



Machine Learning Tools for Radio Map Estimation in Fading-Impaired Channels

Yves Teganya

Yves Teganya

**Machine Learning Tools for Radio Map Estimation in
Fading-Impaired Channels**

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialisation in Information and
Communication Technology

University of Agder
Faculty of Engineering and Science
2020

Doctoral Dissertations at the University of Agder 297
ISSN: 1504-9272
ISBN: 978-82-7117-999-1

©Yves Teganya, 2020

Printed by 07 Media
Kristiansand

Preface and Acknowledgments

This dissertation is the result of the research work carried out at the WISENET Center, Department of Information and Communication Technology (ICT), University of Agder (UiA), Grimstad Campus, Norway, from January 2017 to June 2020.

I am very grateful to my main supervisor Associate Professor Daniel Romero and co-supervisor Professor Baltasar Enrique Beferull-Lozano for their excellent supervision, encouragements, precious comments, visionary ideas, and constant supports. Special thanks to Dr. Luis Miguel Lopez-Ramos, collaborating with him was an enriching opportunity. I am grateful to Associate Professor Linga Reddy Cenkeramaddi and postdoctoral research fellows at the WISENET Center Dr. Leila Ben Saad and Dr. Siddharth Deshmukh for their useful comments, supports, and ideas.

I would like to thank the head of the ICT department Folke Haugland for his support, the administrative staff Emma and Kristine for their help with almost everything. Moreover, I would like to express my gratitude to the WISENET Center Higher Executive Officer Julia for her continual support.

My sincere gratitude goes to my doctoral fellows Bakht, Elnourani, Henning, Siavash, Ravi, Emilio, Jyotirmoy, Rolf, Rohan, Juan, Rahul, Kevin, and Sarang for their cooperation and the friendly working environment.

I express my eternal gratitude to my parents, sisters, and brothers far away for their blessings, love, understanding, and support. I would also like to thank my daughters Nora and Noreen for their love and care while working on my thesis. Last but not least, I am grateful to my wife Yvonne Laura who always loved, supported, motivated, and encouraged me in this endeavor.

Many thanks to all of you for your kindness and encouragement.

Yves Teganya
June 2020
Grimstad, Norway

Abstract

In spectrum cartography, also known as radio map estimation, one constructs maps that provide the value of a given channel metric such as the received power, power spectral density (PSD), electromagnetic absorption, or channel-gain for every spatial location in the geographic area of interest. The main idea is to deploy sensors and measure the target channel metric at a set of locations and interpolate or extrapolate the measurements. Radio maps find a myriad of applications in wireless communications such as network planning, interference coordination, power control, spectrum management, resource allocation, handoff optimization, dynamic spectrum access, and cognitive radio. More recently, radio maps have been widely recognized as an enabling technology for unmanned aerial vehicle (UAV) communications because they allow autonomous UAVs to account for communication constraints when planning a mission. Additional use cases include radio tomography and source localization.

A number of approaches have been developed to construct power, PSD, and channel-gain maps, mainly relying on machine learning tools. Unfortunately, the ability of the existing schemes to obtain accurate map estimates is severely impaired by small- and large-scale fading. Specifically, one of the limitations of all existing schemes is that they require accurate knowledge of the sensor locations. However, location is seldom known in practice and, therefore, must be estimated from localization features such as the received signal strength (RSS), the time of arrival (ToA), the time difference of arrival (TDoA), or the direction of arrival (DoA), of positioning pilot signals transmitted by satellites (e.g. in GPS) or terrestrial base stations (e.g. in LTE or WiFi). Unfortunately, accurate location estimates are often not available in practice since fading corrupts those pilot signals, thereby hindering the applicability of existing map estimation techniques, especially in indoor and dense urban scenarios. A second limitation is that existing approaches build upon interpolation schemes unable to learn how radio frequency signals propagate and, therefore, their performance heavily degrades in strong fading channels.

To overcome the first limitation, this thesis introduces a novel paradigm termed *location-free* radio map estimation, where maps are constructed by relying on features of the positioning signals rather than on location estimates. Within this paradigm, specific learning algorithms are developed here and offer a markedly improved estimation performance as compared with existing approaches, which rely on localization, as demonstrated by simulation studies in an urban canyon and indoor scenarios.

In order to circumvent the second limitation, this thesis proposes a novel approach in which the spatial structure of propagation phenomena such as shadowing (large-scale fading) is learned beforehand from a data set of past measurements. Capitalizing on

the manifold structure of radio maps, a deep completion neural network with an encoder-decoder architecture is proposed to estimate PSD maps. Remarkably, this is the first work to estimate radio maps using deep neural networks. Learning the structure of propagation phenomena through a realistic data set is seen to yield estimators that require fewer measurements to attain a target performance relative to existing schemes. Conversely, when the same number of measurements are given, the proposed data-driven approach significantly outperforms state-of-the-art alternatives by a wide margin.

Thesis Details

Thesis Title : Machine Learning Tools for Radio Map Estimation
in Fading-Impaired Channels

PhD Student : Yves Teganya

Principal supervisor : Associate Professor Daniel Romero

Co-supervisor : Professor Baltasar Enrique Beferull-Lozano

Papers included in the dissertation

- Paper A** Y. Teganya, L. M. Lopez-Ramos, D. Romero, and B. Beferull-Lozano, “Localization-Free Power Cartography,” in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, Calgary, Canada, Apr. 2018, pp. 3549–3553.
- Paper B** Y. Teganya, D. Romero, L. M. Lopez-Ramos, and B. Beferull-Lozano, “Location-Free Spectrum Cartography,” *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4013–4026, Aug. 2019.
- Paper C** Y. Teganya and D. Romero, “Data-Driven Spectrum Cartography via Deep Completion Autoencoders,” in *Proc. IEEE Int. Conf. Commun.*, Dublin, Ireland, Jun. 2020, pp. 1–7.
- Paper D** Y. Teganya and D. Romero, “Deep Completion Autoencoders for Radio Map Estimation,” submitted for publication in the *IEEE Trans. Wireless Commun.* [Online]. Available: arXiv:2005.05964.

Other papers not included in the dissertation

- Paper 1** D. Romero, R. Shrestha, Y. Teganya, and S. P. Chepuri, “Aerial Spectrum Surveying: Radio Map Estimation with Autonomous UAVs,” presented at the *IEEE Int. Workshop Mach. Learn. Signal Process.*, Espoo, Finland, Sep. 2020. [Online]. Available: arXiv:2005.02432.
- Paper 2** L. M. Lopez-Ramos, Y. Teganya, B. Beferull-Lozano, and S.-J. Kim, “Channel-Gain Cartography via Mixture of Experts,” to be presented at the *IEEE Global Commun. Conf.*, Taipei, Taiwan, Dec. 2020.

Contents

Preface and Acknowledgments	v
Abstract	vii
Thesis Details	ix
Contents	xii
List of Figures	xvi
List of Tables	xvii
List of Abbreviations	xix
I Preliminaries	1
1 Introduction	3
1.1 Spectrum cartography	3
1.2 State-of-the-art	4
1.3 Motivation and contributions of the thesis	5
1.4 Structure of the thesis	7
2 Location-Free Radio Map Estimation	9
2.1 Problem formulation	9
2.2 Location-free approach	10
2.2.1 Formulation	10
2.2.2 Kernel-based power map learning	11
2.3 Location-free features	12
2.3.1 Feature extraction	12
2.3.1.1 Sensors are synchronized with base stations	12
2.3.1.2 Sensors are not synchronized with base stations	13
2.3.2 Radio map reconstruction from a reduced set of features	13
2.3.3 Handling unavailable features	14

3	Data-Driven Radio Map Estimation	17
3.1	PSD map estimation problem	17
3.2	Data-driven radio map estimation	18
3.2.1	Map estimation as a tensor completion task	18
3.2.2	Deep completion networks for radio map estimation	19
3.2.3	Exploiting structure in the frequency domain	20
3.2.3.1	Frequency separation	20
3.2.3.2	Output layers for parametric PSD expansions	20
3.2.4	Deep completion autoencoders	22
3.2.5	Training approaches	24
3.2.5.1	Synthetic training data	24
3.2.5.2	Real training data	24
3.2.5.3	Hybrid training	24
4	Evaluation	25
4.1	Location-free radio map estimation	25
4.1.1	Urban canyon	25
4.1.2	Indoor	27
4.1.2.1	LocF vs. LocB	27
4.1.2.2	Feature design	28
4.1.2.3	LocF map estimation with missing features	29
4.2	Data-driven radio map estimation	31
4.2.1	Power map estimation	31
4.2.1.1	Gudmundson data set	31
4.2.1.2	Wireless Insite data set	32
4.2.2	PSD map estimation	33
4.2.2.1	Gudmundson data set	34
4.2.2.2	Wireless Insite data set	35
5	Conclusions and Future Work	37
5.1	Summary of contributions	37
5.2	Limitations and future work	38
	References	44
II	Publications	45
A	Paper A	47
B	Paper B	59
C	Paper C	89
D	Paper D	107

List of Figures

1.1	Power map at 1400MHz	3
1.2	Estimation of spatial coordinates using time difference of arrival (TDoA): (a)-(b) in free space, (c)-(d) indoor where the solid black lines represent the walls of the building; the black dots represent the locations of the anchor base stations. The color of each pixel represents the value of the estimated location coordinate at each point in the 150×150 grid area. Because location estimates in (a)-(b) coincide with the true locations, they can act as colorbars to the estimates in (c)-(d).	6
2.1	Multi-lateration using ToA measurements with circles as possible sensor locations: (a) consistent ToA with the sought sensor location being the intersection of the circles (black square) and (b) inconsistent ToA measurements. The red stars represent the locations of the anchor base stations.	10
2.2	Singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of Φ in non-increasing order for a multipath environment with $L = 4$ transmitters.	14
3.1	Model setup and area discretization.	18
3.2	PSD $\Psi(\mathbf{x}, f)$ at location \mathbf{x} using a basis expansion model with Gaussian functions.	21
3.3	Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying the locations of the measurements, and (right) map estimate.	22
3.4	Autoencoder architecture.	23
4.1	(a) True map, (b) LocF, and (c) LocB estimated maps, $N = 160$	26
4.2	Performance comparison between the LocF and LocB radio map estimation curves.	26
4.3	(left) True map, (middle) LocB, and (right) LocF estimated maps; $N = 300$. The black crosses indicate the sensor locations and the solid white lines represent the walls of the building.	27
4.4	Maps of the $M = 10$ LocF features with $L = 5$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations.	28
4.5	Performance comparison between the LocF and LocB radio maps estimation curves.	29

4.6	Estimated map NMSE with reduced features for different r and without reduced features; $L = 5$	30
4.7	(top) Average number of missing features and (bottom) estimated map NMSE, both as a function of Γ with $L = 5$	30
4.8	Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the measurements; (center right) and (right): map estimates. White areas in the true and reconstructed maps represent buildings.	32
4.9	Comparison with state-of-the-art alternatives. Training and testing maps drawn from the Gudmundson data set.	32
4.10	Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1]. Training and testing maps drawn from the Gudmundson data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$	32
4.11	Comparison with state-of-the-art alternatives. The training and testing maps were obtained from the Wireless InSite data set.	33
4.12	Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1] where the training and testing maps were obtained from the Wireless InSite data set, $Q_t = 10$	33
4.13	Maps of the true and estimated coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^B$ over \mathcal{X} , $B = 4$	34
4.14	PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ with a signal basis formed by using raised-cosine functions.	34
4.15	Performance comparison of the proposed scheme with that of the algorithm in [2].	35
4.16	Map estimate RMSE of the proposed scheme for PSD cartography. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 5$	35
A.1	(a) True map, (b) localization-free, and (c) localization-based estimated maps ($\lambda = 3 \times 10^{-3}$, $N = 160$).	54
A.2	(a) Estimated map NMSE for different values of number of features, M and sensors, N ; and (b) Performance comparison between the localization-free cartography and the localization-based cartography ($\lambda = 3 \times 10^{-3}$, $\sigma = 30$ m).	56
B.1	Estimation of spatial coordinates using TDoA: (a)-(b) in free space, (c)-(d) indoor where the solid black lines represent the walls of the building; the black dots represent the locations of the anchor base stations. The color of each pixel represents the value of the estimated location coordinate at each point in the 150×150 grid area. Because location estimates in (a)-(b) coincide with the true locations, they can act as colorbars to the estimates in (c)-(d).	64
B.2	Multi-lateration using ToA measurements with circles as possible sensor locations: (a) consistent ToA with the sought sensor location being the intersection of the circles (black square) and (b) inconsistent ToA measurements. The red stars represent the locations of the anchor base stations.	66

B.3	Extraction of ToA from digital impulse response measured at two points that are spatially close. In (a), the ToA estimate is proportional to k_2 ; whereas in (b), the ToA estimate is proportional to k_1	70
B.4	Singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of Φ in non-increasing order for a multipath environment with $L = 4$ transmitters.	73
B.5	(left) True map, (middle) LocB ($\lambda' = 3.3 \times 10^{-3}$, $\sigma' = 0.5$ m), and (right) LocF ($\lambda = 1.9 \times 10^{-4}$, $\sigma = 37$ m) estimated maps; $N = 300$, $L = 5$, $B = 20$ MHz, and $K = 10$. The black crosses indicate the sensor locations and the solid white lines represent the walls of the building.	77
B.6	Maps of the $M = 10$ LocF features with $L = 5$, $B = 20$ MHz, and $K = 10$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations.	78
B.7	Performance comparison between the LocF cartography ($\lambda = 1.9 \times 10^{-4}$, $\sigma = 37$ m) and the LocB cartography ($\lambda' = 3.3 \times 10^{-3}$, $\sigma' = 0.5$ m) with $B = 20$ MHz and $K = 10$	79
B.8	LocF and LocB map NMSE as a function of the number of walls for different values of the bandwidth, $L = 5$, $N = 300$	79
B.9	LocF estimated map NMSE for different values of number of features M and sensor locations N , with $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.9 \times 10^{-4}$, and $\sigma = 37$ m.	80
B.10	Maps of: (a) all the $M = 10$ features and (b) $r = 4$ reduced features with $L = 5$, $B = 20$ MHz, and $K = 10$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations. The maps are obtained by representing the value of the feature at every location in \mathcal{X}	81
B.11	Estimated map NMSE with reduced features for different r and without reduced features; $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.6 \times 10^{-3}$, and $\sigma = 25$ m.	82
B.12	(top) Average number of missing features and (bottom) estimated map NMSE, both as a function of Γ with $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.9 \times 10^{-4}$, $\mu = 5.42$, and $\sigma = 37$ m.	83
C.1	Model setup and area discretization.	94
C.2	Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying grid points $\{\xi_{i,j}\}$ with $ \mathcal{A}_{i,j} > 0$, and (right) estimated map.	100
C.3	Autoencoder architecture.	101
C.4	Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the grid points $\{\xi_{i,j}\}$ where $ \mathcal{A}_{i,j} > 0$; (center right) and (right): estimated maps. White areas represent buildings.	101
C.5	Comparison with state-of-the-art alternatives. Even though the parameters of the competing algorithms were tuned for this specific experiment, the proposed network offers a markedly better performance.	103

D.1	Model setup and area discretization.	113
D.2	PSD $\Psi(\mathbf{x}, f)$ at location \mathbf{x} using a basis expansion model with Gaussian functions.	117
D.3	Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying grid points $\{\xi_{i,j}\}$ with $ \mathcal{A}_{i,j} > 0$, and (right) map estimate.	119
D.4	Autoencoder architecture.	120
D.5	Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the grid points $\{\xi_{i,j}\}$ where $ \mathcal{A}_{i,j} > 0$; (center right) and (right): map estimates. White areas represent buildings.	123
D.6	Comparison with state-of-the-art alternatives. Training and testing maps drawn from the Gudmundson data set.	124
D.7	Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1]. Training and testing maps drawn from the Gudmundson data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$	124
D.8	Comparison with state-of-the-art alternatives. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 10$	125
D.9	Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1] where the training and testing maps were obtained from the Wireless InSite data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$	125
D.10	RMSE as a function of the code length N_λ , $ \Omega = 104$	126
D.11	RMSE of the proposed network as a function of $ \Omega $ for two types of output layers for the encoder.	126
D.12	Map estimate RMSE of the proposed approach as a function of the number of layers L of the autoencoder for two different activation functions, $ \Omega = 300$	127
D.13	Decoder outputs of autoencoder architectures with different code length and trained with different data sets: (a) $N_\lambda = 4$ with maps from the free-space propagation model, (b) $N_\lambda = 64$ with maps from the Gudmundson data set, $\alpha = 10$	127
D.14	Maps of the true and estimated coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^B$ over \mathcal{X} , $B = 4$	128
D.15	PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ where the basis expansion model uses Gaussian functions. The non-continuous red (green) curves represent the products $\hat{\pi}_b(\mathbf{x})\beta_b(f)$ estimated by the competing (proposed) algorithm.	129
D.16	PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ with a signal basis formed by using raised-cosine functions.	129
D.17	Performance comparison of the proposed scheme with that of the algorithm in [2].	130
D.18	Map estimate RMSE of the proposed scheme for PSD cartography. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 5$	130

List of Tables

B.1	Parameters used for the experiment in Fig. B.8.	78
C.1	Parameters of the proposed network.	101
D.1	Parameters of the proposed network.	120

List of Abbreviations

C2M	channel-to-map
CoM	center of mass
CR	cognitive radio
DoA	direction of arrival
GPS	global positioning system
IRWSRD-LS	iterative re-weighting squared range difference-least squares
KKF	kriged Kalman filtering
KNN	k -nearest neighbors
LocB	location-based
LocF	location-free
LRGeomCG	linear retraction-based geometric conjugate gradient
LTE	long-term evolution
NMSE	normalized mean square error
OFDM	orthogonal frequency-division multiplexing
PCA	principal component analysis
PSD	power spectral density
PU	primary user
RBF	radial basis function
RF	radio frequency
RFID	radio-frequency identification
RKHS	reproducing-kernel Hilbert space
RLS	regularized least squares
RMSE	root mean square error
RSS	received signal strength
SLF	spatial loss field
SNR	signal-to-noise ratio
SR-LS	square-range-based least squares
SVD	singular value decomposition
SVP	singular value projection
TDoA	time difference of arrival
ToA	time of arrival
UAV	unmanned aerial vehicle

UWB ultra-wideband
WiFi wireless fidelity

Part I

Preliminaries

Chapter 1

Introduction

This chapter introduces the research carried out in this thesis. To this end, Sec. 1.1 reviews spectrum cartography and its motivating applications. Sec. 1.2 outlines existing radio map estimation techniques. Sec. 1.3 describes the limitations of these approaches and how the present thesis addresses them. Finally, the structure of this dissertation is provided in Sec. 1.4.

1.1 Spectrum cartography

Spectrum cartography or radio map estimation deals with constructing maps of a certain channel metric such as the received power, power spectral density (PSD), electromagnetic absorption, or channel-gain for every spatial location across the geographic area of interest; see e.g. [2–6]. For example, Fig. 1.1 illustrates a power map in dBm of a channel with carrier frequency $f = 1400\text{MHz}$, where the received power is high at spatial locations close to the three transmitters and becomes lower with distance to the transmitters. The main idea in radio map estimation is to first deploy sensors to measure the target channel metric at some locations in the considered area and then construct a map by interpolating or extrapolating these measurements.

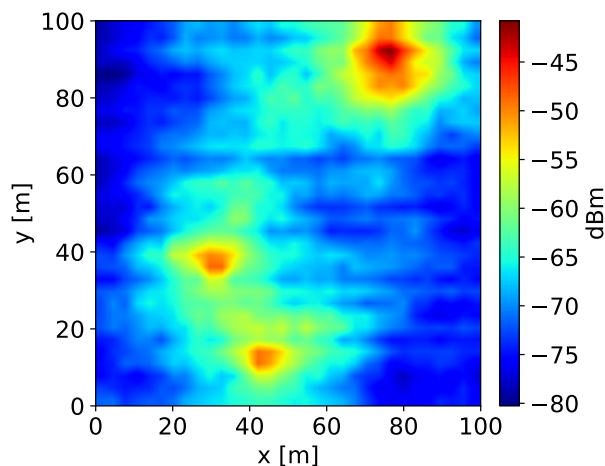


Figure 1.1: Power map at 1400MHz

The constructed radio maps are instrumental in numerous applications within wireless communications. For instance, the transmitter locations [2,7,8] or the kind of propagation terrain [9,10] can be inferred by inspecting a map; see Fig. 1.1. Power maps can also be useful in network design and planning since the former reveal areas of weak coverage, therefore suggesting deployment of new base stations in those areas. PSD maps, which describe the RF signal power as a function of frequency over space, can play a major role in interference mitigation by increasing frequency reuse. This kind of maps may also be of interest to optimize handoff procedures in cellular networks since they allow mobile users to determine the power of all channels in advance. Other applications may include cognitive radio, where radio maps can be informative to secondary users aiming at exploiting underutilized spectrum resources in the space-frequency-time domain. Because they allow autonomous UAVs to account for communication constraints when planning a mission, radio maps have been widely recognized as an enabling technology for UAV communications.

