) optimality gap, (ii) convergence guarantees, and (iii) computation complexity.

In order to practically realize the resource allocation solution, recent research works have proposed training of Deep Neural Networks (DNNs) to a close-to-optimal solution. Motivation for exploiting DNN is primarily due to its universal approximation capability [87] and supplemented by the fact that trained DNN models are computationally very simple [88] to execute. In this context, deep learning-based resource allocation algorithms have been proposed in [89, 90]. However, these deep learning-based resource allocation algorithms are only implementing power control and power allocations, but not the joint resource allocation of both power allocation and channel assignments, as presented in the perfect CSI [70] and imperfect CSI [72] cases. Moreover, these previous works [89, 90] are focused on perfect CSI and imperfect CSI scenario of resource allocation in terms of power control and power allocation respectively but do not include any transfer learning (TL) approach in order to either improve the learning performance or accelerate the training, saving computational resources.

Notice that the success of DNN models in replicating different optimization-based resource solutions relies heavily on the availability of sufficiently large learning datasets. The generation of large datasets depends on the computational complexity of the original resource allocation solution; for example, under the perfect CSI scenario, [70] decouples without loss of optimality, the resource allocation problem into multiple power allocation subproblems and a channel assignment subproblem. In [70], the power allocation subproblems have closed-form solutions and the channel assignment subproblem is solved by integer relaxation. Thus, in this case, it is possible to obtain a large training dataset with reasonable complexity. However, if we consider a similar set-up with imperfect CSI [72], the decoupled power allocation sub-problems are solved iteratively using fractional programming, making the generation of large learning datasets cumbersome work.

The resource allocation under perfect and imperfect CSI can be considered as similar tasks and in this paper, we investigate how to exploit the concept of transfer learning (TL) in order to address the problem of learning with small datasets. Transfer learning [19] is a promising technique in which DNN models trained for one task can be transferred to another similar task which has less learning data. Some related research work on TL includes robust sensing framework [21], and transfer learning via self-imitation for resource allocation [32].

In this work, we address the problem of learning resource allocation in an imperfect CSI scenario [72] by exploiting the TL from the perfect CSI scenario. We first characterize the similarity between the two resource allocation tasks in terms of the outage probability. Next, we train a DNN model under the perfect CSI scenario with the dataset generated from the algorithm presented in [70] and define it as a baseline model for TL. The providing baseline model is then retrained for the case of imperfect CSI with a small dataset generated from the algorithm presented in [72]. Our numerical results illustrate that as the mismatch between two tasks increases, a larger dataset is required for relearning the baseline model; however, the amount of dataset required for relearning the baseline model is significantly smaller, compared to training of the new DNN model.

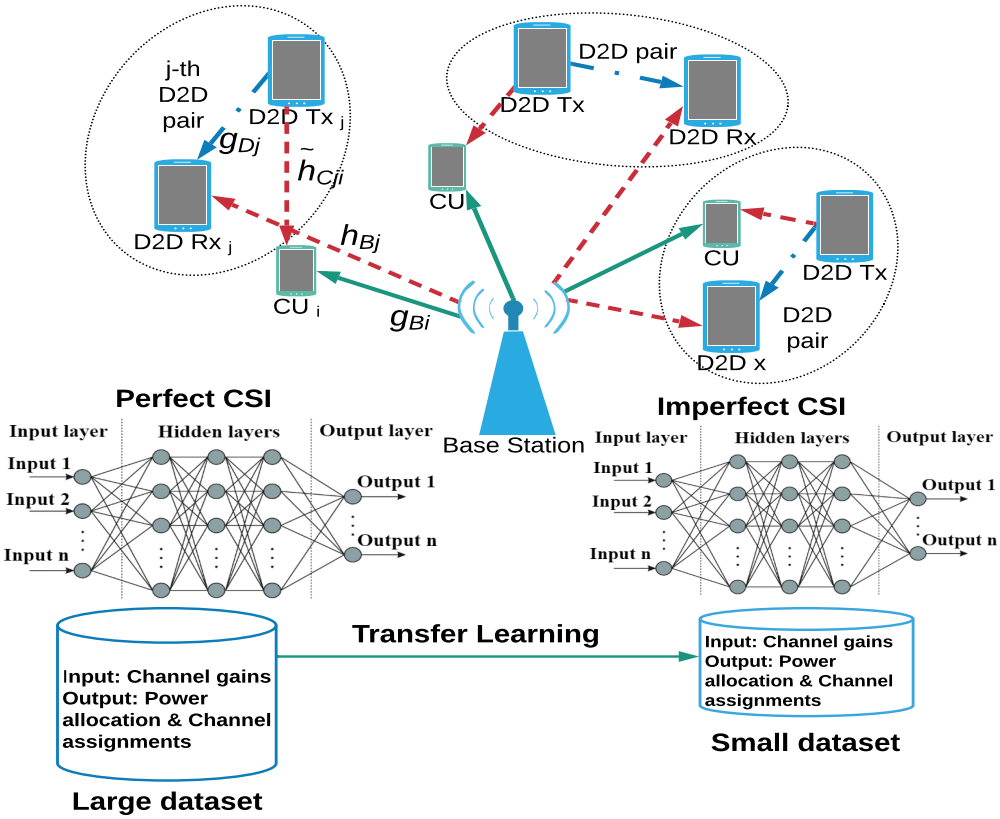


Figure E.1: Illustration of the system model with TL approach.

The remainder of this paper is structured as follows: Section E.2 describes the system model and problem formulation of resource allocation under perfect and imperfect CSI. Section E.3 presents the proposed deep learning and transfer learning-based approach for resource allocation. Section E.4 presents the experimental system setup and Section E.5 presents and discusses the results before the concluding remarks and future directions are provided in Section E.6.