1.2 State-of-the-art

There has been substantial amount of work in the area of radio map estimation, mainly focusing on mapping three channel metrics: power, PSD, and channel-gain. A brief review of existing approaches for each of them is presented next.

Power maps have been constructed by applying interpolation or regression techniques to power measurements acquired by spatially distributed sensors. Some of these schemes include kriging [4,11–13], which allows to estimate the power at unmeasured locations with a weighted linear combination of the power measurements. The kriging weights are given by the linear minimum mean square error estimator [14] based on an assumed statistical relation among the measurements, for instance captured by a spatial covariance function. In [15], the unobserved power values are obtained by formulating the power map estimation problem as a matrix completion task, which is addressed via nuclear norm minimization. Power maps have been also constructed based on compressive sensing [5] by exploiting the sparse spatial distribution of primary users. The works in [16,17] adopt a dictionary learning approach by decomposing the power measurements into linear combinations of channel-gains and transmit-power to construct power maps. In [18,19], power maps have been constructed non-parametrically by estimating the received power at each location as a weighted linear combination of radial basis functions (RBFs). Both the RBF parameters and weights were jointly optimized there. After modeling the path loss with a Laplacian function, a sparse Bayesian learning method is presented in [20] to obtain power maps. A kernel-based learning algorithm is devised in [21], where the power map estimation task involves multiple kernels for capturing pathloss and shadowing.

Since the works mentioned so far can only map the power distribution across space but not across frequency, different approaches have been developed to construct PSD maps. For instance, the schemes in [2,22] use a basis expansion model and exploit the sparsity of power distributions over space and frequency. The work in [23] leverages the framework of kernel-based learning, where power measurements are linearly compressed

and quantized to construct PSD maps. In [24, 25], PSD map estimation is addressed as a tensor completion task.

Instead of mapping power, there exist other approaches that map channel-gain. A representative subset of them are briefly described next. To track the transmit-power and locations of an unknown number of primary users (PUs) via sparse regression in a cooperative cognitive radio (CR) sensing problem, channel-gain maps are constructed and tracked using kriged Kalman filtering (KKF) in [26]. In [27], channel-gain maps are modeled as tomographic “accumulations” of the spatial loss field (SLF) scaled by a weight function, where the former models the absorption of the radio frequency waves at every location in the area of interest. The same work assumes that the SLF has a low-rank structure with sparse deviation and leverages the matrix nuclear norm for estimating the SLF. To avoid introducing heuristic assumptions on the SLF and performing a separate calibration step with measurements in free space, as required in [27, 28], the work in [10] employs kernel-based learning tools to simultaneously learn the SLF and the weight function from attenuation measurements collected by spatially distributed sensors. In order to cope with heterogeneous propagation environments, channel-gain maps have been constructed in [29] by introducing a hidden Markov random field that models a piecewise homogeneous SLF. Moreover, an uncertainty sampling scheme is devised in the same work to adaptively collect informative measurements.

This section described briefly existing radio map estimation schemes. The next section will present their limitations and how the work in this thesis circumvents them.

1.3 Motivation and contributions of the thesis

The ability of all the aforementioned schemes to obtain accurate map estimates is severely impaired by small- and large-scale fading. In particular, one limitation of all existing schemes is that they require accurate knowledge of the sensor locations. For this reason, they are collectively referred to here as *location-based* (LocB) radio map estimation. However, location is rarely known in practice and, therefore, must be estimated from localization features such as the received signal strength (RSS), the time of arrival (ToA), the time difference of arrival (TDoA), or the direction of arrival (DoA), of positioning pilot signals transmitted by satellites (e.g. in GPS) or terrestrial base stations (e.g. in LTE or WiFi [30]) [31, 32]. Unfortunately, accurate location estimates are often not available in practice since fading corrupts those pilot signals, thereby hindering the applicability of existing map estimation techniques, especially in indoor and dense urban scenarios. To see the intuition behind this observation, Figs. 1.2a and 1.2b respectively show the x and y coordinates of the location estimates obtained by applying a state-of-the-art localization algorithm to TDoA measurements of 5 pilot signals received in free space (details of the specific simulation setting can be found in **Paper B**, Sec. B.5). On the other hand, Figs. 1.2c and 1.2d depict the same estimates but in an indoor propagation scenario. As observed, the estimates in the second case are neither accurate nor smooth across space, which prevents any reasonable estimate of a function of them such as a radio map.

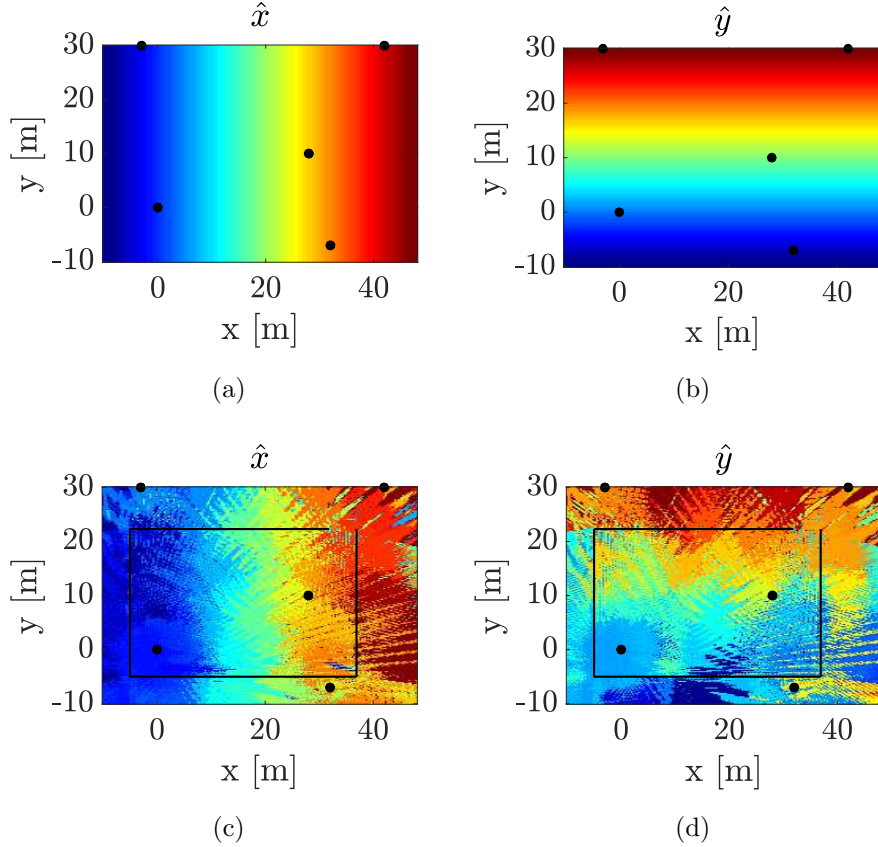


Figure 1.2: Estimation of spatial coordinates using time difference of arrival (TDoA): (a)-(b) in free space, (c)-(d) indoor where the solid black lines represent the walls of the building; the black dots represent the locations of the anchor base stations. The color of each pixel represents the value of the estimated location coordinate at each point in the 150×150 grid area. Because location estimates in (a)-(b) coincide with the true locations, they can act as colorbars to the estimates in (c)-(d).

In the context of indoor localization, there are three main types of systems [33] to counteract this difficulty: (i) those based on ultra-wideband (UWB) [34–36], which require a dedicated infrastructure and, therefore, relatively high costs since synchronized anchor nodes need to be deployed. Thus, localization cannot be carried out in an area where such hardware is not present. (ii) Other indoor positioning systems are based on fingerprinting [33, 37, 38], which involves a manual collection and storage of a data set. This data set may comprise the measured power of multiple beacons at a set of known locations. Note that this process is time consuming and typically expensive because a human or robot should physically go through several known locations to take measurements. Furthermore, if there are significant changes in the propagation environment, these methods would require the acquisition of a new data set. (iii) There exist other indoor positioning systems that combine UWB or fingerprinting with ultrasound [39] or RFID [40]. Thus, they inherit the limitations of (i) and (ii) and require furthermore special sensors and/or line-of-sight propagation conditions.

To sum up, all existing radio map estimation schemes require accurate location information, but this information is not available in dense multipath and indoor scenarios when there are no special localization infrastructure or fingerprinting data sets.

A second limitation of existing radio map estimation approaches is that they rely on interpolation schemes incapable of learning from data and, therefore, are highly ineffective when a map needs to be estimated in channels with a significant fading component. Thus, a substantial performance improvement is expected by learning the spatial signature of relevant propagation effects from past measurements in other environments.

This thesis addresses these limitations along two main directions:

- C1. By proposing the framework of *location-free* (LocF) radio map estimation, where radio maps are constructed by relying on features of the positioning signals rather than on location estimates. Along this direction, additional contributions of this thesis comprise:
 - (a) application of kernel-based learning to devise radio map estimators within this framework;
 - (b) design of pilot signal features suitable for multipath environments;
 - (c) development of a dimensionality reduction scheme motivated by the structure of TDoA measurements to reduce the number of features prior to estimation in order to counteract the so-called *curse of dimensionality* [41, 42];
 - (d) design of a special technique to accommodate scenarios where only a subset of the aforementioned features can be extracted due to low signal-to-noise ratio (SNR) of the localization pilot signals at some measurement locations.

- C2. By proposing the paradigm of *data-driven* radio map estimation methods, where the spatial structure of relevant propagation phenomena such as shadowing, reflection, and diffraction is learned using a data set of past measurements in different environments. Along these lines, additional contributions of this thesis are:
 - (a) development of radio map estimators adhering to this paradigm based on deep neural networks;
 - (b) extension to estimate PSD maps by exploiting the usual parametric structure of wireless communication signals;
 - (c) design of training procedures for learning from a small number of measurements.

Remarkably, this is the first work to estimate radio maps using deep neural networks.

1.4 Structure of the thesis

This dissertation is composed of two parts. **Part I**, which comprises Chapters 1-5, introduces the research carried out in this thesis, summarizes the main findings, and presents

conclusions. **Part II** collects the research papers which present the contributions of this thesis in detail.

The remaining chapters in **Part I** are as follows:

- Chapter 2 deals the algorithm devised in **Papers A** and **B** to develop contribution **C1** of this thesis;
- Chapter 3 briefly describes the approach developed in **Papers C** and **D**, which cover contribution **C2** of this thesis;
- Chapter 4 presents the main results and summarizes the performance evaluation carried out in **Papers A-D** to assess the performance of the algorithms introduced in Chapters 2 and 3;
- Chapter 5 emphasizes the main conclusions and suggests future research directions.

Part II contains the following research papers:

- **Paper A** presents contributions **C1**, **C1a**, and **C1b**. The resulting scheme offers improved prediction performance relative to existing location-based alternatives, as demonstrated by a simulation study in a street canyon scenario. This paper is the conference version of **Paper B**.
- **Paper B** is an extended version of **Paper A**, which additionally includes contributions **C1c** and **C1d** as well as a much more extensive empirical study in dense multipath and indoor environments.
- **Paper C** presents contributions **C2** and **C2a**. Learning the structure of propagation phenomena from past measurements yields estimators that require fewer measurements to attain a target performance relative to existing schemes. This paper is the conference version of **Paper D**.
- **Paper D** is an extended version of **Paper C**, which furthermore includes contributions **C2b** and **C2c** as well as an extensive empirical validation and comparison with existing algorithms in propagation scenarios heavily impaired by large-scale fading.

Chapter 2

Location-Free Radio Map Estimation

As described in Chapter 1, existing radio map estimation schemes rely on accurate location information, which is not available in dense multipath and indoor scenarios. To overcome this limitation, this chapter presents a novel paradigm for radio map estimation, which relies on features of the positioning signals rather than on location estimates. To this end, Sec. 2.1 describes the system model, states the problem, and reviews the basics of LocB radio map estimation. Sec. 2.2 introduces LocF radio map estimation along with the proposed map estimation algorithm, whereas Sec. 2.3 deals with feature design.

2.1 Problem formulation

This section states the general radio map estimation problem and reviews the basics of LocB radio map estimation.

In power map estimation, the goal is to determine the power $p(\mathbf{x})$ of a certain channel, termed *channel-to-map* (C2M), at every location $\mathbf{x} \in \mathcal{X}$ of a geographical region of interest $\mathcal{X} \subset \mathbb{R}^2$. To this end, a certain number of sensors collect N power measurements $\{\tilde{p}_n\}_{n=1}^N$ at locations $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$ not necessarily known, where \tilde{p}_n represents the noisy measurement of the power $p(\mathbf{x}_n)$ at location \mathbf{x}_n .

In LocB radio map estimation [2,4,5,11,15–18,20,22,23,27,29,43], given $\{(\mathbf{x}_n, \tilde{p}_n), n = 1, \dots, N\}$, a fusion center needs to obtain a function estimate $\hat{p}(\mathbf{x})$ that provides the power of the C2M at any query location $\mathbf{x} \in \mathcal{X}$. With this function, a node at location \mathbf{x} can determine the power of the C2M if it knows \mathbf{x} . Note that the above measurement set includes the exact sensor locations $\{\mathbf{x}_n\}_{n=1}^N$. In practice, however, location is typically unknown and, hence, the sensor at the n -th measurement point must obtain an estimate $\hat{\mathbf{x}}_n := \hat{\mathbf{x}}(\mathbf{Y}_n)$ of \mathbf{x}_n by means of some localization algorithm; e.g. [31,32]. Here, \mathbf{Y}_n contains the received pilot signals at the n -th measurement location.

A fusion center then uses $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$ to obtain an estimate $\hat{p}(\mathbf{x})$ of the function $p(\mathbf{x})$. Therefore, if the location estimates $\{\hat{\mathbf{x}}_n\}_{n=1}^N$ are noisy, so will be $\hat{p}(\mathbf{x})$. If a node at an unknown test location wishes to determine the power of the C2M, it will use the pilot signals $\mathbf{Y} \in \mathcal{Y}$ to obtain an estimate $\hat{\mathbf{x}} := \hat{\mathbf{x}}(\mathbf{Y})$ of its location and will evaluate the map estimate as $\hat{p}(\hat{\mathbf{x}})$. Due to multipath propagation effects impacting the pilot signals in \mathbf{Y} , $\hat{\mathbf{x}}(\mathbf{Y})$ may differ considerably from \mathbf{x} , which drastically hinders the estimation of

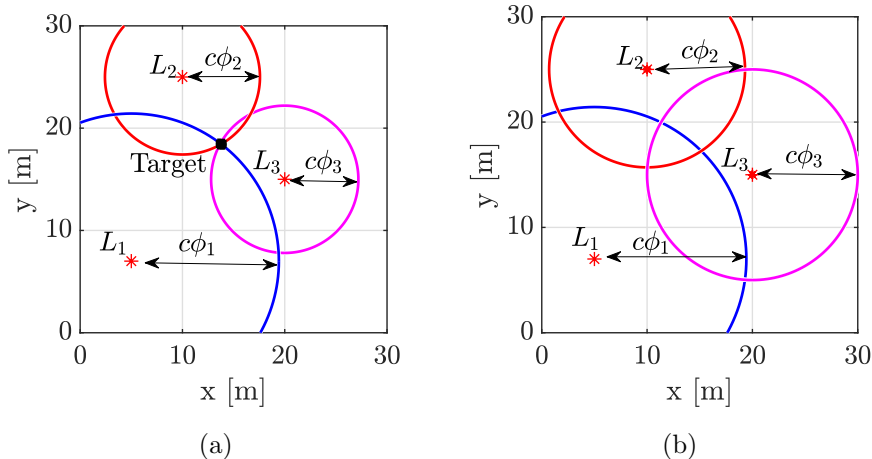


Figure 2.1: Multi-lateration using ToA measurements with circles as possible sensor locations: (a) consistent ToA with the sought sensor location being the intersection of the circles (black square) and (b) inconsistent ToA measurements. The red stars represent the locations of the anchor base stations.

p . Thus, in those cases where the location estimates $\{\hat{\mathbf{x}}(\mathbf{Y}_n)\}_{n=1}^N$ are noisy, the resulting estimate \hat{p} , will also be noisy.

2.2 Location-free approach

This section outlines the location-free (LocF) radio map estimation framework, developed in **Papers A** and **B** to bypass the localization step involved in all existing radio map estimation approaches. To this end, the LocF radio map estimation problem is formulated in Sec. 2.2.1 and solved via kernel-based learning in Sec. 2.2.2.

2.2.1 Formulation

As discussed in Sec. 2.1, existing radio map estimation techniques are heavily impaired by localization errors since the maps they construct are functions of noisy location estimates. The main idea of the proposed framework is to bypass such a dependence.

Recall that the estimated location $\hat{\mathbf{x}}(\mathbf{Y})$ is the result of applying a localization algorithm to the pilot signals \mathbf{Y} . For most existing algorithms, $\hat{\mathbf{x}}(\mathbf{Y})$ can be thought of as the composition of two functions: a function $\phi : \mathcal{Y} \rightarrow \mathcal{F} \subset \mathbb{R}^M$ that obtains M features from $\mathbf{Y} \in \mathcal{Y}$, such as ToA, TDoA, or DoA, and a function $\hat{\mathbf{l}} : \mathcal{F} \rightarrow \mathcal{X}$ that provides a location estimate $\hat{\mathbf{l}}(\phi)$ given a feature vector $\phi \in \mathcal{F}$. The location estimate $\hat{\mathbf{x}}(\mathbf{Y}) = \hat{\mathbf{l}}(\phi(\mathbf{Y}))$ is inaccurate in multipath environments because the algorithm that evaluates $\hat{\mathbf{l}}$ adopts a model where there is a certain “agreement” among features $\phi(\mathbf{Y})$. To see this, consider Fig. 2.1, which illustrates the task of estimating the location of a sensor in an area with $L = 3$ base stations. The features in $\phi \in \mathbb{R}^M$, with $M = L = 3$, used in this example are ToA features. For each pilot signal, there is a circle centered at the base station and whose radius equals c times the ToA, where c is the speed of light. Thus, when there is

no multipath, the ToA features are accurate and the sensor to be located must lie in the intersection of the three circles, as shown in Fig. 2.1a. Hence, the localization algorithm (embodied in $\hat{\mathbf{l}}$) just needs to return the location at which these circles intersect. However, in multipath environments, the ToA features obtained from \mathbf{Y} do not generally equal the time it takes for an electromagnetic wave to propagate from the corresponding base station to the sensor. As a result, the aforementioned circles will not generally intersect; see Fig. 2.1b. In other words, the expected agreement among features is absent and, hence, the localization algorithm will return an inaccurate estimate of the position.

In view of these arguments, the problem in **Papers A** and **B** is to find an estimate $\hat{d}(\phi)$ given $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$, where $\phi_n := \phi(\mathbf{Y}_n)$. By following this approach, the estimated map $\hat{d}(\phi(\mathbf{Y}))$ does not depend on the location estimate. For this reason, this approach will be referred to as *location-free* (LocF) radio map estimation.

For simplicity, the discussion here and in **Papers A** and **B** is based on mapping the power of a single channel. However, the framework is general enough to accommodate PSD or channel-gain maps.

2.2.2 Kernel-based power map learning

This section applies kernel-based learning to provide an algorithm capable of learning the function \hat{d} introduced in Sec. 2.2.1. This is a popular learning approach, mainly due to its simplicity, universality, and typically good performance. Besides, kernel methods have been extensively applied in radio map estimation [10, 21, 23].

In kernel-based learning, one typically seeks \hat{d} in a set of functions known as a *reproducing-kernel Hilbert space* (RKHS) \mathcal{H} and solves a problem of the form

$$\hat{d} = \arg \min_{d \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\tilde{p}_n, \phi_n, d(\phi_n)) + \omega(\|d\|_{\mathcal{H}}), \quad (2.1)$$

where \mathcal{L} is a loss function quantifying the deviation between the observations $\{\tilde{p}_n\}_{n=1}^N$ and the predictions $\{d(\phi_n)\}_{n=1}^N$ returned by a candidate d ; and ω is an increasing function. The first term in the right hand side of (2.1) promotes function estimates that fit the data and the second term promotes estimates that generalize well to unseen data by limiting overfitting. A prominent estimator arises when $\mathcal{L}(\tilde{p}_n, \phi_n, d(\phi_n)) = (\tilde{p}_n - d(\phi_n))^2$ and $\omega(\|d\|_{\mathcal{H}}) = \lambda \|d\|_{\mathcal{H}}^2$, where $\lambda > 0$ is a regularization parameter that balances smoothness and goodness of fit. The resulting \hat{d} is termed *kernel ridge regression* estimate [44, Ch. 4] and is the one used in our experiments in **Papers A** and **B** for simplicity.

The goal is therefore to solve (2.1). However, since \mathcal{H} is generally infinite dimensional, such a task cannot be directly accomplished. Fortunately, one can invoke the so-called *representer theorem* [45], which states that the solution to (2.1) is of the form

$$\hat{d}(\phi) = \sum_{n=1}^N \alpha_n \kappa(\phi, \phi_n) \quad (2.2)$$

for some $\{\alpha_n\}_{n=1}^N$, where κ is a symmetric and positive definite function in \mathcal{H} known as *reproducing kernel* [45]. Although the representer theorem does not provide $\{\alpha_n\}_{n=1}^N$,

these coefficients can be obtained by substituting (2.2) into (2.1) and solving the resulting problem with respect to them since they are finite in number. Applying this procedure for kernel ridge regression results in the problem

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \frac{1}{N} \|\tilde{\boldsymbol{p}} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}, \quad (2.3)$$

where $\boldsymbol{\alpha} := [\alpha_1, \dots, \alpha_N]^\top$, $\tilde{\boldsymbol{p}} := [\tilde{p}_1, \dots, \tilde{p}_N]^\top$, and \mathbf{K} is a positive-definite $N \times N$ matrix whose (n, n') -th entry is $\kappa(\boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'})$. Problem (2.3) can be readily solved in closed-form as $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \tilde{\boldsymbol{p}}$. The estimate \hat{d} solving (2.1) for kernel ridge regression can be recovered by substituting the resulting $\{\alpha_n\}_{n=1}^N$ into (2.2). To obtain the predicted power of the C2M at a query location \boldsymbol{x} where the pilot signals are given by \mathbf{Y} , one just evaluates the LocF estimate $\hat{d}(\boldsymbol{\phi}(\mathbf{Y}))$.

2.3 Location-free features

As described in Sec. 2.1, LocB radio map estimation algorithms learn a function of the location estimates. In the machine learning terminology, the *features* are the spatial coordinates of the sensor locations. On the other hand, the features used by LocF radio map estimation are the entries of $\boldsymbol{\phi}(\cdot)$. In principle, $\boldsymbol{\phi}(\mathbf{Y})$ could be set to contain the same features as the ones used by $\hat{\boldsymbol{l}}(\cdot)$. However, it is generally preferable to use features specifically tailored to LocF radio map estimation. This section discusses briefly the design of these features.

2.3.1 Feature extraction

To simplify the exposition, the scenario where sensors are synchronized with the base stations is presented first. A more practical setup, where this synchronization is not required, will be considered next.

2.3.1.1 Sensors are synchronized with base stations

As discussed in Sec. 2.2.1, features used by typical localization algorithms (hence by LocB radio map estimators) (see also **Paper B**, Sec. B.4.1) are not desirable for LocF radio map estimation. Instead, as an alternative feature inspired by the fact that the ToA may be estimated from the maximum of the channel impulse response, a sensor at the n -th measurement point may extract the *center of mass* (CoM) from the l -th pilot signal as

$$\text{CoM}_{l,n} := \frac{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2 k}{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2},$$

where $\hat{h}_{l,n}[k]$ is the estimate of the k -th sample of the discrete-time channel impulse response between the l -th base station and the sensor at the n -th location whereas K is the number of samples. In this case, with L base stations, the feature vector at the n -th sensor location becomes $\boldsymbol{\phi}_n = [\text{CoM}_{1,n}, \dots, \text{CoM}_{L,n}]^\top$. The key advantage of the CoM feature is that it evolves smoothly over space as demonstrated in **Paper B**, Sec. B.4.1.1 and, therefore, is preferable for LocF radio map estimation.

2.3.1.2 Sensors are not synchronized with base stations

To avoid assuming synchronization of the sensors with the base stations, which requires more expensive equipment and becomes especially challenging in multipath scenarios, the proposed features are of the form

$$\text{CoM}_{l,l',n} := \frac{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2 i}{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2}, \quad (2.4)$$

where $\text{CoM}_{l,l',n}$ is the CoM of the cross-correlation $c_{l,l',n}$ between the l -th and l' -th pilot signals, with $l \neq l'$. With this choice, the feature vector at the n -th measurement location becomes

$$\boldsymbol{\phi}_n = [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^\top. \quad (2.5)$$

The proposed feature has three advantages: i) it is smooth across space as portrayed in Chapter 4, Sec. 4.1.2.1, ii) it does not require synchronization between the localization base stations and the sensors, and iii) it does not require the knowledge of the impulse responses.

2.3.2 Radio map reconstruction from a reduced set of features

As argued in **Paper B**, Sec. B.3.1, learning becomes difficult when the number of input features is high. This section sketches the scheme developed in **Paper B**, Sec. B.4.2, to reduce this number of features in order to improve estimation performance in LocF radio map estimation.

As mentioned in the previous section, in LocB radio map estimation, the feature vectors correspond to the coordinates of the estimated location. Application of the localization algorithm represented by the function $\hat{\boldsymbol{l}}$ naturally reduces dimensionality from the original M features to just 2 or 3. On the other hand, in the case of LocF radio map estimation, a larger number N of measurements to learn $\hat{\boldsymbol{d}}$ may be necessary to attain a target accuracy if M is large. This observation calls for a dimensionality reduction step that condenses the information of the feature vectors $\{\boldsymbol{\phi}_n\}_{n=1}^N \subset \mathbb{R}^M$ into vectors $\{\bar{\boldsymbol{\phi}}_n\}_{n=1}^N \subset \mathbb{R}^r$ of a reduced size r .

In the cases where the feature vectors $\{\boldsymbol{\phi}_n\}_{n=1}^N$ lie close to a low-dimensional subspace, the coordinates of these vectors with respect to a basis for such a subspace may constitute a suitable reduced set of features. First, by considering the scenario of TDoA features, it can be observed (see **Paper B**, Sec. B.4.2) that the rank of $\boldsymbol{\Phi} := [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N]$ is at most $L - 1$ or, equivalently, the vectors $\{\boldsymbol{\phi}_n\}_{n=1}^N$ lie in a subspace of dimension $L - 1$. When effects of noise are noticeable, one would expect that the vectors $\{\boldsymbol{\phi}_n\}_{n=1}^N$ lie *close* to (rather than in) a subspace of dimension $L - 1$.

Similarly, one can expect that when the entries of the vectors $\{\boldsymbol{\phi}_n\}_{n=1}^N$ are given by (2.4), these vectors also lie close to a low-dimensional subspace since CoM features are proportional to the TDoAs in absence of multipath; see **Paper B**, Sec. B.4.1. This phenomenon can be analyzed through simulation (see **Paper B**, Sec. B.5, for more details). To this end, Fig. 2.2 depicts the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of $\boldsymbol{\Phi}$ in non-increasing order for a multipath environment described in **Paper B**, Sec. B.5, with $L = 4$.

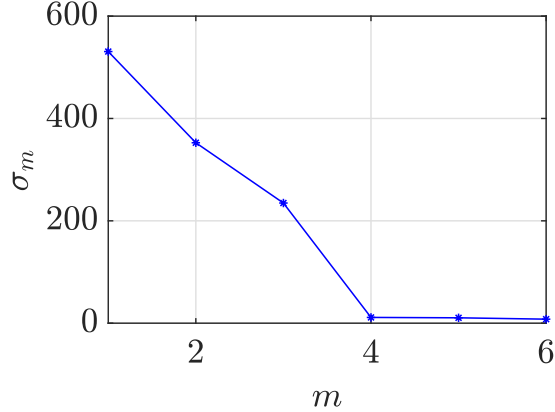


Figure 2.2: Singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of Φ in non-increasing order for a multipath environment with $L = 4$ transmitters.

As expected, roughly $r = L - 1 = 3$ directions capture almost all the energy of the vectors $\{\phi_n\}_{n=1}^N$.

When a set of random vectors lie close to a subspace, a standard approach for dimensionality reduction is principal component analysis (PCA) [42, Ch. 12], which obtains the reduced feature vectors by projecting the input data vectors onto the subspace that preserves most of the energy. This subspace can be determined using the singular value decomposition (SVD) of Φ ; see **Paper B**, Sec. B.4.2. With this technique, the reduced dimensionality feature vectors will be given by $\{\bar{\phi}_n\}_{n=1}^N$, where $\bar{\phi}_n := \mathbf{U}_r^\top \phi_n$ and the columns of \mathbf{U}_r are the left singular vectors of Φ corresponding to its r largest singular values.

The length r of the new feature vectors $\{\bar{\phi}_n\}_{n=1}^N$ may be potentially much smaller than M and is therefore expected to boost estimation performance meaningfully. For instance, when $\{\phi_n\}_{n=1}^N$ are given by (2.5), this reduction is from $M = L(L - 1)/2$ features to $r = L - 1$ features. The LocF map estimate \hat{d} with the aforementioned dimensionality reduction technique is obtained from the pairs $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$ using the approach in Sec. 2.2.2. To evaluate the resulting map at a query location where the received pilot signals are given by \mathbf{Y} , one must simply obtain $\hat{d}(\mathbf{U}_r^\top \phi(\mathbf{Y}))$.

2.3.3 Handling unavailable features

Due to propagation effects, the signal-to-noise ratio (SNR) of some of the received pilot signals may be too low, which implies that the features extracted from these pilot signals may be unreliable or simply unavailable. This section summarizes the techniques devised in **Paper B**, Sec. B.4.3, to cope with such missing features.

Let $\Omega_f \subset \{1, \dots, M\} \times \{1, \dots, N\}$ be such that $(m, n) \in \Omega_f$ iff the m -th feature is available at the n -th measurement location and define the “incomplete” feature matrix $\check{\Phi} \in (\mathbb{R} \cup \{\text{FiM}\})^{M \times N}$ as

$$(\check{\Phi})_{m,n} = \begin{cases} (\phi_n)_m + \varsigma_{m,n} & \text{if } (m, n) \in \Omega_f \\ \text{FiM} & \text{otherwise,} \end{cases} \quad (2.6)$$

where $\varsigma_{m,n}$ explicitly models error in the feature extraction and the symbol FiM represents that the corresponding feature is missing. Since the matrix $\check{\Phi}$ contains missing features, the LocF radio map estimation scheme presented so far is not directly applicable. The missing features must be filled first. Hence, the goal is, given $\check{\Phi}$, find $\Phi \in \mathbb{R}^{M \times N}$ that agrees with $\check{\Phi}$ on Ω_f . A popular approach to address such a matrix completion task is based on rank minimization [46]. Although the resulting problem, described in **Paper B**, Sec. B.4.3, is non-convex, efficient solvers exist based on convex relaxation [47, 48].

In spite of the fact that the rank minimization approach could, in principle, be used, it suffers from two limitations. First, it does not exploit the prior information that Φ can be well approximated by a matrix of rank r , where r is typically $L - 1$; see Fig. 2.2. Second, the reconstructed matrix is sensitive to the noise present in $\check{\Phi}$; see **Paper B**, Sec. B.4.3, for more details. Thus, an appealing alternative would be

$$\begin{aligned} \mathring{\Phi} := \underset{\Phi}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathcal{P}_{\Omega_f}(\Phi) - \mathcal{P}_{\Omega_f}(\check{\Phi})\|_F^2 \\ \text{subject to} \quad & \Phi \in \mathcal{M}_r, \end{aligned} \tag{2.7}$$

where $\mathcal{M}_r := \{\Phi \in \mathbb{R}^{M \times N} : \operatorname{rank}(\Phi) = r\}$ is the smooth manifold of r -rank $M \times N$ matrices and

$$\begin{aligned} \mathcal{P}_{\Omega_f} : (\mathbb{R} \cup \{\text{FiM}\})^{M \times N} &\longrightarrow \mathbb{R}^{M \times N} \\ \check{\Phi} &\longmapsto \mathcal{P}_{\Omega_f}(\check{\Phi}) \end{aligned}$$

is a projection operator with

$$\left(\mathcal{P}_{\Omega_f}(\check{\Phi})\right)_{m,n} := \begin{cases} (\check{\Phi})_{m,n} & \text{if } (m,n) \in \Omega_f \\ 0 & \text{if } (m,n) \notin \Omega_f. \end{cases}$$

There exist algorithms to find local minima of the non-convex problem (2.7). One example based on manifold optimization [49] is the linear retraction-based geometric conjugate gradient (LRGeomCG) method from [50]. A less computationally expensive alternative is the singular value projection (SVP) method in [51], which is based on the traditional projected subgradient descent method.

After solving (2.7), all the columns of $\mathring{\Phi} := [\mathring{\phi}_1, \dots, \mathring{\phi}_N]$ clearly lie in a subspace of dimension r . From the arguments in Sec. 2.3.2, learning the map can be improved by suppressing this redundancy. Estimating a map using the proposed LocF approach in presence of missing features proceeds as follows. First, the completed matrix $\mathring{\Phi}$ is obtained from matrix $\check{\Phi}$ using LRGeomCG or SVP. Then, one learns \hat{d} from $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$, where $\bar{\phi}_n := \hat{U}_r^\top \mathring{\phi}_n$, using the approach in Sec. 2.2.2. Here, the columns of \hat{U}_r are the left singular vectors of $\mathring{\Phi}$ corresponding to its r largest singular values.

To evaluate the estimated map at a test location, one would require in principle the feature vector $\phi \in \mathbb{R}^M$ at that location or, alternatively, its reduced-dimensionality version $\bar{\phi} \in \mathbb{R}^r$. However, due to the propagation phenomena described earlier, only some of the features of ϕ may be available, which can be collected in the vector $\check{\phi} \in (\mathbb{R} \cup \{\text{FiM}\})^M$. The problem now is to find the reduced-dimensionality feature vector $\bar{\phi}$ given $\check{\phi}$. By exploiting the fact that the feature vector at the testing point $\phi \in \mathbb{R}^M$ approximately lies

in the subspace for which the columns of $\mathring{\mathbf{U}}_r$ form an orthonormal basis, the sought vector is found by using the well-known regularized least squares (RLS) method; see **Paper B**, Sec. B.4.3, for more details.

Chapter 3

Data-Driven Radio Map Estimation

As indicated in Chapter 1, Sec. 1.3, existing radio map estimation schemes rely on interpolation algorithms unable to learn from data. In contrast, the novel approach in this chapter learns the spatial structure of propagation phenomena such as shadowing using a data set of past measurements. Notably, this is the first work to estimate radio maps using deep neural networks. This chapter outlines the framework developed in **Papers C and D**, which constitutes the second main contribution of this thesis; see Sec. 1.3. To this end, Sec. 3.1 formulates the problem of PSD map estimation, whereas Sec. 3.2 introduces the proposed data-driven radio map estimation paradigm and proposes a deep neural network architecture based on completion autoencoders.

3.1 PSD map estimation problem

This section formulates the problem of PSD map estimation. The problem where power maps must be estimated can be recovered as a special case of PSD map estimation in a single frequency.

Consider L transmitters, or sources, located in a geographic region of interest $\mathcal{X} \subset \mathbb{R}^2$ and operating in a certain frequency band. Let $\Upsilon_l(f)$ denote the transmit PSD of the l -th source and let $H_l(\mathbf{x}, f)$ represent the frequency response of the channel between the l -th source and a receiver with an isotropic antenna at location $\mathbf{x} \in \mathcal{X}$.

If the L transmitted signals are uncorrelated, the PSD at $\mathbf{x} \in \mathcal{X}$ is

$$\Psi(\mathbf{x}, f) = \sum_{l=1}^L \Upsilon_l(f) |H_l(\mathbf{x}, f)|^2 + v(\mathbf{x}, f), \quad (3.1)$$

where $v(\mathbf{x}, f)$ aggregates the effects of thermal noise, background radiation noise, and interference from remote sources. A certain number of devices with sensing capabilities, e.g. user terminals in a cellular network, collect PSD measurements $\{\tilde{\Psi}(\mathbf{x}_n, f)\}_{n=1}^N$ at N locations $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$ and at a finite set of frequencies $f \in \mathcal{F}$. These frequency measurements can be obtained using e.g. periodograms or spectral analysis methods such as the Welch method [52].

These measurements are sent to a fusion center, which may be e.g. a base station, a mobile user, or a cloud server, depending on the application. Given $\{(\mathbf{x}_n, \tilde{\Psi}(\mathbf{x}_n, f)), n = 1, \dots, N, f \in \mathcal{F}\}$, the problem that the fusion center needs to solve is to find an estimate $\hat{\Psi}(\mathbf{x}, f)$ of $\Psi(\mathbf{x}, f)$ at every location $\mathbf{x} \in \mathcal{X}$ and frequency $f \in \mathcal{F}$.

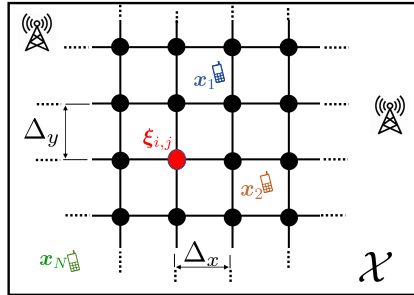


Figure 3.1: Model setup and area discretization.

3.2 Data-driven radio map estimation

All existing map estimators rely on interpolation algorithms that do not learn from data. However, it seems natural that an algorithm can be trained to learn how to solve the problem in Sec. 3.1 using a record of past measurements, possibly in other geographic regions. Specifically, besides $\mathcal{D}:=\{(\mathbf{x}_n, \tilde{\Psi}(\mathbf{x}_n, f)), \mathbf{x}_n \in \mathcal{X}, f \in \mathcal{F}, n = 1, \dots, N\}$, a number of measurement records of the form $\mathcal{D}_t:=\{(\mathbf{x}_{nt}, \tilde{\Psi}_t(\mathbf{x}_{nt}, f)), \mathbf{x}_{nt} \in \mathcal{X}_t, f \in \mathcal{F}, n = 1, \dots, N_t\}, t = 1, \dots, T$, may be available, where \mathcal{D}_t contains N_t measurements collected in the geographic area \mathcal{X}_t and T is the number of training examples. With this additional data, a better performance is expected when estimating $\Psi(\mathbf{x}, f)$.

The rest of this section summarizes the deep learning estimators developed in **Paper D**, Sec. D.3, to address this data-aided formulation. To this end, Sec. 3.2.1 starts by reformulating the problem at hand as a tensor completion task amenable to application of deep neural networks. Subsequently, Sec. 3.2.2 addresses unique aspects of tensor/matrix completion via deep learning. Sec. 3.2.3 discusses briefly how to exploit structure in the frequency domain. Finally, Secs. 3.2.4 and 3.2.5 respectively describe how to learn the spatial structure of propagation phenomena via the notion of *completion autoencoders* and how these networks can be trained in real-world scenarios.

3.2.1 Map estimation as a tensor completion task

Observe that N and N_t depend on the number and movement of the sensors. As argued in **Paper D**, Sec. D.3.1, using a separate estimator for each possible value of N is inefficient in terms of memory, computation, and yields high estimation error. Instead, it is more practical to rely on a single estimator that can accommodate an arbitrary number of measurements.

Given their well-documented merits in a number of tasks, deep neural networks constitute a sensible framework to develop radio map estimators. However, regular feedforward neural networks cannot directly accommodate inputs of a variable size. To bypass this difficulty, the approach pursued here relies on a spatial discretization amenable to application of feedforward architectures [53, Ch. 6]. Similar discretizations have been applied in [15, 24, 54, 55].