E.2 System Model

In this work, we investigate the transfer of a DNN model trained for resource allocation (channel assignment and power allocation) to cellular users (CUs) and D2D pairs from the case of perfect CSI conditions to the case of practical operations under imperfect CSI conditions, as shown in Figure E.1. Here, we consider a cell enabled by a base station (BS), which communicates with N_C cellular users through N_C downlink channels. The cell is assumed to operate in fully loaded mode; thus, CUs can be indexed by $\mathcal{C} = \{1, ..., N_C\}$. Next, we consider N_D D2D pairs (indexed by $\mathcal{D} = \{1, ..., N_D\}$) wishing to communicate in underlay using the aforementioned N_C downlink channels. The notations for channels are as follows: g_{B_i} and g_{D_j} denote respective direct channel gains between, BS to i-th CU and transmitter and receiver in D2D pair; h_{B_j} and $h_{C_{j,i}}$ denotes respective interference channel gain between BS to the receiver of j-th D2D pair and the transmitter of j-th D2D pair to i-th CU. Further, we denote the total noise power in any channel by N_0.

Let $\beta_{i,j}$ be a binary variable denoting the channel assignment to the j-th D2D

pair; $\beta_{i,j} = 1$, if i-th CU shares channel with j-th D2D pair and $\beta_{i,j} = 0$ otherwise. The D2D pairs are allowed to simultaneously access multiple channels; thus, improving their individual sum rate. However, in order to limit interference among D2D pairs, sharing of a channel is restricted to at most one D2D pair, i.e., $\sum_{j=1}^{N_D} \beta_{i,j} \leq 1, \forall i$. Similarly, let P_{B_i} and $P_{D_{j,i}}$ denote respectively transmit power allocated to the BS over the i-th channel and the j-th D2D pair when accessing the i-th channel. The corresponding transmit powers are constrained as: $P_{B_i} \leq P_{B_{\max}}$ and $P_{D_{ji}} \leq P_{D_{\max}}$. Given the above system model, we define the following two resource allocation tasks:

E.2.1 Task I: Resource Allocation under Perfect CSI

In this task, we follow the same formulation as in [70]. They assume that all channel gains g_{B_i}, g_{D_j}, h_{B_j} and $h_{C_{j,i}}$; $1 \leq i \leq N_C$, $1 \leq j \leq N_D$ are perfectly known at the BS. The objective of this task is the sum rate maximization of both the cellular and the D2D network along with a fairness measure in channel assignment to the D2D pairs. Under the assumption of capacity-achieving codes, let $\Gamma(z) := \log_2(1+z)$; the sum rate over i-th channel is defined as: $R_i := \sum_{j\in\mathcal{D}} \beta_{i,j}[R_{C_{i,j}} + R_{D_{j,i}}] + (1 - \sum_{j\in\mathcal{D}} \beta_{i,j}) R_{C_{i,0}}$, where $R_{C_{i,j}} = \Gamma(P_{B_i} g_{B_i}/(N_0 + P_{D_{ji}} h_{C_{j,i}}))$ denotes the rate of the i-th CU when sharing the channel with the j-th D2D pair ($\beta_{ij} = 1$); $R_{D_{j,i}} = \Gamma(P_{D_{ji}} g_{D_j}/(N_0 + P_{B_i} h_{B_j}))$ the rate of the j-th D2D pair when sharing the channel with the i-th CU ($\beta_{ij} = 1$); and $R_{C_{i,0}} = \Gamma(P_{B_{max}} g_{B_i}/N_0)$ the rate of the i-th CU when it shares its channel with no D2D pair ($\beta_{ij} = 0\ \forall j$). The overall network rate of both cellular and D2D networks can be expressed as $R := \sum_{i\in\mathcal{C}} R_i$. For consideration of fairness in channel assignment to D2D pairs, they define unfairness measure (from [71]): $\delta^2(B) := 1/(N_D x_0^2) \sum_{j=1}^{N_D} (x_j(B) - x_0)^2$, where $x_j := \sum_{i=1}^{N_C} \beta_{i,j}$ is the number of channels assigned to the j-th D2D pair, $x_0 := N_C/N_D$ and B is channel assignemnet. Here, if N_C is an integer multiple of N_D, then $x_j = x_0\ \forall j$ is the fairest channel assignment possible. Finally, the overall resource allocation optimization problem under perfect CSI can be formulated as:

$$\max_{B,P_B,P_D} \quad R(B, P_B, P_D) - \gamma\delta^2(B) \tag{E.1a}$$

$$\text{subject to} \quad \beta_{i,j} \in \{0,1\}, \ \sum_{j=1}^{N_D} \beta_{i,j} \leq 1\, \forall i \tag{E.1b}$$

$$0 \leq P_{B_i} \leq P_{B_{max}}\, \forall i \quad 0 \leq P_{D_{ji}} \leq P_{D_{max}}\, \forall j, i \tag{E.2a}$$

$$\forall i, j, \ \frac{P_{B_i} g_{Bi}}{N_0 + P_{D_{ji}} h_{C_{j,i}}} \geq \eta^{C_{min}} \text{ if } \beta_{ij} = 1 \tag{E.2b}$$

$$\forall i, j, \ \frac{P_{D_{ji}} g_{Dj}}{N_0 + P_{B_i} h_{B_j}} \geq \eta^{D_{min}} \text{ if } \beta_{ij} = 1. \tag{E.2c}$$

where $\eta^{C_{min}}$ and $\eta^{D_{min}}$ are the respective minimum signal-to-interference plus noise ratio (SINR) requirements for CUs and D2D pairs. Notice that problem (E.1) is a mixed-integer non-convex program, which involves combinatorial complexity for

obtaining the optimal solution. A close-to-optimal solution to problem (E.1) is provided in [70], where joint power allocation and channel assignment are optimally decoupled in several power allocation problems and a channel assignment sub-problem. The decoupled power allocation subproblems have closed-form solutions and the channel assignment subproblem is solved by integer relaxation. This solution is computationally efficient. Thus, due to low complexity, one can easily obtain a large dataset to transfer a DNN model.

E.2.2 Task II: Resource Allocation under Imperfect CSI

In this task, we follow the same formulation as in [72]. The interference channel gain from the transmitter of the j-th D2D pair to the i-th CU, i.e., $h_{C_{j,i}}$, is considered to be estimated with minimum cooperation between the cellular and D2D networks. Thus, the interference channel gain is assumed to be exponentially distributed (Rayleigh fading) and is denoted by $\tilde{h}_{C_{j,i}}$; $1 \leq i \leq N_C$, $1 \leq j \leq N_D$. Due to imperfect CSI, the resource allocation optimization problem presented in (E.1) incurs the following modifications: (i) the objective function, and (ii) the stochastic minimum SINR constraint for the CUs.

The stochastic SINR constraint (E.2b) can be replaced with probabilistic constraint to guarantee a minimum outage probability ϵ, expressed as:

$$\Pr\left\{\frac{P_{B_i} g_{B_i}}{N_0 + P_{D_{ji}} \tilde{h}_{C_{j,i}}} \geq \eta^C_{min}\right\} \geq (1-\epsilon) \;\text{ if } \beta_{ij} = 1,\; \forall i, j \tag{E.3}$$

The probabilistic SINR constraint can be expressed in a closed form expression for a given statistical distribution of $h_{C_{j,i}}$. Thus, constraint (E.3) can be equivalently expressed as:

$$\frac{P_{B_i} g_{B_i}}{N_0 + P_{D_{ji}} F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon)} \geq \eta^C_{min} \tag{E.4}$$

where, $F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon)$ is the inverse cumulative distribution function (CDF) for $\tilde{h}_{C_{j,i}}$ evaluated at $(1-\epsilon)$.

Next, focusing on the stochastic objective function, the objective function (E.1a) can be replaced by the criterion to maximize the minimum network rate exceeded for $(1-\epsilon)$ portion of the time. The minimum network rate can be considered by analyzing the lower bound of the total rate at channel i, which is defined as $R^{LB}_i := (1 - \sum_{j\in\mathcal{D}} \beta_{i,j}) R_{C_{i,0}} + \sum_{j\in\mathcal{D}} \beta_{i,j}[R_{D_{j,i}} + R^{LB}_{C_{i,j}}]$, where, $R^{\text{LB}}_{C_{i,j}}$ denotes the lower bound (which must be at least achieved $(1-\epsilon)$ portion of the time) of the rate of the i-th CU when sharing the channel with the j-th D2D pair ($\beta_{ij} = 1$). Since $\tilde{h}_{C_{j,i}}$ is random, we can compute $R^{LB}_{C_{i,j}} = \Gamma(z^{LB}_{C_{i,j}})$ where $z^{LB}_{C_{i,j}} : \Pr\{z^{LB}_{C_{i,j}} \leq P_{B_i} g_{B_i}/(N_0 + P_{D_{ji}} \tilde{h}_{C_{j,i}})\} = 1-\epsilon$. The minimum sum rate is therefore $R(B, P_B, P_D) := \sum_{i\in\mathcal{C}} R^{LB}_i$. The fairness part is the same.

Once again, note that the resource allocation problem (E.1) with minimum network rate objective and minimum SINR constraint for CUs expressed in (E.4), is

a non-convex mixed-integer program and obtaining the optimal solution requires combinatorial complexity. An efficient close-to-optimal solution to the resource allocation problem for the case of imperfect CSI is provided in [72], where once again the joint power allocation and channel assignment problem is decoupled, without loss of optimality, in several power allocation and channel assignment subproblems. The decoupled power allocation subproblems are solved using iterative fractional programming, whose solutions are later used to obtain channel assignment by integer relaxation. Notice that due to the iterative fractional programming in solving the power allocation sub-problem, generating large datasets to train a DNN is cumbersome work due to larger computational complexity. Thus, in the next section, we explore the similarity between these two tasks: (i) resource allocation in the perfect CSI case, and (ii) resource allocation in the imperfect CSI case to exploit transfer learning.

E.3 Transfer Learning based Resource Allocation

The fundamental requirement in transfer learning is to establish a similarity between the tasks over which TL is performed. For the presented resource allocation problem, the similarity between two tasks: (i) resource allocation in perfect CSI and (ii) resource allocation in imperfect CSI can be established by following Lemma.

Lemma E.3.1. *Under the assumption that the interference channel gains $\tilde{h}_{C_{j,i}}$ follows an exponential distribution with the mean denoting the true channel gain value in the perfect CSI case, the two resource allocation tasks for perfect CSI and imperfect CSI coincide for outage probability $\epsilon = \frac{1}{e}$.*

Proof: Notice that the minimum network rate objective function with the modified constraint (E.4) for resource allocation in the imperfect CSI case, is the same as for the perfect CSI case when $F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon) = h_{C_{j,i}}$. For exponential distribution, $F^{-1}_{\tilde{h}_{C_{j,i}}}(1-\epsilon) = \mathbb{E}[\tilde{h}_{C_{j,i}}] \ln\left(\frac{1}{\epsilon}\right)$. Since the true CSI is equal to the mean of $\tilde{h}_{C_{j,i}}$, i.e., $h_{C_{j,i}} = \mathbb{E}[\tilde{h}_{C_{j,i}}]$, the two tasks are the same for $\epsilon = \frac{1}{e} = 0.3679$.
Next, we briefly discuss the DNN architecture, which is used in this work to obtain the resource allocation solution for the perfect CSI case.