This discretization is briefly outlined next for \mathcal{D} ; the extension to \mathcal{D}_t follows the same lines. Define an $N_y \times N_x$ rectangular grid over \mathcal{X} , as depicted in Fig. 3.1, where $\xi_{i,j}$ is the

(i, j) -th grid point, with $i = 1, \dots, N_y$, $j = 1, \dots, N_x$. After assigning the measurement locations to the grid points (as detailed in **Paper D**, Sec. D.3.1), one can therefore collect the true PSD values at the grid points in matrix $\Psi(f) \in \mathbb{R}^{N_y \times N_x}$, $f \in \mathcal{F}$, whose (i, j) -th entry is given by $[\Psi(f)]_{i,j} = \Psi(\xi_{i,j}, f)$. By letting $\mathcal{F} = \{f_1, \dots, f_{N_f}\}$, it is also possible to concatenate these matrices to form the tensor $\Psi \in \mathbb{R}^{N_y \times N_x \times N_f}$, termed *true map*, where $[\Psi]_{i,j,n_f} = \Psi(\xi_{i,j}, f_{n_f})$, $n_f = 1, \dots, N_f$. The PSD measurements at the grid points can also be collected into $\tilde{\Psi}(f) \in \mathbb{R}^{N_y \times N_x}$, defined as

$$[\tilde{\Psi}(f)]_{i,j} = \begin{cases} \tilde{\Psi}(\xi_{i,j}, f) & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise,} \end{cases} \quad (3.2)$$

where $\Omega \subset \{1, \dots, N_y\} \times \{1, \dots, N_x\}$ contains the locations of the measurements. The entries of $\tilde{\Psi}(f)$ corresponding to grid points where no measurements are collected may be filled with zeros, yet other possibilities are discussed in **Papers C** and **D**. As before, the matrices $\tilde{\Psi}(f)$, $f = 1, \dots, N_f$, can be concatenated to form $\tilde{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f}$, termed *sampled map*, where $[\tilde{\Psi}]_{i,j,n_f} = [\tilde{\Psi}(f_{n_f})]_{i,j}$. With this notation, the radio map estimation problem stated in Sec. 3.1 will be approximated as estimating Ψ given Ω and $\tilde{\Psi}$.

3.2.2 Deep completion networks for radio map estimation

The given data in the problem formulation at the end of Sec. 3.2.1 cannot be handled by plain feedforward neural networks since they cannot directly accommodate set-valued inputs like Ω . This section summarizes how this difficulty is addressed.

Papers C and **D** start by analyzing the simplest possibilities for completing the tensor of measurements $\tilde{\Psi}$ using a deep neural network. Specifically, it is observed that the number of parameters of the neural network in [56] is prohibitively large, which renders the network challenging to train for a given T . To alleviate this limitation, a simple alternative would be to directly feed $\tilde{\Psi}$ to the neural network and train by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t - p_{\mathbf{w}}(\tilde{\Psi}_t) \right) \right\|_F^2, \quad (3.3)$$

where $p_{\mathbf{w}}$ is the neural network function with weights collected in the vector \mathbf{w} , $\|\mathbf{A}\|_F^2 := \sum_{i,j,n_f} [\mathbf{A}]_{i,j,n_f}^2$ is the squared Frobenius norm of tensor \mathbf{A} , and $\mathcal{P}_{\Omega}(\mathbf{A})$ is defined by

$$[\mathcal{P}_{\Omega}(\mathbf{A})]_{i,j,n_f} = \begin{cases} [\mathbf{A}]_{i,j,n_f} & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

After (3.3) is solved, $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\tilde{\Psi})$. However, because the completion $p_{\mathbf{w}}(\tilde{\Psi})$ does not account for Ω , poor performance is expected since the network cannot distinguish missing entries from measurements close to the filling value. Hence, a preferable alternative is to complement the input map with a binary mask that indicates which entries are observed, along the lines of some works in the image inpainting literature [57]. Specifically, a mask $\mathbf{M}_{\Omega} \in \{0, 1\}^{N_y \times N_x}$ can be used to represent Ω by setting

$$[\mathbf{M}_{\Omega}]_{i,j} = \begin{cases} 1 & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise.} \end{cases} \quad (3.4)$$

To simplify notation, let $\tilde{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f + 1}$ denote a tensor obtained by concatenating $\tilde{\Psi}$ and \mathbf{M}_Ω along the third dimension. The neural network can therefore be trained as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t - p_{\mathbf{w}}(\tilde{\Psi}_t) \right) \right\|_F^2, \quad (3.5)$$

and, afterwards, a tensor $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\tilde{\Psi})$. Then, this scheme is simple to train, inexpensive to test, and is aware of the locations of the measurements.

In addition to the mask containing the information about the locations of the measurements, a certain number $N_m - 1$ of additional masks can be formed to accommodate additional side information, such as the height of buildings at $\xi_{i,j}$ or the kind of propagation terrain (e.g. urban, suburban, etc) where $\xi_{i,j}$ lies, that may assist in map estimation; see Remark 7 in **Paper D**, Sec. D.3.2. Like \mathbf{M}_Ω , these masks can be concatenated to $\tilde{\Psi}$ to form an augmented tensor $\tilde{\Psi}$. The rest of this dissertation will use symbol $\tilde{\Psi}$ to refer to the result of concatenating $\tilde{\Psi}$ with the available masks.

3.2.3 Exploiting structure in the frequency domain

In practice, different degrees of prior information may be available when estimating a PSD map. This section summarizes the techniques devised in **Paper D**, Sec. D.3.3, to exploit such prior information.

3.2.3.1 Frequency separation

When the number of frequencies N_f is significant, **Paper D**, Sec. D.3.3.1, first argues that the plain training approach in (3.5) is likely to be ill-posed in practical scenarios when the network does not enforce or exploit any structure in the frequency domain. In this case, it may be preferable to separate the problem across frequencies, in other words, estimate N_f power maps rather than a single PSD map, by noting that propagation effects at similar frequencies are expected to be similar. Building upon this principle, $p_{\mathbf{w}}$ can operate separately at each frequency f . This means that training can be accomplished through

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{TN_f} \sum_{t=1}^T \sum_{f \in \mathcal{F}} \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t(f) - p_{\mathbf{w}}(\tilde{\Psi}_t(f)) \right) \right\|_F^2, \quad (3.6)$$

where the input $\tilde{\Psi}_t(f) \in \mathbb{R}^{N_y \times N_x \times (1+N_m)}$ is formed by concatenating $\tilde{\Psi}_t(f)$ and N_m masks.

As observed, the number of variables is roughly reduced by a factor of N_f , whereas the “effective” number of training examples has been multiplied by N_f ; cf. number of summands in (3.6). This is a drastic improvement especially for moderate values of N_f . Although such a network would not exploit structure across the frequency domain, the fact that it would be better trained is likely to counteract this limitation in many setups.

3.2.3.2 Output layers for parametric PSD expansions

Real-world communication systems typically adhere to standards that specify transmission masks by means of carrier frequencies, channel bandwidth, roll-off factors, number

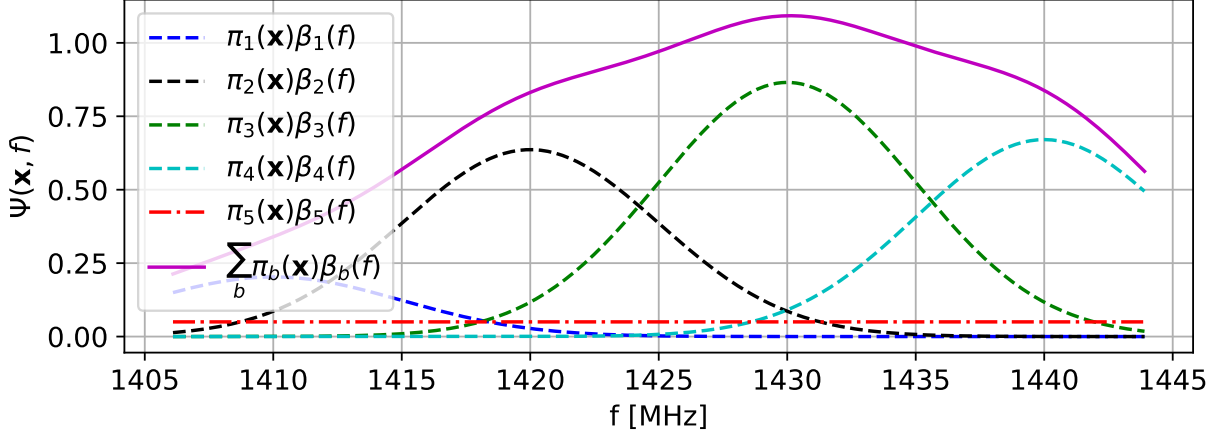


Figure 3.2: PSD $\Psi(\mathbf{x}, f)$ at location \mathbf{x} using a basis expansion model with Gaussian functions.

of OFDM subcarriers, and so on. It seems, therefore, reasonable to capitalize on such prior information for radio map estimation by means of a basis expansion model in the frequency domain like the one in [23, 58, 59]. Even when the frequency form of the transmit PSD is unknown, a basis expansion model is also motivated due to its capacity to approximate any PSD to some extent; e.g. [2, 22].

Under a basis expansion model, the transmit PSD of each source is expressed as

$$\Upsilon_l(f) = \sum_{b=1}^{B-1} \pi_{lb} \beta_b(f), \quad (3.7)$$

where π_{lb} denotes the expansion coefficients and $\{\beta_b(f)\}_{b=1}^{B-1}$ is a collection of $B-1$ given basis functions such as raised-cosine or Gaussian functions. Substituting (3.7) into (3.1), the PSD at $\mathbf{x} \in \mathcal{X}$ is written as

$$\Psi(\mathbf{x}, f) = \sum_{l=1}^L \sum_{b=1}^{B-1} \pi_{lb} \beta_b(f) |H_l(\mathbf{x}, f)|^2 + v(\mathbf{x}, f).$$

Now assume that $|H_l(\mathbf{x}, f)|^2$ remains approximately constant over the support of each basis function, i.e., $|H_l(\mathbf{x}, f)|^2 \approx |H_{lb}(\mathbf{x})|^2$ for all f in the support of $\beta_b(f)$. Then, the PSD at \mathbf{x} can be written as

$$\Psi(\mathbf{x}, f) = \sum_{b=1}^{B-1} \pi_b(\mathbf{x}) \beta_b(f) + v(\mathbf{x}, f), \quad (3.8)$$

where $\pi_b(\mathbf{x}) := \sum_{l=1}^L \pi_{lb} |H_{lb}(\mathbf{x})|^2$. More explanations and justification of the assumption made are provided in **Paper D**, Sec. D.3.3.2.

Observe that the noise PSD $v(\mathbf{x}, f)$ can be similarly expressed in terms of a basis expansion. To simplify the exposition, suppose that $v(\mathbf{x}, f)$ is expanded with a single term as $v(\mathbf{x}, f) \approx \pi_B(\mathbf{x}) \beta_B(f)$, which in turn implies that (3.8) becomes

$$\Psi(\mathbf{x}, f) = \sum_{b=1}^B \pi_b(\mathbf{x}) \beta_b(f). \quad (3.9)$$

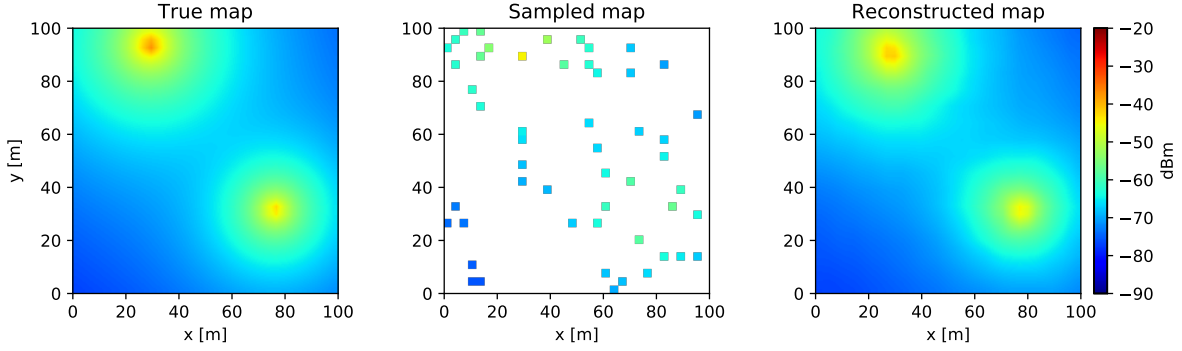


Figure 3.3: Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying the locations of the measurements, and (right) map estimate.

Fig. 3.2 illustrates this expansion for $B = 5$ when $\{\beta_b(f)\}_{b=1}^4$ are Gaussian radial basis functions and $\beta_5(f)$ is set to be constant to model the PSD of white noise. Note that the adopted basis expansion furthermore allows estimation of the noise power $\pi_B(\mathbf{x})$ at every location (see more details in **Paper D**, Sec. D.3.3.2), thereby solving a fundamental problem in applications such as cognitive radio [60].

With the above expansion, the tensor $\Psi \in \mathbb{R}^{N_y \times N_x \times N_f}$ introduced in Sec. 3.2.1 can be expressed as $[\Psi]_{i,j,n_f} = \sum_{b=1}^B [\Pi]_{i,j,b} \beta_b(f_{n_f})$, where tensor $\Pi \in \mathbb{R}^{N_y \times N_x \times B}$ contains the coefficients $[\Pi]_{i,j,b} = \pi_b(\xi_{i,j})$. In a deep neural network, this structure can be naturally enforced by setting all but the last layer to obtain an estimate $\hat{\Pi}$ of Π and the last layer to produce $\hat{\Psi}$; see Fig. 3.4 in Sec. 3.2.4. Specifically, the neural network can be expressed schematically as

$$\begin{array}{ccc} \mathcal{L} & \xrightarrow{\bar{p}_w} & \mathbb{R}^{N_y \times N_x \times B} & \xrightarrow{p^{(L)}} & \mathbb{R}^{N_y \times N_x \times N_f} \\ \check{\Psi} & \longrightarrow & \hat{\Pi} & \longrightarrow & \hat{\Psi}, \end{array}$$

where $\mathcal{L} \subset \mathbb{R}^{N_y \times N_x \times (N_f + N_m)}$ is the input space, function $\bar{p}_w(\check{\Psi}) := p_{w_{L-1}}^{(L-1)}(\dots p_{w_1}^{(1)}(\check{\Psi}))$ groups the first $L - 1$ layers, and $p^{(L)}$ denotes the last layer. With this notation, $\hat{\Pi} = \bar{p}_w(\check{\Psi})$ and $\hat{\Psi} = p^{(L)}(\hat{\Pi}) \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\hat{\Psi}]_{i,j,n_f} = \sum_{b=1}^B [\hat{\Pi}]_{i,j,b} \beta_b(f_{n_f})$. Observe that, as reflected by the notation, the last layer $p^{(L)}$ does not involve trainable parameters.

The network structure presented in this section entails a significant reduction in the number of parameters of the network and, as discussed in Sec. 3.2.3.1, contributes to improve estimation performance for a given T .

3.2.4 Deep completion autoencoders

The previous section summarized the proposed approach to complete the tensor of measurements using a deep neural network. This section discusses briefly the architecture of such a network. To capitalize on the manifold structure of radio maps, the proposed architecture is based on deep completion autoencoders.

A (conventional) autoencoder [53, Ch. 12] is a neural network p_w that can be expressed as the composition of a function ϵ_w termed *encoder* and a function δ_w called *decoder*, $p_w(\Phi) = \delta_w(\epsilon_w(\Phi)) \forall \Phi$. The output of the encoder $\lambda := \epsilon_w(\Phi) \in \mathbb{R}^{N_\lambda}$ is referred to as

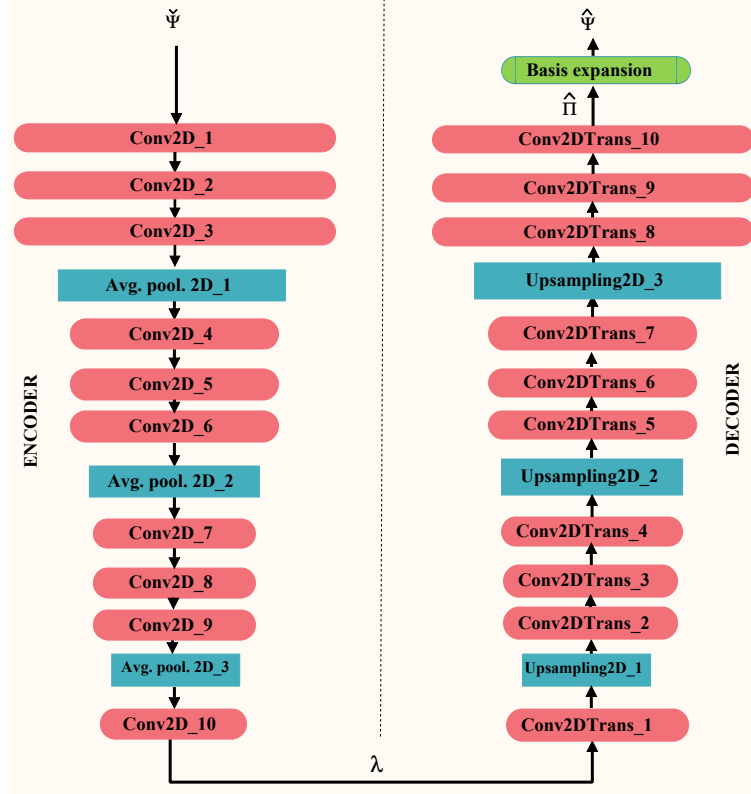


Figure 3.4: Autoencoder architecture.

the *code* or vector of *latent variables* and is, in the typical undercomplete form, of a much lower dimension than the input Φ . An autoencoder is trained to reproduce its input. Since the goal in this work is to complete the tensor of measurements $\tilde{\Psi}$, a *completion* autoencoder is proposed. It adheres to the same principles as conventional autoencoders except for the fact that the encoder must determine the latent variables from a subset of the entries of the input.

As indicated before, undercomplete autoencoders are useful only when most of the information in the input can be condensed in N_λ variables, i.e., when the possible inputs lie close to a manifold of dimension N_λ . To see that this is indeed the case in radio map estimation, an illustrating toy example is presented next. Consider two sources transmitting with a different but fixed power at arbitrary positions in \mathcal{X} and suppose that propagation occurs in free space. All possible spectrum maps in this setup can therefore be uniquely identified by $N_\lambda = 4$ quantities, namely the x and y coordinates of the two sources. Fig. 3.3 illustrates this effect, where the left panel of Fig. 3.3 depicts a true map Ψ and the right panel shows its estimate using the proposed completion autoencoder when $N_\lambda = 4$. The quality of the estimate clearly supports the aforementioned manifold hypothesis. Details about the network and simulation setup are provided in **Paper D**, Sec. D.4. In a real-world scenario, there may be more than two sources, their transmit-power may not always be the same, and there are shadowing effects, which means that $N_\lambda > 4$ will be generally required.

The proposed architecture mainly comprises *convolutional* and *pooling* layers for the encoder with the corresponding *convolution transpose* and *up-sampling* layers for the

decoder. The architecture adopted in **Paper C** comprises also fully connected layers, whereas the one in **Paper D** is fully convolutional as shown in Fig. 3.4.

3.2.5 Training approaches

After discussing the proposed deep neural network architecture in the previous section, this section discusses possible training approaches.

3.2.5.1 Synthetic training data

Since collecting a large number of training maps may be slow or expensive, one can instead generate maps using a mathematical model or simulator that captures the structure of propagation phenomena such as path loss and shadowing; see e.g. [61]. In this case, the neural network can be trained on the data $\{(\check{\Psi}_t, \Psi_t)\}_{t=1}^T$ by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \|\Psi_t - p_{\mathbf{w}}(\check{\Psi}_t)\|_F^2. \quad (3.10)$$

3.2.5.2 Real training data

In practice, real maps may be available for training. However, in most cases, it will not be possible to collect measurements at all grid points. To bypass this difficulty, one may only fit the map at the observed grid points. To improve the training of the autoencoder, the approach proposed here is to use part of the measurements as the input and another part as the output (target). Specifically, for each t , $t = 1, \dots, T$, construct Q_t pairs of (not necessarily disjoint) subsets $\Omega_{t,q}^{(I)}, \Omega_{t,q}^{(O)} \subset \Omega_t$, $q = 1, \dots, Q_t$. Using these subsets, subsample $\check{\Psi}_t$ to yield $\check{\Psi}_{t,q}^{(I)} := \mathcal{P}_{\Omega_{t,q}^{(I)}}(\check{\Psi}_t)$ and $\check{\Psi}_{t,q}^{(O)} := \mathcal{P}_{\Omega_{t,q}^{(O)}}(\check{\Psi}_t)$.

With the resulting $\sum_t Q_t$ training instances, one can think of solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{\sum_t Q_t} \sum_{t=1}^T \sum_{q=1}^{Q_t} \left\| \mathcal{P}_{\Omega_{t,q}^{(O)}} \left(\check{\Psi}_{t,q}^{(O)} - p_{\mathbf{w}} \left(\check{\Psi}_{t,q}^{(I)} \right) \right) \right\|_F^2, \quad (3.11)$$

where $\check{\Psi}_{t,q}^{(I)}$ is formed by concatenating $\check{\Psi}_{t,q}^{(I)}$ and $\mathbf{M}_{\Omega_{t,q}^{(I)}}$.

3.2.5.3 Hybrid training

In practice, one expects to have real data, but only in a limited amount. It then makes sense to apply the notion of *transfer learning* [53, Ch. 15] as follows: first, learn an initial parameter vector \mathbf{w}^* by solving (3.10) with synthetic data. Second, solve (3.11) with real data, but using \mathbf{w}^* as initialization for the optimization algorithm. The impact of choosing this initialization is that the result of solving (3.11) in the second step will be generally closer to a “better” local optimum than if a random initialization were adopted. Hence, this approach combines the information of both synthetic and real data sets.

Chapter 4

Evaluation

The previous two chapters introduced the frameworks of LocF and data-driven map estimation and presented special algorithms based on them that respectively rely on kernel-based learning and deep completion autoencoders. This chapter summarizes the performance evaluation of these algorithms. In particular, Sec 4.1 assesses the performance of the LocF scheme in urban canyon and indoor scenarios whereas Sec 4.2 evaluates the proposed deep neural network architecture in scenarios heavily impaired by large-scale fading.

4.1 Location-free radio map estimation

The performance of the LocF radio map estimation approach summarized in Chapter 2 is evaluated in urban canyon and indoor scenarios, where localization algorithms cannot achieve accurate location estimates. To better illustrate the merits of the LocF framework, kernel ridge regression is used for both LocB [18, 21, 23] and LocF (see Sec. 2.2.2 and **Paper B**) radio map estimation. Both algorithms use Gaussian kernels because of their universality [62]. Quantitative evaluation compares the normalized mean square error (NMSE), defined as

$$\text{NMSE} = \frac{\mathbb{E}\{|p(\mathbf{x}) - \hat{p}(\mathbf{Y}(\mathbf{x}))|^2\}}{\mathbb{E}\{|p(\mathbf{x}) - \bar{p}|^2\}},$$

where \bar{p} is the spatial average of $p(\mathbf{x})$ and $\mathbb{E}\{\cdot\}$ denotes expectation over the locations of the sensors and noise.

4.1.1 Urban canyon

In this case, the urban canyon comprises two parallel vertical planes modeling the walls (or buildings) at each side of the street and a horizontal plane modeling the ground. Propagation is characterized by the so called six-ray model [63]. More details of the simulation setting can be found in **Paper A**, Sec. A.4.

For simplicity and robustness to multipath, the features used in this section by the proposed LocF algorithm are those in Chapter 2, Sec. 2.3.1.1. For localization, the *square-*

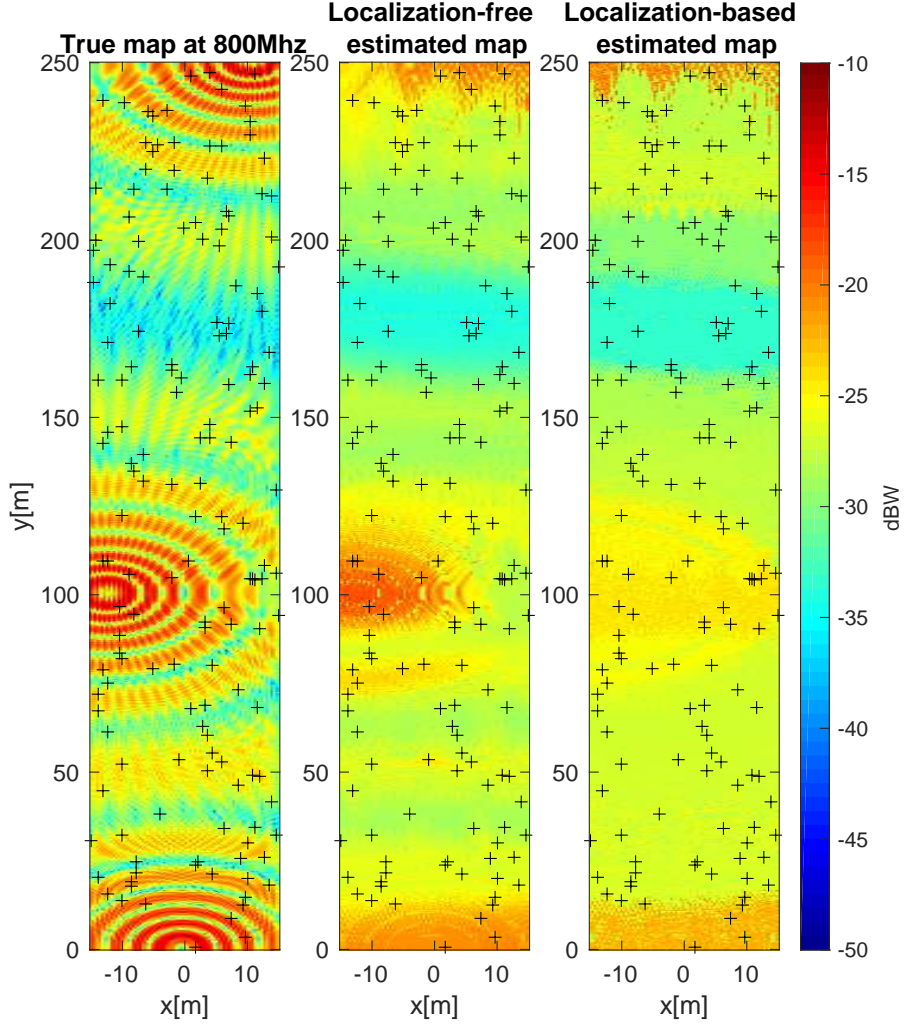


Figure 4.1: (a) True map, (b) LocF, and (c) LocB estimated maps, $N = 160$.

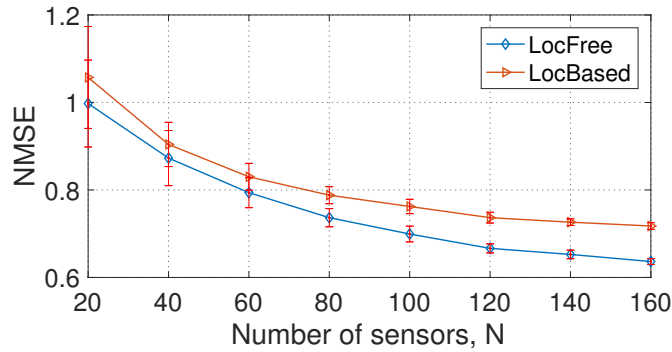


Figure 4.2: Performance comparison between the LocF and LocB radio map estimation curves.

range-based least squares (SR-LS) algorithm [64] is applied to the ToA measurements obtained from the pilots.

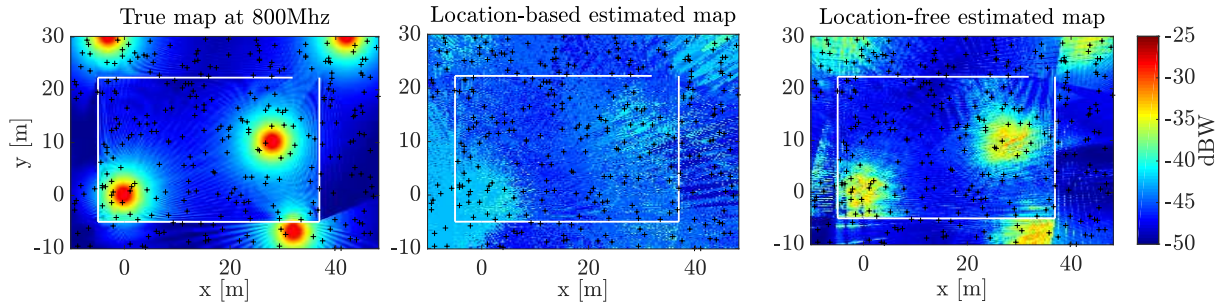


Figure 4.3: (left) True map, (middle) LocB, and (right) LocF estimated maps; $N = 300$. The black crosses indicate the sensor locations and the solid white lines represent the walls of the building.

The true map generated through the canyon model is depicted in the left panel of Fig. 4.1. The middle and right panels respectively show the LocF and LocB map estimates. Visually, the quality of the LocF estimate is higher than that of the LocB estimate. Fig. 4.2 shows the NMSE as a function of the number of sensor measurements N . With significant evidence, one may claim that the proposed LocF radio map estimation scheme outperforms its LocB counterpart when $N > 60$ since the error bars in Fig. 4.2 span over 6 standard deviations of the NMSE across realizations.

4.1.2 Indoor

The simulations are carried out in a structure comprising several parallel vertical planes modeling the walls of a building. Propagation adheres to the *Motley-Keenan multi-wall radio propagation model* [65]. More details of the simulation setup can be found in **Paper B**, Sec. B.5.

4.1.2.1 LocF vs. LocB

To avoid the need for synchronization between transmitters and sensors, the LocF algorithm utilizes the features in Chapter 2, Sec. 2.3.1.2. Additionally, these features provide robustness to multipath and evolve smoothly over space. Since the center of mass that they obtain can be thought of as a lag, it is scaled by the sampling period T and speed of light c to obtain the corresponding range difference, i.e.:

$$\phi_n := Tc [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^T. \quad (4.1)$$

For localization, the *iterative re-weighting squared range difference-least squares* (IRWSRD-LS) algorithm [66] is applied over TDoA features extracted from pilots.

Fig. 4.3 (left) depicts the true map generated through the multi-wall model, the middle and right panels respectively show the LocB and LocF map estimates. It is observed that the quality of the LocF estimate is considerably higher than that of the LocB estimate. The cause for the poor performance of the LocB algorithm is that the location estimates evolve in a non-smooth fashion across space and, then, attempting to learn the C2M from

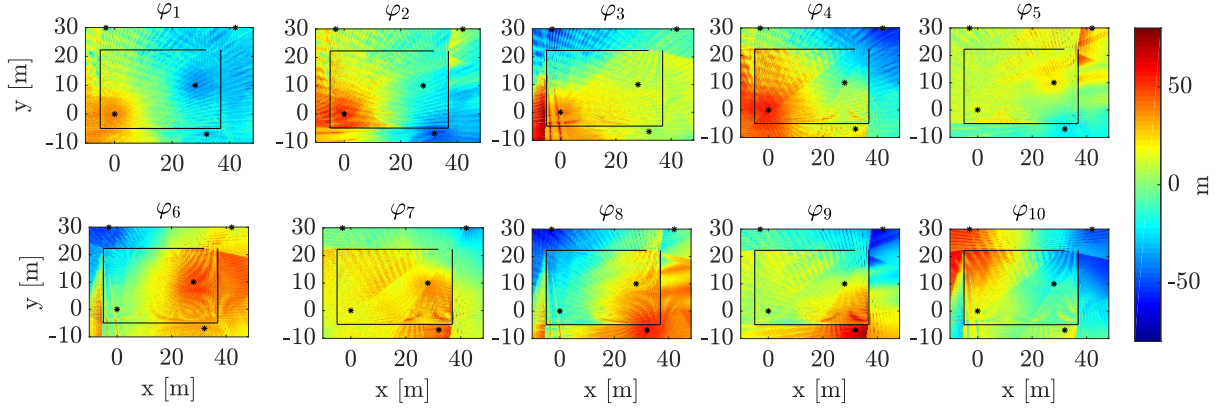


Figure 4.4: Maps of the $M = 10$ LocF features with $L = 5$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations.

such non-smooth features is more challenging; see Figs. 1.2c and 1.2d and the discussion in Sec. 1.3. To illustrate how the LocF approach alleviates this issue, Fig. 4.4 depicts the features used by the LocF estimator across \mathcal{X} . Specifically, if $\phi(\mathbf{x})$ denotes the feature vector, obtained as in (4.1) for location \mathbf{x} , then the m -th panel titled φ_m in Fig. 4.4 corresponds to the m -th entry of $\phi(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{X}$. It is observed that the evolution of these proposed features across space is significantly smoother than the one in Figs. 1.2c and 1.2d.

A quantitative comparison is provided in Fig. 4.5, which shows the NMSE as a function of the number of sensor locations N for $L = 4$ and 7 transmitters. It is observed that, with high significance, the proposed LocF radio map estimation scheme outperforms its LocB counterpart for both values of L when $N > 150$. This thesis also studies the impact of multipath on the LocF and LocB radio map estimation approaches by varying the number of walls. More details about this experiment can be found in **Paper B**, Sec. B.5.1.

4.1.2.2 Feature design

This section summarizes the empirical support in **Paper B**, Sec. B.5.2, for the considerations in Sec. 2.3.2. After investigating the impact of the number of features, which in the simulations of Sec. 4.1.2 was equal to $M = L(L - 1)/2$, it is observed in **Paper B**, Sec. B.5.2, that the NMSE remains approximately the same for $M \geq 7$ in a setup with $L = 5$. Clearly, this effect motivates the feature dimensionality reduction techniques proposed in Sec. 2.3.2. The first step to apply these techniques is to determine the number of reduced features to be used. It is found that $r = 4$ retains at least 99% of the energy of the original features (for more details, please see **Paper B**, Sec. B.5.2). Thus, in principle, a map can be learned using the reduced features $\bar{\phi}_n := \mathbf{U}_4^\top \phi_n \in \mathbb{R}^4$ without meaningfully sacrificing estimation performance. In the same section in **Paper B**, it is observed that the reduced features inherit the spatial smoothness of the original features.

To quantify the impact of reducing the dimensionality of the feature vectors, Fig. 4.6 compares the NMSE of the LocF map estimate that relies on the original features ($M = 10$) with the one that relies on the reduced features ($r = 2, 3, 4$). As observed, using just the 4 reduced features attains a similar performance to the estimator built on the 10

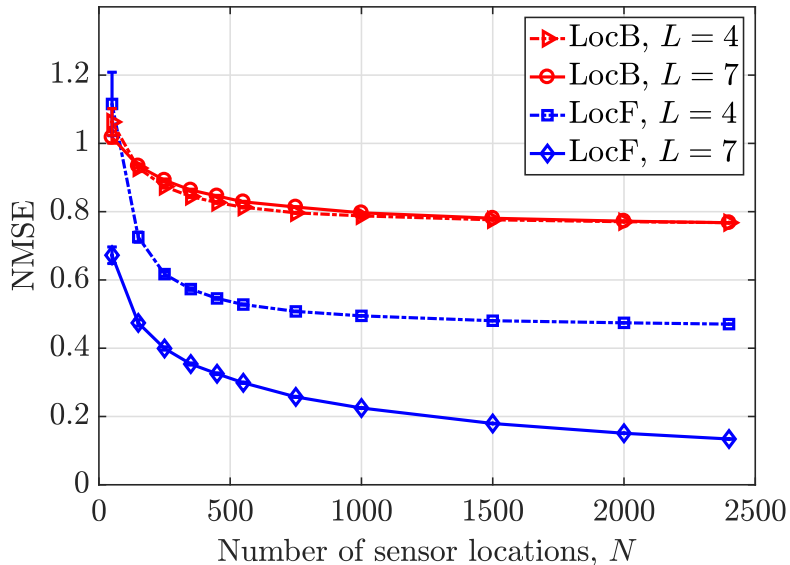


Figure 4.5: Performance comparison between the LocF and LocB radio maps estimation curves.

original features.

4.1.2.3 LocF map estimation with missing features

This section assesses the performance of the approach devised in Sec. 2.3.3 to cope with missing features.

A feature is deemed missing at a given sensor location if the received power of at least one of the two associated pilot signals is below a sensitivity threshold Γ . The top panel of Fig. 4.7 depicts the average number of missing features as a function of Γ . The average is taken with respect to the sensor locations and noise. The bottom panel of Fig. 4.7 shows the LocF map NMSE also as a function of Γ . The matrix completion problem in (2.7) is solved with both singular value projection (SVP) and linear retraction-based geometric conjugate gradient (LRGeomCG); the implementation for the latter is the one provided in the ManOpt toolbox [67]. For higher values of N , the performance of both algorithms is clearly strongly determined by the average number of missing features. SVP seems to outperform LRGeomCG in terms of NMSE. Besides, the computation time of SVP is roughly half the one of LRGeomCG.

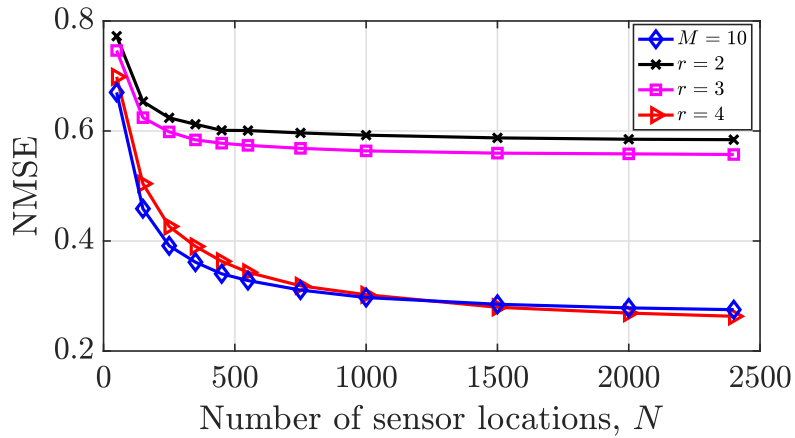


Figure 4.6: Estimated map NMSE with reduced features for different r and without reduced features; $L = 5$.

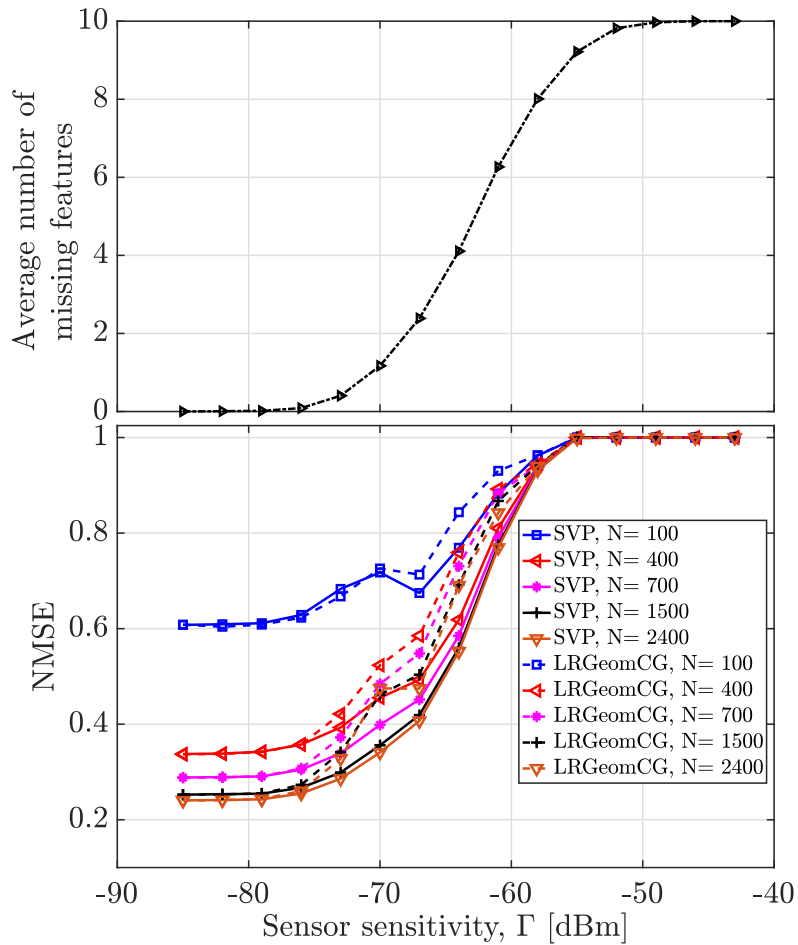


Figure 4.7: (top) Average number of missing features and (bottom) estimated map NMSE, both as a function of Γ with $L = 5$.