E.3.1 Fully-connected DNN Architecture

A deep neural network (DNN) architecture consists of multiple numbers of layers between the input and output, each of which consists of a linear operation followed by a point-wise non-linearity, also known as the activation function.

Consider a feed-forward DNN with L layers, labelled $l = 1, ..., L$ and each with a corresponding dimension q_l, as shown in Figure E.2. The layer l is defined by the linear operation $\mathbf{W_l} \in \mathbb{R}^{q_l - 1 \times q_l}$ followed by a non-linear activation function $\sigma_l : R^{q_l} \rightarrow R^{q_l}$. Layer l receives input from the $l-1$ layer denoted as, $\mathbf{W_{l-1}} \in R^{q_{l-1}}$,

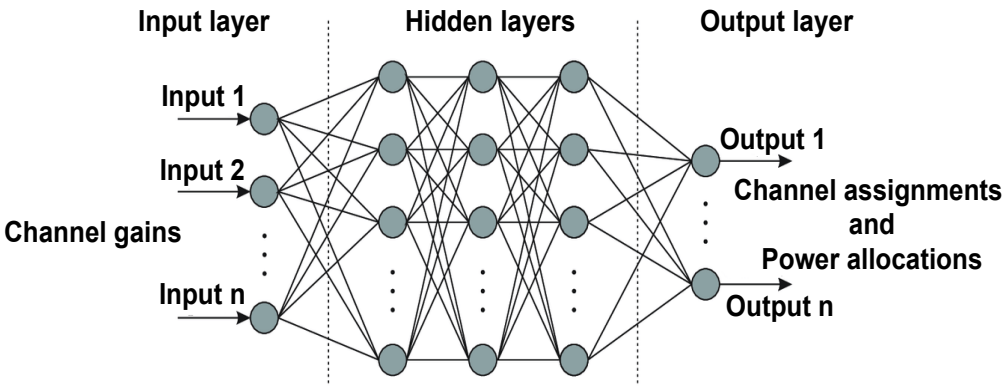


Figure E.2: Architecture of fully-connected DNN with input and output.

the resulting output of the layer l, $W_l \in R^{q_l}$, is then computed as $\mathbf{W_l} := \sigma_l(W_l \mathbf{W_{l-1}})$, where $\sigma_l(\cdot)$ is point-wise activation function. The final output of the DNN, W_L, is then related to the input W_0 by propagating through the various layers of the DNN as $W_L = \sigma_L(W_L(\sigma_{L-1}(W_{L-1}(....(\sigma_1(W_1 W_0))))))$. The DNN learns the layer-wise weights $W_1, W_2, ..., W_L$. Our input to DNN is channel gains and output is channel assignments and power allocations. The activation function σ_l includes a rectifier function (commonly referred to as ReLU), defined as $\sigma_l(x) = 0$ for $x < 0$ and x for $x > 0$.

E.3.2 Output Discretization and Scaling

In order to satisfy constraints (E.1b) and (E.2a), the output vector in the resource allocation problem, that is, the channel assignments and power allocations, are discretized and scaled respectively. A binary channel assignment to j-th D2D pair is assigned as 1 if the channel is assigned and 0 if the channel is not assigned. Since each channel can be assigned to at most one of the N_D D2D pairs, for a j-th D2D pair, we discretize maximum among N_C possible channel assignments values $\{\beta_{i,j}\}_{i \in \mathcal{C}}$ to one and other assignments to zero. Similarly, the power value should range between 0 and $P_{\max}$. However, the obtained output power values may be higher than $P_{\max}$ and lesser than 0. Thus, in order to bring it to the specified range, the power value is thresholded with 1 such that if the power is more than 1, then it will be mapped to 1 and if it is less than 0, then it is mapped to 0, followed by multiplication by $P_{\max}$.

E.3.3 Transfer Learning Strategy

In transfer learning, a DNN model trained for some specified task is transferred to perform a similar task decreasing substantially the retraining [91]. A given target task can be trained with fewer data by taking the trained model of a similar task as the baseline model. Here the trained baseline DNN model refers to the model trained for resource allocation in the perfect CSI scenario and the target task is resource allocation for the imperfect CSI scenario.

Thus, for this work, we train a baseline model for the perfect CSI case using a large dataset generated by the computationally efficient algorithm presented in [70]. Next, we use this baseline model to initialise the weights of the DNN model and fine-tune it for the imperfect CSI scenario, where we assume that we have an insufficient amount of dataset (generated from [72]) for training and testing, due to the huge complexity of the considered algorithm.

E.4 System Setup

The proposed DNN and TL-based approaches are implemented in Python 3.7.3 with TensorFlow 2.2.0 on a Windows 10 laptop having an Intel Core i5 8th generation processor, Intel UHD Graphics 620, and 16 GB of memory. The original solver for the perfect and the imperfect CSI model of resource allocation is implemented in MATLAB R2018a. Therefore, we have compared the computational performance of the perfect and imperfect CSI model with the proposed DNN and TL-based approach.

E.4.1 Data Generation

The simulation setup to generate data comprises a circular cell of a 500m radius in which the CUs and D2D transmitters are placed uniformly at random. Each D2D receiver is placed uniformly at random inside a circle of radius 5m centred at the corresponding transmitter. The channel gains are calculated using a path-loss model with exponent 2 and gain -5 dB at a reference distance of 1m. We assume $\tilde{h}_C$ to be exponentially distributed with the mean value obtained from the mentioned path-loss model. Averages over 100,000 independent realizations of the user locations with parameters $BW = 15$ kHz, $\gamma = 50 \times BW$, $N_D = 5$, $N_C = 5$, $N_0 = -70$dBW (γ is scaled with BW to ensure that the unfairness and the achieved rate are of comparable values) are performed. Thus, the input to the DNN is the set of channel gains, that accounts for 40 inputs. The output to the DNN is a joint set of 50 power allocations and 25 channel assignment variables. For training the baseline model with perfect CSI, we consider 100,000 input-output pair samples.