4.2 Data-driven radio map estimation

This section summarizes the performance assessment of the proposed deep neural network architecture in scenarios heavily impaired by shadowing.

The region of interest \mathcal{X} is a square area of side 100 m, discretized into a grid with $N_y = N_x = 32$. Two data sets are constructed as summarized next. First, $T = 4 \cdot 10^5$ maps are generated where propagation adheres to the Gudmundson correlated shadowing model [68]. A second data set of $T = 1.25 \cdot 10^5$ maps is generated using Remcom’s Wireless InSite ray tracing software. The network proposed in Chapter 3, Sec. 3.2.4, is implemented in TensorFlow and trained using the ADAM solver [69]. More details of the simulation setup can be found in **Paper D**, Sec. D.4. Quantitative evaluation compares the root mean square error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{\mathbb{E}\{\|\Psi - \hat{\Psi}\|_F^2\}}{N_x N_y N_f}}, \quad (4.2)$$

where Ψ is the true map, $\hat{\Psi}$ is the map estimate, and $\mathbb{E}\{\cdot\}$ denotes expectation over maps, noise, and sensor locations.

4.2.1 Power map estimation

To analyze the most fundamental radio map estimation aspects, \mathcal{F} is set to the singleton $\mathcal{F} = \{1400 \text{ MHz}\}$ in this section.

The proposed algorithm is compared against a representative set of competitors, which includes the kriging algorithm in [4], the multikernel algorithm in [21], the matrix completion via nuclear norm minimization in [1], and the k -nearest neighbors (KNN) algorithm [42].

4.2.1.1 Gudmundson data set

The performance is assessed next using the training approach in Sec. 3.2.5.1 with $\{(\tilde{\Psi}_t, \Psi_t)\}_{t=1}^T$ given by the Gudmundson data set.

To analyze estimation of real maps when the proposed network is trained over synthetic maps, the first experiment shows two map estimates when the true (test) map is drawn from the Wireless Insite data set. Specifically, the first panel of Fig. 4.8 depicts the true map, the second shows $\tilde{\Psi}$, and the remaining two panels show estimates using different numbers of measurements. Observe that with just $|\Omega| = 52$ measurements, the estimate is already of a high quality. The second experiment here compares the RMSE of the proposed method with that of the competing algorithms. From Fig. 4.9, the proposed scheme performs approximately a 25 % better than the next competing alternative. Due to the high RMSE of the matrix completion algorithm in [1] for the adopted range of $|\Omega|$ in Fig. 4.9, its RMSE is shown in Fig. 4.10 along with that of the proposed algorithm for larger values of $|\Omega|$. The proposed method still outperforms this competitor by a wide margin except when the number of measurements is very large, close to $N_y N_x = 1024$.

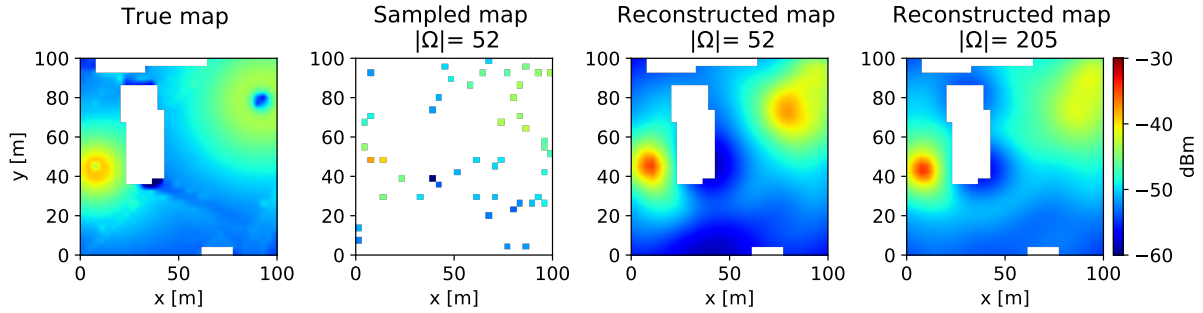


Figure 4.8: Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the measurements; (center right) and (right): map estimates. White areas in the true and reconstructed maps represent buildings.

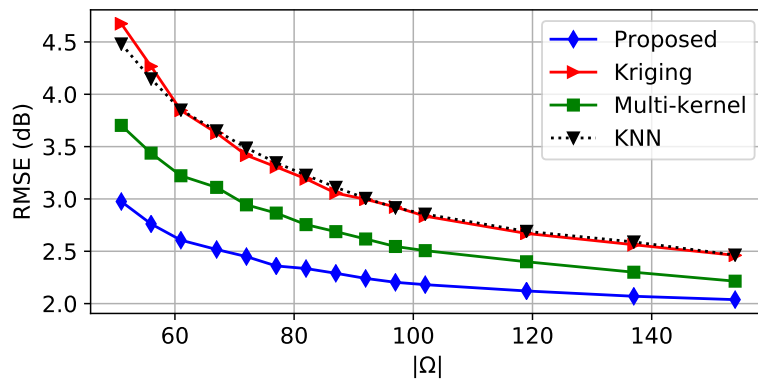


Figure 4.9: Comparison with state-of-the-art alternatives. Training and testing maps drawn from the Gudmundson data set.

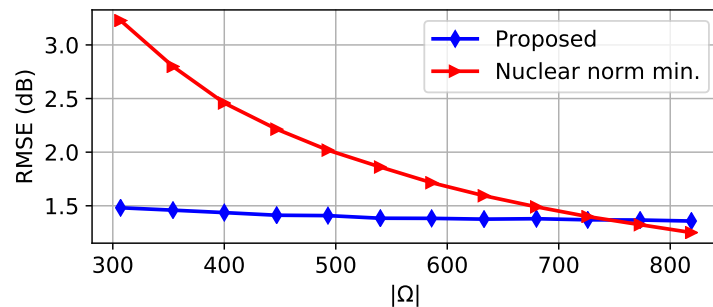


Figure 4.10: Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1]. Training and testing maps drawn from the Gudmundson data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$.

4.2.1.2 Wireless Insite data set

To investigate how the proposed network would perform in real-world scenarios, training uses the Wireless Insite data set in combination with the technique in Sec. 3.2.5.2, where the sets $\Omega_{t,q}^{(I)}$ and $\Omega_{t,q}^{(O)}$ are drawn from Ω_t uniformly at random without replacement with $|\Omega_{t,q}^{(I)}| = |\Omega_{t,q}^{(O)}| = 1/2|\Omega_t|$, $q = 1, \dots, Q_t$, and $Q_t = 10 \forall t$. Fig. 4.11 shows the RMSE as a function of $|\Omega|$ for the proposed scheme and competing alternatives. By the performance

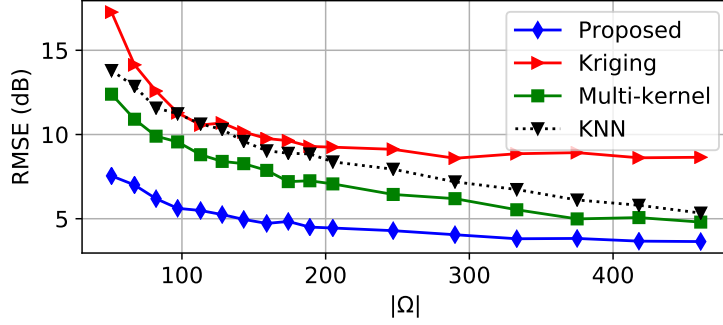


Figure 4.11: Comparison with state-of-the-art alternatives. The training and testing maps were obtained from the Wireless InSite data set.

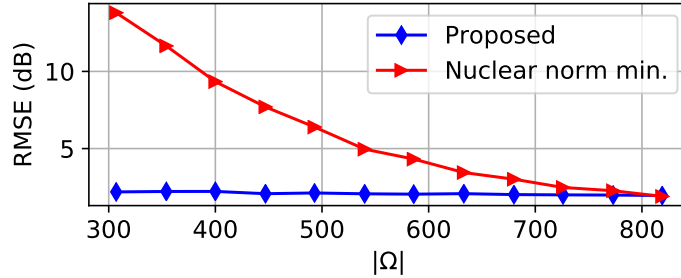


Figure 4.12: Performance comparison of the proposed scheme with that of the matrix completion algorithm in [1] where the training and testing maps were obtained from the Wireless InSite data set, $Q_t = 10$.

degradation of all four approaches relative to Fig. 4.9, it follows that estimating real maps is more challenging than estimating maps in the Gudmundson data set. The performance gap is increased, where the proposed approach now performs roughly 50 % better than the next competing alternative. Again, the algorithm in [1] is not displayed for the same reason as in Fig. 4.9. Its RMSE is shown in Fig. 4.12 along with that of the proposed algorithm. The later still outperforms this competitor except when $|\Omega|$ is very large, close to $N_y N_x$, as in Fig. 4.10.

To justify the main design decisions regarding the proposed network such as the choice of an autoencoder structure, the type of the last layer of the encoder, the number of layers, and the choice of the activation functions, four experiments have been carried out. The findings of these experiments are reported in **Paper D**, Sec. D.4.2. Although neural networks are mainly treated as black boxes, some visualization techniques offer interpretability of the features that they extract and, therefore, shed light on the nature of the information that is learned. To this end, the experiment in **Paper D**, Sec. D.4.3, studies the decoder output when different latent vectors $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda}$ are fed at its input.

4.2.2 PSD map estimation

This section summarizes the empirical support in **Paper D**, Sec. D.4.4, for PSD reconstruction. To this end, two types of signal basis functions are investigated: Gaussian and raised-cosine functions. This summary presents the findings with the raised-cosine functions. The simulation results with the Gaussian functions are provided in **Paper**

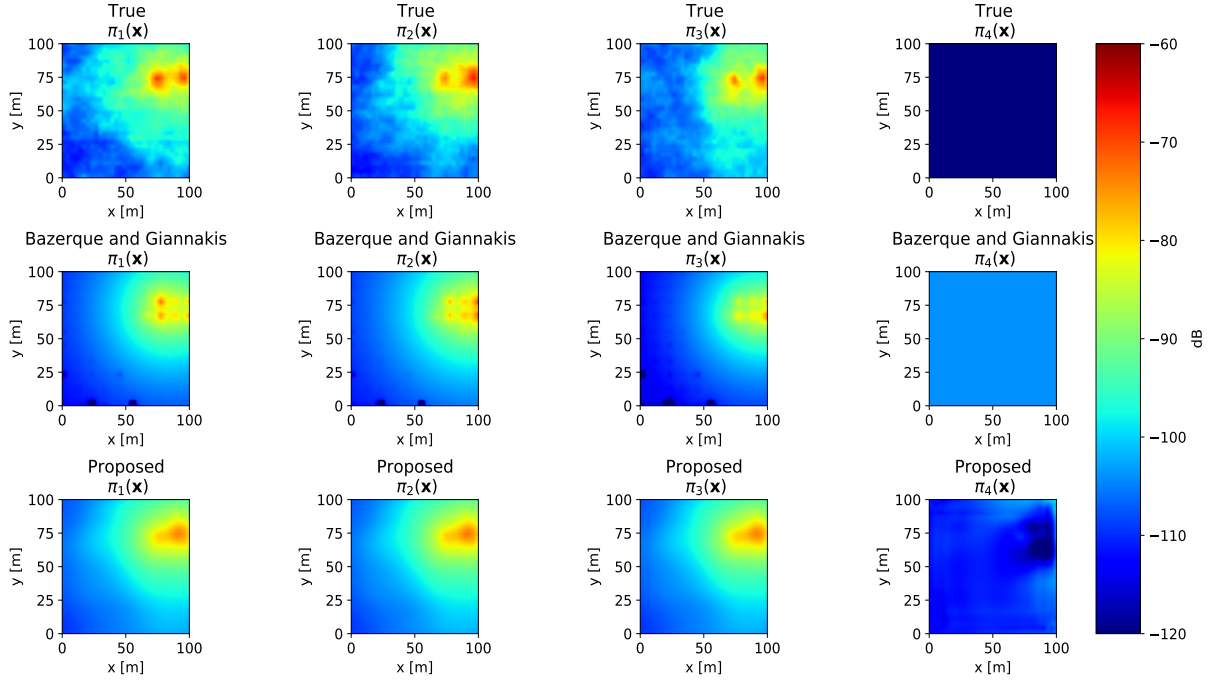


Figure 4.13: Maps of the true and estimated coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^B$ over \mathcal{X} , $B = 4$.

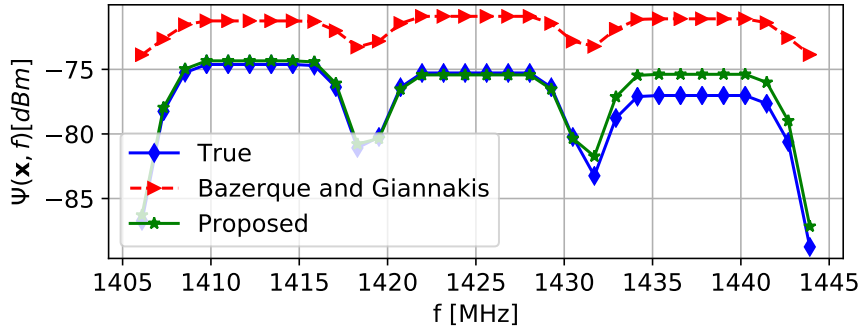


Figure 4.14: PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ with a signal basis formed by using raised-cosine functions.

D, Sec. D.4.4.1. Each sensor samples the received PSD at $N_f = 32$ uniformly spaced frequency values in the band of interest. The performance of the proposed method is compared with that of the non-negative Lasso radio map estimator in [2].

4.2.2.1 Gudmundson data set

The first part of this section summarizes the performance assessment of the proposed scheme using the training approach in Sec. 3.2.5.1 when the training and testing maps were obtained from the Gudmundson data set.

The top row of Fig. 4.13 portrays the maps of the true coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^4$ over \mathcal{X} ; the second and last rows show their estimates with both schemes when $|\Omega| = 512$. Visually, the proposed scheme produces better estimates. To demonstrate the reconstruction quality of the proposed scheme, Fig. 4.14 shows the true and estimated PSDs at a random location $\mathbf{x} \in \mathcal{X}$. As observed, the PSD estimate produced by the proposed scheme follows

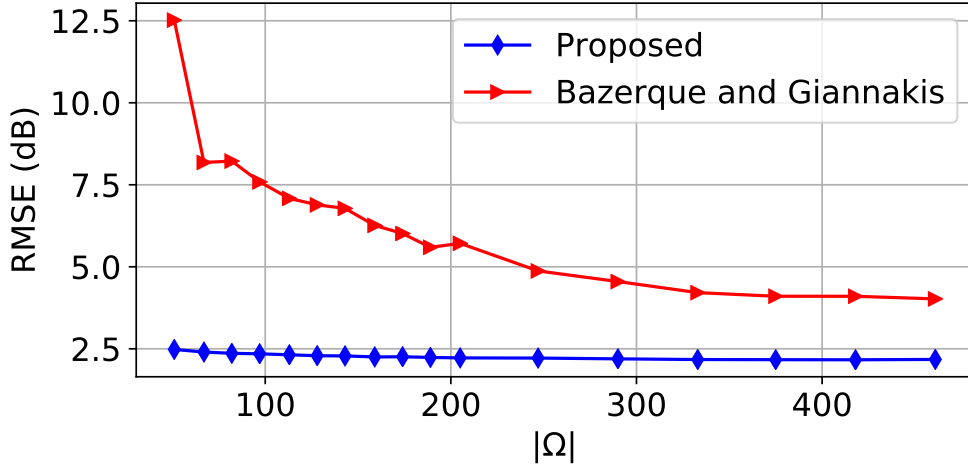


Figure 4.15: Performance comparison of the proposed scheme with that of the algorithm in [2].

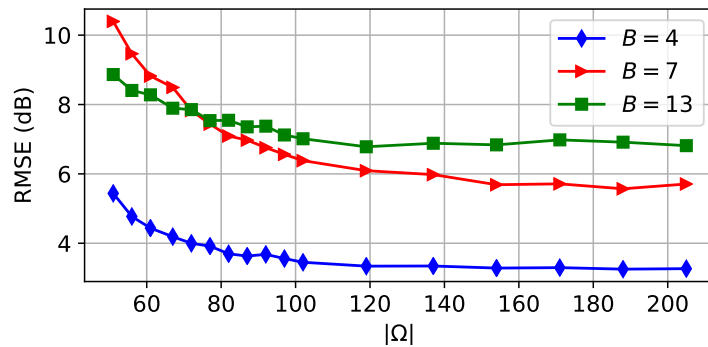


Figure 4.16: Map estimate RMSE of the proposed scheme for PSD cartography. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 5$.

the true PSD more closely compared to the one produced by the competing algorithm. A quantitative comparison is provided in Fig. 4.15, which shows the RMSE as a function of the number of measurements $|\Omega|$. As observed, the proposed method outperforms the competing approach with significant margin for small $|\Omega|$.

4.2.2.2 Wireless Insite data set

The second part of this section summarizes the performance evaluation of the proposed scheme using the training approach in Sec. 3.2.5.2, where the sets $\Omega_{t,q}^{(I)}$ and $\Omega_{t,q}^{(O)}$ are drawn from Ω_t uniformly at random without replacement with $|\Omega_{t,q}^{(I)}| = |\Omega_{t,q}^{(O)}| = 1/2|\Omega_t|$, $q = 1, \dots, Q_t$, and $Q_t = 5 \forall t$. The training and testing maps were obtained from the Wireless InSite data set.

Fig. 4.16 shows the RMSE of the proposed method as a function of the number of measurements $|\Omega|$. Because of the high RMSE of the competing approach [2], its performance is not shown on the figure. As observed, the proposed scheme yields a low RMSE in this realistic scenario which emulates training with real measurements.

Chapter 5

Conclusions and Future Work

This chapter summarizes the main contributions (Sec. 5.1) and puts forth a list of future research directions (Sec. 5.2).

5.1 Summary of contributions

A considerable number of algorithms have been devised to construct power, PSD, and channel-gain maps in the literature. Unfortunately, their ability to obtain accurate map estimates is drastically impaired by small- and large-scale fading. In particular, one of the limitations of all existing schemes is that they require accurate knowledge of the sensor locations, which is not available in practice due to propagation phenomena affecting localization pilot signals such as multipath. A second limitation is that existing approaches rely on interpolation schemes incapable of learning how radio frequency signals propagate and, therefore, they fare poorly in strong fading channels. This thesis circumvents these limitations with the following contributions:

- Location-free (LocF) radio map estimation is proposed as an alternative to classical location-based (LocB) schemes. The central idea is to learn a map as a function of certain features of the localization pilot signals instead of location estimates. Building upon this approach, a map estimator based on kernel ridge regression is developed to estimate power maps from these features. Simulations corroborate the merits of LocF radio map estimation relative to LocB alternatives.
- Features comprising the center of mass of the impulse response or sample cross-correlation of the localization pilots are proposed. In addition to being robust to multipath, the proposed features evolve smoothly over space and, therefore, are preferable for LocF radio map estimation. Inspired by the structure of TDoA measurements, this thesis develops a dimensionality reduction scheme to reduce the number of features prior to learning in order to counteract the curse of dimensionality, thereby improving the estimation performance in the LocF approach. Due to propagation effects, the signal-to-noise ratio of some of the received pilot signals may be too low, which means that the features extracted from these pilots may be unreliable or simply unavailable. This thesis also devises an approach to cope with such missing features based on low-rank matrix completion techniques.

- Data-driven radio map estimation is proposed to learn the spatial structure of propagation phenomena such as shadowing (large-scale fading), reflection, and diffraction. Motivated by the observation that radio maps lie close to a low-dimensional manifold embedded in a high-dimensional space, a deep completion network with an encoder-decoder architecture is proposed to estimate PSD maps. The resulting schemes significantly outperform state-of-the-art alternatives. Notably, this is the first work to propose a deep learning approach for radio map estimation.
- By exploiting the usual parametric structure of wireless communication signals, this thesis develops a methodology for PSD estimation with basis expansion models. Furthermore, training approaches are designed for learning from a small number of measurements.

5.2 Limitations and future work

A few limitations of the algorithms in this thesis are presented next along with possible future research directions.

Concerning the LocF scheme:

- If the location estimates of the measurement sensors are accurate, then the LocB radio map estimation scheme will perform well. In these cases, to attain the same performance, the LocF counterpart may require a higher number of measurements. This motivates hybrid LocB-LocF schemes.
- All the missing features need to be filled prior to learning. Instead of completing them, one could devise schemes which first obtain *local maps* based on available features and then find a *global map* from the locally estimated maps.

Regarding the data-driven approach:

- If the area of interest is densely discretized, the required size of the kernels of convolutional layers of the neural network may be large, which can render the number of training parameters correspondingly large. Therefore, a larger training size would be required to attain a target accuracy. To address this limitation, alternative architectures that do not involve discretizations may be pursued.
- As required in deep neural networks, the size of the training data set is typically large. Collecting such a high number of data samples can be slow or expensive. Furthermore, training neural networks from scratch can be time consuming. Although we started to look into means to effectively acquire measurements in [13], more sophisticated techniques are required.

Since the studied data-driven approach is location-based, meaning that input to the neural network contains information about the locations of the measurements, a possibility to extend the work in this thesis is to devise a *data-driven location-free* approach, which combines the strengths of the two main contributions of this thesis. This approach can possibly be developed within the framework of *federated learning* or collaborative learning,

where the training is executed in a decentralized fashion across multiple edge devices or servers storing local data samples.

References

- [1] G. Ding, J. Wang, Q. Wu, Y. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jan. 2016.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] Y. O. Isselmou, H. Wackernagel, W. Tabbara, and J. Wiart, “Geostatistical interpolation for mapping radio-electric exposure levels,” in *Proc. European Conf. Antennas Propag.*, Nice, France, Nov. 2006, pp. 1–6.
- [4] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [5] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.
- [6] H. B. Yilmaz, T. Tugcu, F. Alagöz, and S. Bayhan, “Radio environment map as enabler for practical cognitive radio networks,” *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 162–169, Dec. 2013.
- [7] P. Huang, O. Castañeda, E. Gönültaş, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, “Improving channel charting with representation-constrained autoencoders,” in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun.*, Cannes, France, Jul. 2019, pp. 1–5.
- [8] M. Joneidi, H. Yazdani, A. Vosoughi, and N. Rahnavard, “Source localization and tracking for dynamic radio cartography using directional antennas,” in *Proc. IEEE Int. Conf. Sensing Commun. Netw.*, Boston, MA, Jun. 2019, pp. 1–9.
- [9] N. Patwari and P. Agrawal, “Effects of correlated shadowing: Connectivity, localization, and RF tomography,” in *Int. Conf. Info. Process. Sensor Networks*, St. Louis, MO, Apr. 2008, pp. 82–93.
- [10] D. Romero, Donghoon Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, 2018.
- [11] G. Boccolini, G. Hernandez-Penalzoza, and B. Bekerull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.

- [12] A. Agarwal and R. Gangopadhyay, “Predictive spectrum occupancy probability-based spatio-temporal dynamic channel allocation map for future cognitive wireless networks,” *Trans. Emerging Telecommun. Technol.*, vol. 29, no. 8, pp. e3442, 2018.
- [13] D. Romero, R. Shrestha, Y. Teganya, and S. P. Chepuri, “Aerial spectrum surveying: Radio map estimation with autonomous UAVs,” in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process., Sep. 2020 (Submitted)*, arXiv preprint arXiv:2005.02432.
- [14] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory*, Prentice-Hall, 1993.
- [15] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [16] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, “Joint link learning and cognitive radio sensing,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.
- [17] S.-J. Kim and G. B. Giannakis, “Cognitive radio spectrum prediction using dictionary learning,” in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [18] M. Hamid and B. Beferull-Lozano, “Non-parametric spectrum cartography using adaptive radial basis functions,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [19] S. Zha, J. Huang, Y. Qin, and Z. Zhang, “An novel non-parametric algorithm for spectrum map construction,” in *Proc. IEEE Int. Symp. Electromagn. Compat.*, Amsterdam, Netherlands, Aug. 2018, pp. 941–944.
- [20] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, “Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [21] J.-A. Bazerque and G. B. Giannakis, “Nonparametric basis pursuit via kernel-based learning,” *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [22] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, “Group-lasso on splines for spectrum cartography,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [23] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.
- [24] M. Tang, G. Ding, Q. Wu, Z. Xue, and T. A. Tsiftsis, “A joint tensor completion and prediction scheme for multi-dimensional spectrum map construction,” *IEEE Access*, vol. 4, pp. 8044–8052, Nov. 2016.

- [25] G. Zhang, X. Fu, J. Wang, and M. Hong, “Coupled block-term tensor decomposition based blind spectrum cartography,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2019, pp. 1644–1648.
- [26] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.
- [27] D. Lee, S.-J. Kim, and G. B. Giannakis, “Channel gain cartography for cognitive radios leveraging low rank and sparsity,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5953–5966, Jun. 2017.
- [28] J. Wilson, N. Patwari, and O. G. Vasquez, “Regularization methods for radio tomographic imaging,” in *Virginia Tech Symp. Wireless Personal Commun.*, Blacksburg, VA, Jun. 2009.
- [29] D. Lee, D. Berberidis, and G. B. Giannakis, “Adaptive bayesian radio tomography,” *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 1964–1977, Mar. 2019.
- [30] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen, “Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on WiMAX networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 283–294, Aug. 2009.
- [31] P. S. Naidu, *Distributed Sensor Arrays: Localization*, CRC Press, 2017.
- [32] A. Bensky, *Wireless Positioning Technologies and Applications*, Artech House, 2016.
- [33] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Trans. Syst., Man, Cybernetics C., Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
- [34] InfSoft, “Ultra-wideband,” [Online]. Available: <https://www.ultrawideband.io/en/technology.php>.
- [35] L. Yang and G. B. Giannakis, “Ultra-wideband communications: An idea whose time has come,” *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 26–54, Nov. 2004.
- [36] D. Dardari, C.-C. Chong, and M. Win, “Threshold-based time-of-arrival estimators in UWB dense multipath channels,” *IEEE Trans. Commun.*, vol. 56, no. 8, pp. 1366–1378, Aug. 2008.
- [37] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Computer Networks*, vol. 47, no. 6, pp. 825–845, Apr. 2005.
- [38] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, “On indoor position location with wireless LANs,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Lisboa, Portugal, Sep. 2002, vol. 2, pp. 720–724.
- [39] HP, “SmartLOCUS,” [Online]. Available: <https://www.rfidjournal.com>.

- [40] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID,” in *Proc. IEEE Int. Conf. Pervasive Computing Commun.*, Fort Worth, TX, Mar. 2003, pp. 407–415.
- [41] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 2007.
- [42] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. Springer, 2006.
- [43] D. Romero, D. Lee, and G. B. Giannakis, “Blind channel gain cartography,” in *Proc. IEEE Global Conf. Signal Inf. Process.*, Greater Washington, D. C., Dec. 2016, pp. 1110–1115.
- [44] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
- [45] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *Proc. Comput. Learning Theory*, Amsterdam, The Netherlands, Jul. 2001, pp. 416–426.
- [46] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, 2002.
- [47] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.
- [48] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM J. Opt.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.
- [49] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2009.
- [50] B. Vandereycken, “Low-rank matrix completion by riemannian optimization,” *SIAM J. Opt.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.
- [51] P. Jain, R. Meka, and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” in *Proc. Advances in Neural Inf. Proces. Syst.*, Vancouver, Canada, Dec. 2010, pp. 937–945.
- [52] P. Stoica and R. L. Moses, “Spectral analysis of signals,” 2005.
- [53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.
- [54] D. Romero, D. Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, Apr. 2018.
- [55] B. R. Hamilton, X. Ma, R. J. Baxley, and S. M. Matechik, “Propagation modeling for radio frequency tomography in wireless networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 1, pp. 55–65, Feb. 2014.

- [56] J. Fan and T. Chow, “Deep learning based matrix completion,” *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.
- [57] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Trans. Graphics*, vol. 36, no. 4, pp. 107, Jul. 2017.
- [58] G. Vázquez-Vilar and R. López-Valcarce, “Spectrum sensing exploiting guard bands and weak channels,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6045–6057, Sep. 2011.
- [59] D. Romero and G. Leus, “Wideband spectrum sensing from compressed measurements using spectral prior information,” *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6232–6246, Dec. 2013.
- [60] R. Tandra and A. Sahai, “SNR walls for signal detection,” *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 4–17, Feb. 2008.
- [61] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of communication systems: modeling, methodology and techniques*, Springer Science & Business Media, 2006.
- [62] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal kernels,” *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006.
- [63] F. Pérez Fontán and P. Mariño Espiñeira, *Modeling the wireless propagation channel: a simulation approach with Matlab*, Wiley, 2008.
- [64] A. Beck, P. Stoica, and J. Li, ,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, 2008.
- [65] S. Hosseinzadeh, H. Larijani, and K. Curtis, “An enhanced modified multi wall propagation model,” in *Proc. IEEE Global Internet of Things Summit*, Geneva, Switzerland, Jun. 2017, pp. 1–4.
- [66] D. Ismailova and W.-S. Lu, “Improved least-squares methods for source localization: An iterative re-weighting approach,” in *Proc. IEEE Int. Conf. Dig. Signal Process.*, Singapore, Jul. 2015, pp. 665–669.
- [67] N. Boumal, B. Mishra, P-A. Absil, and R. Sepulchre, “ManOpt, a MATLAB toolbox for optimization on manifolds,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1455–1459, Apr. 2014.
- [68] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electron. Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.
- [69] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Part II
Publications

Appendix A

Paper A

- Title:** Localization-Free Power Cartography
- Authors:** Yves Teganya, Luis Miguel Lopez-Ramos, Daniel Romero, and Baltasar Beferull-Lozano
- Affiliation:** Dept. of Information and Communication Technology, University of Agder, P. O. Box 509, NO-4898 Grimstad, Norway
- Conference:** *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 3549–3553
- Copyright ©:** IEEE
-

Localization-Free Power Cartography

Yves Teganya, Luis Miguel Lopez-Ramos, Daniel Romero,
and Baltasar Beferull-Lozano

Abstract—Spectrum cartography constructs maps of metrics such as channel gain or received signal power across a geographic area of interest using measurements of spatially distributed sensors. Applications of these maps include network planning, interference coordination, power control, localization, and cognitive radio to name a few. Existing spectrum cartography methods necessitate knowledge of sensor locations, but such locations cannot be accurately determined from pilot positioning signals (such as those in LTE or GPS) in indoor or dense urban scenarios due to multipath. To circumvent this limitation, this paper proposes localization-free cartography, where spectral maps are directly constructed from features of these positioning signals rather than from location estimates. The proposed algorithm capitalizes on the framework of kernel-based learning and offers improved prediction performance relative to existing alternatives, as demonstrated by a simulation study in a street canyon.

Keywords— Spectrum cartography, localization-free cartography, kernel-based learning, spectrum map.

A.1 Introduction

Spectrum cartography constructs maps of a certain channel metric, such as received signal power, interference power, or channel gain over the geographical area of interest [1–3]. Spectral maps are of utmost interest in wireless networks, especially for tasks such as network planning, interference coordination, power control, and dynamic spectrum access [4–6]. Further applications include source localization [2].

Existing approaches typically apply some spatial interpolation or regression technique to measurements collected by spatially distributed sensors. Examples of these approaches for mapping power over space include kriging [1, 7, 8], compressive sensing [3], matrix completion [9], dictionary learning [10, 11], Bayesian models [12], and adaptive radial basis functions [13]. Schemes to map power spectral density (PSD) have also been devised by exploiting the sparsity of power distribution over space and frequency [2] and by leveraging the frameworks of thin-plate spline regression [4, 14] and kernel-based learning [4]. Further schemes have been proposed to map alternative metrics such as channel gain [15–17].

Since all the aforementioned schemes rely on the knowledge of the sensor locations, they will be collectively referred to as *localization-based cartography*. In practice, location is seldom known and therefore it must be estimated from features such as the RSSI, the time (difference) of arrival, or the direction of arrival of positioning pilot signals transmitted by satellites (e.g. in GPS) or terrestrial base stations (e.g. in LTE or

WiFi [18]). Unfortunately, accurate location estimates are often not available in practice due to propagation phenomena affecting those pilot signals such as multipath, which limits the applicability of existing cartography techniques, especially in indoor and dense urban scenarios.

The main contribution of this paper is to circumvent this limitation by proposing localization-free cartography. The idea is that the localization step introduces significant errors in the spectrum map estimation when the aforementioned features are not reliable. Bypassing this step, the proposed approach obtains spectrum maps indexed directly by (or as a function of) the features of the received pilots. As a byproduct of skipping the localization step, the resulting cartography algorithm is also computationally less expensive than its localization-based counterparts. For simplicity, this work focuses on constructing power maps, but the proposed algorithm carries over to other metrics. Such an algorithm is developed within the framework of kernel-based learning not only because of the high simplicity, flexibility, and performance of kernel-based estimators, but also because it has well-documented merits in spectrum cartography [4, 14].

The rest of this paper is organized as follows: Sec. A.2 describes the problem and reviews location-based cartography. Sec. A.3 presents the main contribution of the paper, which is localization-free cartography. Simulations and conclusions are respectively provided in Sec. A.4 and Sec. A.5.

A.2 Preliminaries

The goal is to determine the power $p(\mathbf{x})$ of a certain channel, termed *channel-to-map* (C2M), at every location $\mathbf{x} \in \mathcal{X}$ of the geographical region $\mathcal{X} \subset \mathbb{R}^2$ of interest. To this end, N sensors are deployed across \mathcal{X} at locations $\{\mathbf{x}_n\}_{n=1}^N$ not necessarily known. The n -th sensor acquires a measurement \tilde{p}_n of the power $p(\mathbf{x}_n)$ at its location \mathbf{x}_n .

In localization-based cartography, a fusion center is ideally given pairs $\{(\mathbf{x}_n, \tilde{p}_n)\}_{n=1}^N$, which include the exact sensor locations $\{\mathbf{x}_n\}_{n=1}^N$, and obtains a function estimate $\hat{p}(\mathbf{x}_q)$ that provides the power of the C2M at any query location $\mathbf{x}_q \in \mathcal{X}$. With this function, a node at \mathbf{x}_q can determine the power of the C2M if it knows \mathbf{x}_q . In practice, however, location is typically unknown and hence the n -th sensor must estimate \mathbf{x}_n by relying on pilot signals $\{y_{m,n}[k]\}_{m=1}^M$, where $y_{m,n}[k]$ denotes the k -th sample of the m -th pilot signal received by the n -th sensor. For convenience, form the $M \times K$ matrix \mathbf{Y}_n whose (m, k) -th entry is $y_{m,n}[k]$. From \mathbf{Y}_n , the n -th sensor computes an estimate $\hat{\mathbf{x}}_n(\mathbf{Y}_n)$ of \mathbf{x}_n by means of some localization algorithm; see Sec. A.4 for a specific example. The fusion center then uses $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$ to obtain an estimate $\hat{p}(\mathbf{x})$ of the function $p(\mathbf{x})$. Therefore, if the location estimates $\{\hat{\mathbf{x}}_n\}_{n=1}^N$ are noisy, so will be $\hat{p}(\mathbf{x})$. If a node at a query location \mathbf{x}_q wishes to know the power of the C2M, it will use the pilot signals \mathbf{Y}_q to obtain an estimate $\hat{\mathbf{x}}_q := \hat{\mathbf{x}}(\mathbf{Y}_q)$ of its location and will evaluate the map estimate as $\hat{p}(\hat{\mathbf{x}}_q)$. Here, \mathbf{Y}_q is a matrix whose (m, k) -th entry is given by the k -th sample of the m -th pilot signal $y_{m,q}[k]$ at the query location \mathbf{x}_q . Thus, such an evaluation has two sources of error: first, the location estimation error in $\hat{\mathbf{x}}_q$ and, second, the map estimation error in $\hat{p}(\hat{\mathbf{x}}_q)$.

From a more general perspective, the function that is actually learned in this approach

can be expressed as $p(\mathbf{Y}) := p(\hat{\mathbf{x}}(\mathbf{Y}))$, where $\hat{\mathbf{x}}(\mathbf{Y})$ denotes the output of the chosen localization algorithm when the pilot signals are given by \mathbf{Y} . From this perspective, the problem that is being solved is: given $\{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$, find an estimate $\hat{p}(\mathbf{Y})$ of $p(\mathbf{Y})$. Indeed, localization-based cartography seeks an estimate for the latter function within a certain family of functions that can be expressed as $p(\mathbf{Y}) = g(\hat{\mathbf{x}}(\mathbf{Y}))$ for some function $g : \mathcal{X} \rightarrow \mathbb{R}$. The next section investigates estimates with alternative forms, which will be preferable whenever $\hat{\mathbf{x}}(\mathbf{Y})$ is not an accurate estimator of \mathbf{x} .

Remark 1 *One may argue that a node can determine the power of the C2M at its location more efficiently by measuring it rather than by locating itself and evaluating a map. While this may be the case for a single C2M, determining the power of many C2Ms, or other channel parameters such as the impulse response, may incur a higher cost. In these cases, the benefits of spectrum cartography would be more significant.*

A.3 Localization-free Cartography

This section proposes localization-free cartography, which bypasses the localization step involved in all existing cartography approaches. To this end, the localization-free cartography problem is formulated as a function estimation task in Sec. A.3.1 and solved via kernel-based learning in Sec. A.3.2.

A.3.1 Map Estimate as a Function Composition

From an abstract perspective, spectrum cartography amounts to learning a function $p : \mathbb{C}^{M \times K} \rightarrow \mathbb{R}$ that provides the power $p(\mathbf{Y})$ of the C2M at a location in \mathcal{X} where the pilot signals \mathbf{Y} are received. The *direct approach* to spectrum cartography would be to learn such a function directly from data $\{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$. Since learning a multivariate function up to a reasonable accuracy generally requires the number of data points to be several times larger than the number of input variables, the direct approach would need N to be significantly larger than MK , which is prohibitively large since MK is typically in the order of hundreds or thousands. For this reason, existing (localization-based) cartography schemes do not follow such a direct approach. Instead, they avoid its complexity by confining the search for estimates of $p(\mathbf{Y})$ to those functions that can be expressed as the composition of a fixed function $\hat{\mathbf{x}} : \mathbb{C}^{M \times K} \rightarrow \mathcal{X} \subset \mathbb{R}^2$, where $\hat{\mathbf{x}}(\mathbf{Y})$ corresponds to the output of a localization algorithm when the pilot signals are \mathbf{Y} , and a map function $g : \mathcal{X} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ that needs to be determined; (cf. Sec. A.2). Clearly, finding g requires a significantly smaller N than learning the general function $p : \mathbb{C}^{M \times K} \rightarrow \mathbb{R}$ since g has only two scalar inputs. When $\hat{\mathbf{x}}(\mathbf{Y})$ is a reasonable estimate of the location \mathbf{x} at which \mathbf{Y} has been observed, such a localization-based approach works well. However, due to propagation effects impacting the pilot signals in \mathbf{Y} , $\hat{\mathbf{x}}(\mathbf{Y})$ may be very different from \mathbf{x} and it is easy to see that this drastically hinders the estimation of g . From this observation, it can be concluded that the two scalar outputs of $\hat{\mathbf{x}}(\mathbf{Y})$ fail to capture the relevant information in \mathbf{Y} : more outputs are needed. In summary, neither the above direct approach, which estimates a function with MK inputs, nor the localization-based

approach, which estimates a function of 2 inputs, are appropriate in presence of multipath effects, as is the case in indoors or urban scenarios.

To tackle this difficulty, the proposed approach is to estimate a function whose number of inputs is larger than 2 and smaller than MK . To answer the question on which inputs should be used, it is worth delving further into why the above localization-based approach fails. Localization algorithms typically proceed in two steps: first, they extract some *features* from \mathbf{Y} , and then they feed these features to an algorithm L that exploits a spatial model to determine the location. Those features comprise e.g. estimates of distance, time (difference) of arrival, or angle of arrival. If $\phi(\mathbf{Y}) \in \mathcal{D} \subset \mathbb{R}^M$ denotes the vector stacking these M features and $\mathbf{l}(\phi)$ denotes the output of algorithm L , it follows that $\hat{\mathbf{x}}(\mathbf{Y}) = \mathbf{l}(\phi(\mathbf{Y}))$. The root of the problem is therefore that the model assumed by L is inaccurate: it typically assumes free space propagation, which would imply a certain consistency between the features in $\phi(\mathbf{Y})$ that does not hold in presence of multipath. Combining these observations, a sensible approach is to (i) preserve the dimensionality reduction capability of ϕ (from MK to M); and (ii) avoid the error introduced by $\mathbf{l}(\phi)$. Thus, one can seek *localization-free* function estimates of the form $\hat{p}^{\text{LF}}(\mathbf{Y}) = f(\phi(\mathbf{Y}))$ for some $f : \mathcal{D} \subset \mathbb{R}^M \rightarrow \mathbb{R}$. In this localization-free setup, $\phi(\mathbf{Y})$ comprises M features of the pilot signals, but they need not be those used by the localization algorithms (e.g. time (difference) or angle of arrival). In short, whereas localization-based cartography learns a function of the spatial location estimated from features of the pilot signals, the proposed localization-free approach directly learns a function of such features.