E.4.2 Parameter Selection

To obtain a satisfactory baseline model, we train DNNs with a different number of hidden layers, each having a variable number of neurons. Since output values (channel assignments and power allocations) are positive, the activation function to the hidden layers is a rectified linear unit (ReLU) and the output layer is linear. The weights of the baseline DNN are initialized randomly. We use mean square error (MSE) as the loss function and ADAM optimizer with a learning rate of 0.001 for stochastic optimization. We standardize the dataset by taking the mean and scaling to unit variance. We use a mini-batch of 256 samples. Training epochs are set empirically.

E.4.3 Training and Testing Stage

For the baseline DNN training, we divide the whole data into 80:20 ratio, that is, 80 % for training and 20 % for testing. During testing, for each channel realization, we pass it through the trained network and collect the optimized power allocation and channel assignments. Then, we also evaluate the satisfaction of power constraints (E.2b), SINR (E.2c), and (E.4).

E.5 Results and Discussions

This section presents numerical results and a discussion to showcase the effectiveness of the proposed DNN and TL-based approach. The performance accuracy of the model is evaluated in terms of MSE. There are 100,000 input-output samples in both scenarios.

Table E.1 presents the different training and testing errors that are carried out for the perfect CSI scenario to train the DNN model and set it as our baseline model for TL. It can be observed that the DNN model having 3 hidden layers (highlighted) is performing better as compared to the other combinations. The accuracy, i.e., test MSE, are comparable to the training part, thus illustrating no over-fitting in the model. Moreover, the percentage of constraint satisfaction (E.2b) and (E.2c) in the testing stage are high and nearly the same as its counterpart in training.

Table E.2 shows the results for the imperfect CSI scenario with outage probability $\epsilon = 0.1$ when (i) trained via a DNN model with the same training-testing split and configurations as in the perfect CSI scenario, and (ii) trained with a baseline model learnt from perfect CSI (TL) with 50% training-testing split. It can be noticed that the TL-based approach has obtained nearly the same MSE as the direct DNN-based approach with the random initialization, and has a similar percentage of constraint satisfaction. This indicates that our baseline model obtained/transferred from the

Table E.1: Model learning for Perfect CSI

Number of hidden layers	**3**	6	3	6
Input layer neurons	**40**	40	40	40
Hidden layer activation function	**ReLU**	ReLU	ReLU	ReLU
Hidden layer neurons	**40,20,5**	40,20,5, 20,40,60	40,60,80	40,60,80, 100,80,60
Output layer activation function	**Linear**	Linear	Linear	Linear
Output layer neurons	**75**	75	75	75
Test MSE	**0.038**	0.039	0.205	0.090
Train MSE	**0.039**	0.039	0.206	0.090
Test E.2b; E.2c constraints satisfied (in %)	**94.53; 100**	94.69; 100	81.78; 86.70	84.67; 92.96
Train E.2b; E.2c constraints satisfied (in %)	**94.55; 100**	94.58; 99.99	81.78; 86.62	84.53; 92.93

Table E.2: Model learning for Imperfect CSI

	Test MSE	Train MSE	Test E.2b constraint satisfied	Train E.2b constraint satisfied	Test E.4 constraint satisfied	Train E.4 constraint satisfied
DNN	0.017	0.017	95.28	95.32	100.0	100.0
TL	0.017	0.017	95.01	94.92	100.0	100.0

perfect CSI scenario improves the learning for the imperfect CSI case with less amount of training data (usually termed as sample-complexity).

The amount of training data required to retrain the baseline model depends on the degree of task mismatch. It can be observed from Figure E.3 that when two tasks are similar, i.e., outage probability $\epsilon = 0.3679$, then just with 20% of training data, the MSE of TL is 0.04. Moreover, with 30% of training data, the transferred baseline model is completely trained. However, as illustrated in Figure E.4, when a lot of mismatch presents between the two tasks, $\epsilon = 0.05$, then almost 80% of training data is required for retraining the transferred baseline model. Since the problem is highly non-convex, the solution that is achieved may correspond to local minima. In Figure E.4, the training of the DNN seems to converge to a local optima when the amount of training data is below 70%, but then it achieves a better solution with 80% of the training data.

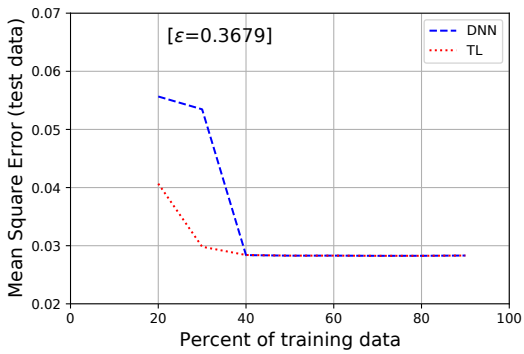


Figure E.3: Percent of training data vs MSE (No. of epochs=10).

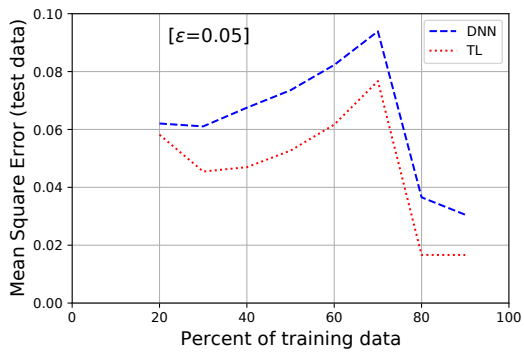


Figure E.4: Percent of training data vs MSE (No. of epochs=10).

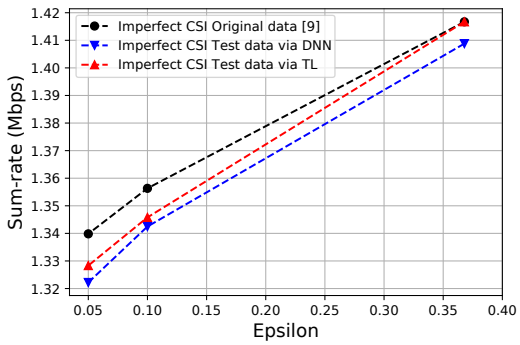


Figure E.5: Sum-rate for different values of outage probability ϵ.

Figure E.5 shows the plot of sum-rate obtained for different values of outage probability ϵ, that is, 0.05, 0.1 and 0.3679, for the case of imperfect CSI [72]. The black dashed line is the sum-rate obtained by averaging the 100,000 realizations of the original solutions calculated numerically using optimization method [72]. The blue and red dashed lines, respectively, are the sum-rates obtained by training the imperfect CSI model via DNN and TL with 50% training-testing separation of samples. It can be noticed that the sum-rate of TL is higher than the one obtained for the DNN, showing a better performance. Specifically, the TL is achieving the same sum-rate as the original for $\epsilon = 0.3679$, a condition in which the two resource allocation tasks for perfect CSI and imperfect CSI (under outage probability $\epsilon = 0.3679$) are similar. This signifies that even with a small number of training samples, TL achieves nearly the same performance as the original, and superior performance, as compared to using a DNN without TL, hence resulting in saving computational resources.

Table E.3 presents the comparison of computational performance of the proposed TL-based approach with the DNN-based approach (both implemented in Python) and with the numerical optimization methods proposed in [70, 72] (implemented in MATLAB). It can be noticed that the MATLAB implementation of both algorithms requires substantially more computational time as compared to the TL-based approach. Moreover, the case of imperfect CSI requires a lesser number of epochs to

Table E.3: Computational performance for perfect and Imperfect CSI

	Optimization methods [70, 72]	DNN		TL	
Scenario	Time (second)	Train epochs (80,000 samples)	Test time (second)	Train epochs (50,000 samples)	Test time (second)
Perfect CSI	46.0	51	0.166	-	-
Imperfect CSI	214.0	13	0.161	19	0.125

train the DNN as compared to the perfect CSI for the same training data. The TL-based approach requires a lesser number of training iterations, that is, 950,000 (19 epochs $\times$ $50,000$ samples) as compared to the imperfect CSI, which is 1040,000 (13 epochs $\times$ $80,000$ samples). As expected, the TL-based approach requires less training data (sample complexity), thus saving computational resources.

E.6 Conclusion and Future Work

In this paper, we have first designed a DNN-based algorithm for wireless resource allocation in the case of a perfect CSI model. The optimal DNN model is considered as the baseline for transfer learning in order to train an imperfect CSI model. Our results show that for the joint problem of channel assignments and power allocations over D2D communications, DNNs can approximate accurately this solver between cellular users and D2D pairs, with the knowledge of the channel-state-information (CSI). Moreover, the transferred model to the imperfect CSI scenario performs better than the DNN model without transfer learning and requires less amount of training data, reducing the sample complexity. Our results show that DNN has great potential to solve real-time wireless resource allocation problems and transfer learning can reduce the data-hungry nature of DNN, saving computational resources. TL can lead to a good performance across similar problems with a limited amount of training data.

Bibliography

[1] Mohamed Khalaf-Allah. Time of arrival (TOA)-based direct location method. In *16th International Radar Symposium (IRS)*, pages 812–815. IEEE, 2015.

[2] Rahul Kumar Jaiswal, Mohamed Elnourani, Siddharth Deshmukh, and Baltasar Beferull-Lozano. A Data-driven Transfer Learning Method for Radio Map Estimation. TechRxiv, 2023.

[3] Ron Levie, Çağkan Yapar, Gitta Kutyniok, and Giuseppe Caire. RadioUNet: Fast radio map estimation with convolutional neural networks. *IEEE Transactions on Wireless Communications*, 20(6):4001–4015, 2021.

[4] Jaana Laiho, Achim Wacker, and Tomáš Novosad. *Radio Network Planning and Optimisation for UMTS*, volume 2. Wiley Online Library, 2002.

[5] Daniel Romero, Seung-Jun Kim, Georgios B Giannakis, and Roberto López-Valcarce. Learning Power Spectrum Maps from Quantized Power Measurements. *IEEE Transactions on Signal Processing*, 65(10):2547–2560, 2017.

[6] Yves Teganya, Daniel Romero, Luis Miguel Lopez Ramos, and Baltasar Beferull-Lozano. Location-free Spectrum Cartography. *IEEE Transactions on Signal Processing*, 67(15):4013–4026, 2019.

[7] Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

[8] Rahul Jaiswal, Mohamed Elnourani, Siddharth Deshmukh, and Baltasar Beferull-Lozano. Location-free Indoor Radio Map Estimation using Transfer learning. In *97th Vehicular Technology Conference*, pages 1–7. IEEE, 2023.

[9] Suzhi Bi, Jiangbin Lyu, Zhi Ding, and Rui Zhang. Engineering radio maps for wireless resource management. *IEEE Wireless Communications*, 26(2):133–141, 2019.

[10] Minkyu Lee and Dongsoo Han. Voronoi Tessellation Based Interpolation Method for Wi-Fi Radio Map Construction. *IEEE Communications Letters*, 16(3):404–407, 2012.

[11] Yves Teganya, Luis M Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano. Localization-free power cartography. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3549–3553, 2018.

[12] John Krumm and John Platt. Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System. *Microsoft Research*, pages 1–9, 2003.

[13] Din-Hwa Huang, Sau-Hsuan Wu, Wen-Rong Wu, and Peng-Hua Wang. Cooperative Radio Source Positioning and Power Map Reconstruction: A Sparse Bayesian Learning Approach. *IEEE Transactions on Vehicular Technology*, 64(6):2318–2332, 2014.

[14] Olaf Ronneberger and Philipp Fischer. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241. Springer, 2015.

[15] Yves Teganya and Daniel Romero. Deep completion autoencoders for radio map estimation. *IEEE Transactions on Wireless Communications*, 21(3):1710–1724, 2021.

[16] Songyang Zhang, Achintha Wijesinghe, and Zhi Ding. RME-GAN: A Learning Framework for Radio Map Estimation based on Conditional Generative Adversarial Network. *IEEE Internet of Things Journal*, 2023.

[17] Subash Timilsina, Sagar Shrestha, and Xiao Fu. Quantized radio map estimation using tensor and deep generative models. *arXiv preprint arXiv:2303.01770*, 2023.

[18] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A Survey of Transfer Learning. *Journal of Big data*, 3(1):1–40, 2016.

[19] F. Zhuang, Z. Qi, K. Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[20] Chuanting Zhang and Haixia Zhang. Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. *IEEE Journal on Selected Areas in Communications*, 37(6):1389–1401, 2019.

[21] Q. Peng, A. Gilman, N. Vasconcelos, P. C Cosman, and L. B Milstein. Robust Deep Sensing through Transfer Learning in Cognitive Radio. *IEEE Wireless Communications Letters*, 9(1):38–41, 2019.

[22] Yuwen Yang and Feifei Gao. Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems. *IEEE Transactions on Communications*, 68(12):7485–7497, 2020.

[23] Siqi Bai and Yongjie Luo. Transfer learning for wireless fingerprinting localization based on optimal transport. *Sensors*, 20(23):6994, 2020.

[24] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

[25] Claudia Parera, Qi Liao, Cristian Tatino, Alessandro EC Redondi, and Matteo Cesana. Transfer learning for tilt-dependent radio map prediction. *IEEE Transactions on Cognitive Communication and Networking*, 6(2):829–843, 2020.

[26] Xu Han, Lei Xue, Ying Xu, and Zunyang Liu. A two-phase transfer learning-based power spectrum maps reconstruction algorithm for underlay cognitive radio networks. *IEEE Access*, 8:81232–81245, 2020.

[27] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G Yen. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 22(4):501–514, 2017.

[28] Joshua Kwesi Desbordes, Kai Zhang, Xiaoming Xue, Xiaopeng Ma, Qin Luo, Zhaoqin Huang, Sun Hai, and Yao Jun. Dynamic production optimization based on transfer learning algorithms. *Journal of Petroleum Science and Engineering*, 208:109278, 2022.

[29] Lei Chen, Hai-Lin Liu, Kay Chen Tan, and Ke Li. Transfer learning-based parallel evolutionary algorithm framework for bilevel optimization. *IEEE Transactions on Evolutionary Computation*, 26(1):115–129, 2021.

[30] Mohammad Mahdi Behzadi and Horea T Ilieş. Real-time topology optimization in 3D via deep transfer learning. *Computer-Aided Design*, 135:103014, 2021.

[31] Yifei Shen, Yuanming Shi, Jun Zhang, and Khaled B Letaief. LORM: Learning to optimize for resource management in wireless networks with few training samples. *IEEE Transactions on Wireless Communications*, 19(1):665–679, 2019.

[32] Yifei Shen, Yuanming Shi, Jun Zhang, and Khaled B Letaief. Transfer Learning for Mixed-integer Resource Allocation Problems in Wireless Networks. In *IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[33] Remcom: The Electromagnetic Simulation Software. https://www.remcom.com/wireless-insite-em-propagation-software.

[34] Rahul Jaiswal, Mohamed Elnourani, Siddharth Deshmukh, and Baltasar Beferull-Lozano. Deep Transfer Learning Based Radio Map Estimation for Indoor Wireless Communications. In *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2022.

[35] Rahul Kumar Jaiswal, Mohamed Elnourani, Siddharth Deshmukh, and Baltasar Beferull-Lozano. Leveraging Transfer Learning for Radio Map Estimation via Mixture of Experts. TechRxiv, 2023.

[36] Rahul Jaiswal, Siddharth Deshmukh, Mohamed Elnourani, and Baltasar Beferull-Lozano. Transfer Learning Based Joint Resource Allocation for Underlay D2D Communications. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1479–1484, 2022.

[37] Alessio Zappone, Marco Di Renzo, and Mérouane Debbah. Wireless networks design in the era of deep learning: Model-based, AI-based, or both? *IEEE Transactions on Communications*, 67(10):7331–7376, 2019.

[38] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[39] Richard Roy and Thomas Kailath. ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989.

[40] Yingbo Hua and Tapan K Sarkar. Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(5):814–824, 1990.

[41] Chang-Fa Yang, Boau-Cheng Wu, and Chuen-Jyi Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *IEEE Transactions on Antennas and Propagation*, 46(6):907–919, 1998.

[42] Amir Beck, Petre Stoica, and Jian Li. Exact and approximate solutions of source localization problems. *IEEE Transactions on Signal Processing*, 56(5):1770–1778, 2008.

[43] Alireza Zaeemzadeh, Mohsen Joneidi, Behzad Shahrasbi, and Nazanin Rahnavard. Robust target localization based on squared range iterative reweighted least squares. In *14th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 380–388. IEEE, 2017.

[44] Darya Ismailova and Wu-Sheng Lu. Improved Least-Squares Methods for Source Localization: An Iterative Re-weighting Approach. In *IEEE International Conference on Digital Signal Processing*, pages 665–669. IEEE, 2015.

[45] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[46] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[47] Enas Elgeldawi, Awny Sayed, Ahmed R Galal, and Alaa M Zaki. Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. In *Informatics*, volume 8, page 79. MDPI, 2021.

[48] Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly Media, 2017.

[49] Mohssen Mohammed and Eihab Bashier Mohammed Bashier. *Machine Learning: Algorithms and Applications*. CRC Press, 2016.

[50] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021.

[51] Marco De Angelis and Ander Gray. Why the 1-Wasserstein distance is the area between two marginal CDFs. *arXiv preprint arXiv:2111.03570*, 2021.

[52] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[53] Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2009.

[54] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *12th European Conference on Computer Vision*, pages 214–227. Springer, 2012.

[55] Ruan Lakemond, Clinton Fookes, and Sridha Sridharan. Affine adaptation of local image features using the Hessian matrix. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 496–501, 2009.

[56] David G Lowe. Distinctive Image Features from Scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[57] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[58] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[59] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer, 2010.

[60] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. BRISK: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision*, pages 2548–2555. IEEE, 2011.

[61] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *European Conference on Computer Vision*, pages 183–196. Springer, 2010.

[62] Wang Yuanji and Jiang Qinzhong. Image quality evaluation based on image weighted separating block peak signal to noise ratio. In *IEEE International Conference on Neural Networks and Signal Processing*, volume 2, pages 994–997, 2003.

[63] D Poobathy and R Manicka Chezian. Edge detection operators: Peak signal to noise ratio based comparison. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, 10:55–61, 2014.

[64] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[65] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[66] Saeed Masoudnia and Reza Ebrahimpour. Mixture of Experts: A Literature Survey. *The Artificial Intelligence Review*, 42(2):275, 2014.

[67] Robert A Jacobs, Michael I Jordan, and Geoffrey E Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991.

[68] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards Understanding Mixture of Experts in Deep Learning. *arXiv:2208.02813*, 2022.

[69] Wen Zhu and Nancy Zeng. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG Proceedings: Health Care and Life Sciences*, 19:67–75, 2010.

[70] Mohamed Elnourani, Mohamed Hamid, Daniel Romero, and Baltasar Beferull-Lozano. Underlay Device-to-device Communications on Multiple Channels. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3684–3688, 2018.

[71] Chen Xu, Lingyang Song, Zhu Han, Qun Zhao, Xiaoli Wang, and Bingli Jiao. Interference-aware Resource Allocation for Device-to-device Communications as an Underlay using Sequential Second Price Auction. In *IEEE International Conference on Communications (ICC)*, pages 445–449, 2012.

[72] Mohamed Elnourani, Baltasar Beferull-Lozano, Daniel Romero, and Siddharth Deshmukh. Reliable Underlay Device-to-Device Communications on Multiple Channels. In *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2019.

[73] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[74] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the Trade*, pages 55–69. Springer, 2002.

[75] Aishwarya Asesh. Normalization and Bias in Time Series Data. In *9th Machine Intelligence and Digital Interaction Conference*, pages 88–97. Springer, 2022.

[76] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations (ICLR)*, 2018.

[77] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*, 2015.

[78] Mark de Rooij and Wouter Weeda. Cross-validation: A method every psychologist should know. *Advances in Methods and Practices in Psychological Science*, 3(2):248–263, 2020.

[79] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of ML Research*, 15(1):1929–1958, 2014.

[80] Luis M Lopez-Ramos, Yves Teganya, Baltasar Beferull-Lozano, and Seung-Jun Kim. Channel Gain Cartography via Mixture of Experts. In *IEEE Global Communications Conference*, pages 1–7, 2020.

[81] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[82] Theodore S Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[83] Alan Bundy and Lincoln Wallen. Difference of Gaussians. *Catalogue of Artificial Intelligence Tools*, pages 30–30, 1984.

[84] Rui Yin, Guanding Yu, Caijun Zhong, and Zhaoyang Zhang. Distributed Resource Allocation for D2D Communication Underlaying Cellular Networks. In *IEEE International Conference on Communications (ICC)*, pages 138–143, 2013.

[85] Arash Asadi, Qing Wang, and Vincenzo Mancuso. A Survey on Device-to-device Communication in Cellular Networks. *IEEE Communications Surveys and Tutorials*, 16(4):1801–1819, 2014.

[86] Daquan Feng, Lu Lu, Yi Yuan-Wu, Geoffrey Ye Li, Shaoqian Li, and Gang Feng. Device-to-device Communications in Cellular Networks. *IEEE Communications Magazine*, 52(4):49–55, 2014.

[87] H. Sun, X. Chen, Q. Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos. Learning to Optimize: Training Deep Neural Networks for Wireless Resource Management. In *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6, 2017.

[88] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wireless Communications Letters*, 7(1):114–117, 2017.

[89] Woongsup Lee, Minhoe Kim, and Dong-Ho Cho. Deep Learning Based Transmit Power Control in Underlaid Device-to-device Communication. *IEEE Systems Journal*, 13(3):2551–2554, 2018.

[90] Fei Liang, Cong Shen, Wei Yu, and Feng Wu. Power Control for Interference Management via Ensembling Deep Neural Networks. In *IEEE/CIC International Conference on Communications in China (ICCC)*, pages 237–242, 2019.

[91] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for Transfer Learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200, 2007.