A.3.2 Kernel-based Power Map Learning

This section provides a kernel-based learning algorithm to learn the function f introduced in Sec. A.3.1. Given pairs $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$, where $\phi_n := \phi(\mathbf{Y}_n)$, the regression problem is informally to find f such that $f(\phi(\mathbf{Y})) \approx p(\mathbf{Y})$ for all \mathbf{Y} . To address this problem, one must specify in which family of functions such an f must be found. In kernel-based learning, one seeks f in a set known as a *reproducing-kernel Hilbert space* (RKHS) and given by

$$\mathcal{F} := \left\{ f : f(\phi) = \sum_{i=1}^{\infty} \alpha_i \kappa(\phi, \bar{\phi}_i), \bar{\phi}_i \in \mathcal{D}, \alpha_i \in \mathbb{R} \right\},$$

where $\kappa : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ is a symmetric and positive definite function known as *reproducing kernel* [19]. A common choice is the so-called Gaussian *radial basis function* $\kappa(\phi, \phi') := \exp[-\|\phi - \phi'\|^2/(2\sigma^2)]$, where σ is a parameter selected by the user. Like any Hilbert space, \mathcal{F} has an associated inner product and norm. For an RKHS function $f(\phi) = \sum_{i=1}^{\infty} \alpha_i \kappa(\phi, \bar{\phi}_i)$, the latter is given by

$$\|f\|_{\mathcal{F}}^2 := \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_j \kappa(\bar{\phi}_i, \bar{\phi}_j). \quad (\text{A.1})$$

Kernel-based learning typically solves a problem of the form

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\tilde{p}_n, \phi_n, f(\phi_n)) + \Omega(\|f\|_{\mathcal{F}}), \quad (\text{A.2})$$

where \mathcal{L} is a loss function quantifying the deviation between the observations $\{\tilde{p}_n\}_{n=1}^N$ and the predictions $\{f(\phi_n)\}_{n=1}^N$ returned by a candidate f ; and Ω is an increasing function. The first term in (A.2) promotes function estimates that fit well the data whereas the second term promotes “smooth” estimates; where the notion of smoothness is determined by the RKHS norm $\|\cdot\|_{\mathcal{F}}$. Typical choices are $\mathcal{L}(\tilde{p}_n, \phi_n, f(\phi_n)) = (\tilde{p}_n - f(\phi_n))^2$ and $\Omega(\|f\|_{\mathcal{F}}) = \lambda \|f\|_{\mathcal{F}}^2$, where $\lambda > 0$ is termed regularization parameter and balances smoothness and goodness of fit. For this choice, \hat{f} is termed *kernel ridge regression* estimate [20], and is the one pursued here for simplicity. The goal is therefore to solve (A.2). However, since \mathcal{F} is infinite dimensional in general, (A.2) cannot be directly solved. Fortunately, one can invoke the *representer theorem* [19], which states that the solution to (A.2) is of the form

$$\hat{f}(\phi) = \sum_{n=1}^N \alpha_n \kappa(\phi, \phi_n). \quad (\text{A.3})$$

for some $\{\alpha_n\}_{n=1}^N$. Although the representer theorem does not provide the coefficients $\{\alpha_n\}_{n=1}^N$, they can be obtained by substituting (A.3) into (A.2) and solving the resulting problem with respect to these coefficients. Applying this procedure for kernel ridge regression results in the problem

$$\hat{\alpha} = \arg \min_{\alpha} \frac{1}{N} \|\tilde{\mathbf{p}} - \mathbf{K}\alpha\|^2 + \lambda \alpha^\top \mathbf{K}\alpha, \quad (\text{A.4})$$

where $\alpha := [\alpha_1, \dots, \alpha_N]^\top$, $\tilde{\mathbf{p}} := [\tilde{p}_1, \dots, \tilde{p}_N]^\top$, and \mathbf{K} is an $N \times N$ matrix whose (n, n') -th entry is $\kappa(\phi_n, \phi_{n'})$. Problem (A.4) can be solved in closed form as

$$\hat{\alpha} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}. \quad (\text{A.5})$$

The estimate \hat{f} solving (A.2) for kernel ridge regression can be recovered by substituting (A.5) into (A.3). To obtain the predicted power of the C2M at a query location \mathbf{x}_q where the pilot signals are given by \mathbf{Y}_q , one just evaluates $\hat{p}^{\text{LF}}(\mathbf{Y}_q) = \hat{f}(\phi(\mathbf{Y}_q))$.

A.4 Numerical tests

This section evaluates the performance of localization-free cartography in a scenario with multipath. The latter is a *urban canyon* or *street canyon*, which comprises two parallel vertical planes modeling the walls (or buildings) at each side of the street and a horizontal plane modeling the ground. Propagation is characterized by the so called *six-ray model* [21], which accounts for the direct path, the ground reflection, 2 first-order wall reflections, and 2 wall-to-wall second-order reflections. The sensors are spread uniformly at random over the street, which is 250 m long and 30 m wide.

For simplicity, the pilot signals are impulses centered at time 0 filtered to the pilot channel with bandwidth 5 MHz and carrier frequency 800 MHz, which implies that \mathbf{Y}_n comprises the impulse responses of the bandlimited channels between the M transmitters of pilot signals and the n -th sensor. For simplicity and robustness to timing errors, the

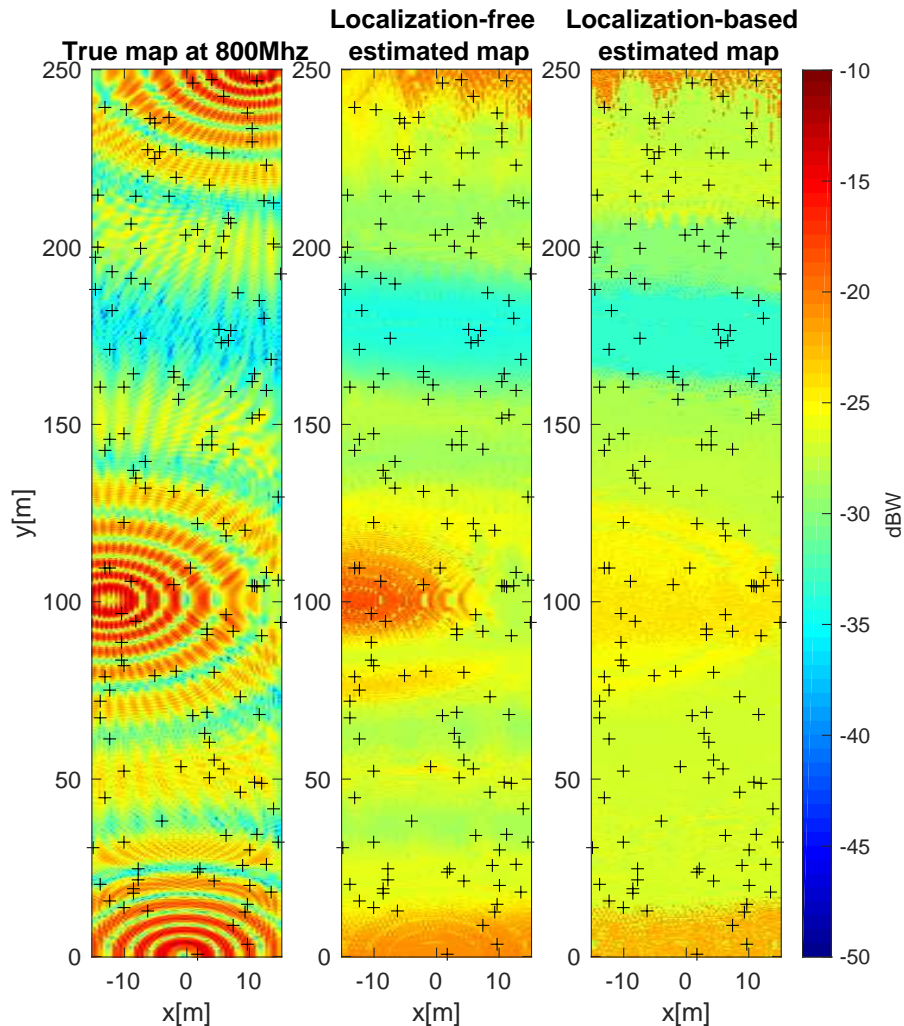


Figure A.1: (a) True map, (b) localization-free, and (c) localization-based estimated maps ($\lambda = 3 \times 10^{-3}$, $N = 160$).

features used by the proposed localization-free algorithm equal the center of mass of the corresponding impulse responses, that is,

$$[\phi_n]_m := \frac{\sum_{k=1}^K t_k |y_{m,n}[k]|^2}{\sum_{k=1}^K |y_{m,n}[k]|^2},$$

where t_k is the time of the k -th sample.

The proposed algorithm, which uses Gaussian radial basis functions with $\sigma = 30$ m, is compared with its localization-based counterpart, which is a special case of the estimators in [2, 4, 22] for estimating power maps. We use Gaussian RBFs because they are universal kernels [23], i.e., able to approximate arbitrary functions. For localization, the *square-range-based least squares* (SR-LS) algorithm [24] is applied to the time-of-arrival measurements obtained from the pilots $\{\mathbf{Y}_n\}_{n=1}^N$. Function g (cf. Sec. A.2) is obtained by applying a similar procedure as in the proposed localization-free algorithm: Given $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$, the estimate of g is given by $g(\hat{\mathbf{x}}_q) = \boldsymbol{\kappa}'^\top(\hat{\mathbf{x}}_q) \hat{\boldsymbol{\beta}}$ where $\boldsymbol{\kappa}'(\hat{\mathbf{x}}_q) :=$

$[\kappa'(\hat{\mathbf{x}}_q, \hat{\mathbf{x}}_1), \dots, \kappa'(\hat{\mathbf{x}}_q, \hat{\mathbf{x}}_N)]^\top$, $\hat{\boldsymbol{\beta}} = (\mathbf{K}' + \lambda N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}$, and \mathbf{K}' is an $N \times N$ matrix with (n, n') -th entry $\kappa'(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_{n'})$ and κ is a Gaussian radial basis function with $\sigma = 35$ m.

Quantitative evaluation will compare the normalized mean square error (NMSE) defined as

$$\text{NMSE} = \frac{\mathbb{E}\{|p(\mathbf{x}) - \hat{p}(\mathbf{Y}(\mathbf{x}) + \boldsymbol{\Upsilon}, \mathcal{T})|^2\}}{\mathbb{E}\{|p(\mathbf{x}) - \bar{p}|^2\}},$$

where $\mathbf{Y}(\mathbf{x})$ comprises the received pilot signals at location \mathbf{x} , $\boldsymbol{\Upsilon}$ represents noise, \bar{p} is the spatial average of $p(\mathbf{x})$, and \mathcal{T} is the training set, defined as $\mathcal{T} := \{(\mathbf{Y}_n + \boldsymbol{\Upsilon}_n, \tilde{p}_n + \epsilon_n)\}_{n=1}^N$ with $\boldsymbol{\Upsilon}_n$ and ϵ_n representing noise. Specifically, $\{\epsilon_n\}_{n=1}^N$ are independent log-normal random variables with zero-mean and standard deviation 0.5 dB (\tilde{p}_n is measured in dBW). Furthermore $\mathbb{E}\{\cdot\}$ denotes expectation over a random location \mathbf{x} uniformly distributed across \mathcal{X} , the locations of the sensors, and noise.

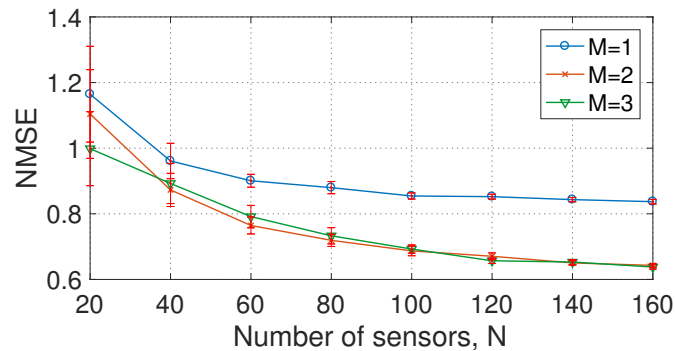
The true map generated through the canyon model is depicted to the left of Fig. A.1. The middle and right panels respectively show the localization-free and localization-based map estimates, which are obtained by placing a query sensor at each location. Black crosses indicate the positions of the N sensors used to estimate the map. As expected, the estimation is better in areas with more sensors. Visually, the quality of the localization-free estimate is higher than that of the localization-based estimate due to multipath.

Fig. A.2a shows the NMSE as a function of N for different numbers M of pilot signals. Each point is obtained by averaging 200 independent Monte Carlo iterations. As anticipated, performance improves with N . Furthermore, for fixed N , the NMSE is non-increasing with M , yet $M = 2$ and 3 yield roughly the same NMSE because of the geometry of the simulation setup.

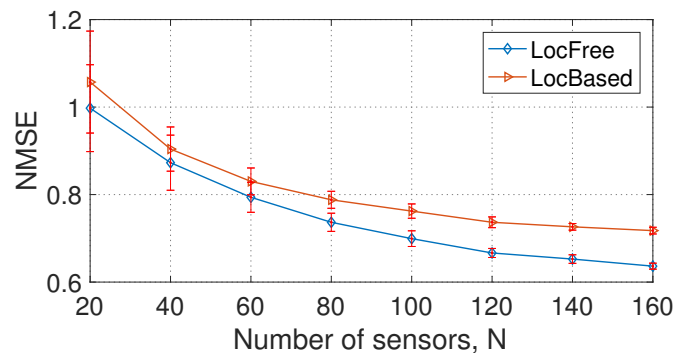
Fig. A.2b shows the NMSE as a function of the number of sensors N used to estimate p^{LF} and p^{LB} . With significant evidence, one may claim that the proposed localization-free cartography scheme outperforms its localization-based counterpart when $N > 60$ since the error bars in Fig. A.2 span over 6 standard deviations of the NMSE across realizations. The reason for a poorer performance of the localization-based scheme is that multipath propagation can mislead the localization algorithm, inducing errors in location estimation that increase deviations in the map estimation as well.

A.5 Conclusions

Localization-free cartography has been proposed as an alternative to classic localization-based schemes, which do not operate properly when multipath impairs the propagation of localization pilot signals. Kernel-ridge regression was applied to estimate power maps from features of those pilot signals collected by a number of sensors. Simulations corroborate the merits of localization-free cartography relative to localization-based methods. Future research will include an extensive simulation study in indoor environments and develop distributed and online extensions.



(a)



(b)

Figure A.2: (a) Estimated map NMSE for different values of number of features, M and sensors, N ; and (b) Performance comparison between the localization-free cartography and the localization-based cartography ($\lambda = 3 \times 10^{-3}, \sigma = 30$ m).

References

- [1] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.
- [4] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.

- [5] S. Grimoud, S. B. Jemaa, B. Sayrac, and E. Moulines, “A REM enabled soft frequency reuse scheme,” in *Proc. IEEE Global Commun. Conf.*, Miami, FL, Dec. 2010, pp. 819–823.
- [6] E. Dall’Anese, S.-J. Kim, G. B. Giannakis, and S. Pupolin, “Power control for cognitive radio networks under channel uncertainty,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Aug. 2011.
- [7] G. Boccolini, G. Hernandez-Penaloza, and B. Beferull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.
- [8] W.C.M.V. Beers and J.P.C. Kleijnen, “Kriging interpolation in simulation: A survey,” in *Proc. IEEE Winter Simulation Conf.*, Washington, D. C., Dec. 2004, vol. 1, pp. 113–121.
- [9] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [10] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, “Joint link learning and cognitive radio sensing,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.
- [11] S.-J. Kim and G. B. Giannakis, “Cognitive radio spectrum prediction using dictionary learning,” in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [12] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, “Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [13] M. Hamid and B. Beferull-Lozano, “Non-parametric spectrum cartography using adaptive radial basis functions,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [14] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, “Group-lasso on splines for spectrum cartography,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [15] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using kriged Kalman filtering,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, feb. 2011.
- [16] D. Romero, D. Lee, and G. B. Giannakis, “Blind channel gain cartography,” in *Proc. IEEE Global Conf. Signal Inf. Process.*, Greater Washington, D. C., Dec. 2016, pp. 1110–1115.

- [17] D. Lee, S.-J. Kim, and G. B. Giannakis, “Channel gain cartography for cognitive radios leveraging low rank and sparsity,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5953–5966, Jun. 2017.
- [18] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen, “Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on WiMAX networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 283–294, Aug. 2009.
- [19] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *Proc. Comput. Learning Theory*, Amsterdam, The Netherlands, Jul. 2001, pp. 416–426.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. Springer, 2006.
- [21] F. Pérez Fontán and P. Mariño Espiñeira, *Modeling the wireless propagation channel: a simulation approach with Matlab*, Wiley, 2008.
- [22] J.-A. Bazerque and G. B. Giannakis, “Nonparametric basis pursuit via kernel-based learning,” *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [23] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal kernels,” *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006.
- [24] A. Beck, P. Stoica, and J. Li, ,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, 2008.

Appendix B

Paper B

Title: Location-Free Spectrum Cartography

Authors: Yves Teganya, Daniel Romero, Luis Miguel Lopez-Ramos, and Baltasar Beferull-Lozano

Affiliation: Dept. of Information and Communication Technology, University of Agder, P. O. Box 509, NO-4898 Grimstad, Norway

Journal: *IEEE Transactions on Signal Processing*, vol. 67, no. 15, pp. 4013–4026, Aug. 2019

Copyright ©: IEEE

Location-Free Spectrum Cartography

Yves Teganya, Daniel Romero, Luis Miguel Lopez-Ramos,
and Baltasar Beferull-Lozano

Abstract—Spectrum cartography constructs maps of metrics such as channel gain or received signal power across a geographic area of interest using spatially distributed sensor measurements. Applications of these maps include network planning, interference coordination, power control, localization, and cognitive radios to name a few. Since existing spectrum cartography techniques require accurate estimates of the sensor locations, their performance is drastically impaired by multipath affecting the positioning pilot signals, as occurs in indoor or dense urban scenarios. To overcome such a limitation, this paper introduces a novel paradigm for spectrum cartography, where estimation of spectral maps relies on features of these positioning signals rather than on location estimates. Specific learning algorithms are built upon this approach and offer a markedly improved estimation performance than existing approaches relying on localization, as demonstrated by simulation studies in indoor scenarios.

Keywords—Spectrum cartography, location-free cartography, kernel-based learning, spectrum map.

B.1 Introduction

Spectrum cartography constructs maps of a certain channel metric, such as received signal power, power spectral density (PSD), or channel gain over a geographical area of interest by relying on measurements collected by radio frequency (RF) sensors [1–3]. The obtained maps are of utmost interest in a number of tasks in wireless communication networks, such as network planning, interference coordination, power control, and dynamic spectrum access [4–6]. For instance, power maps can be useful in network planning since the former indicate areas of weak coverage, thus suggesting locations where new base stations must be deployed. Since PSD maps characterize the distribution of the RF signal power per channel over space, they can play a major role in increasing frequency reuse to mitigate interference. These maps may also be of interest to speed up hand-off in cellular networks since they enable mobile users to determine the power of all channels at a given location without having to spend time measuring it. Additional use cases may include cognitive radios, where secondary users aim at exploiting underutilized spectrum resources in the space-frequency-time domain, or source localization, where the locations of certain transmitters may be estimated by inspecting a map [2].

Existing methods for mapping RF power apply spatial interpolation or regression techniques to power measurements collected by spatially distributed sensors. Some of these methods include kriging [1, 7, 8], orthogonal matching pursuit [3], matrix completion [9], dictionary learning [10, 11], sparse Bayesian learning [12], or kernel-based learning [13, 14]. Since these works can only map power distribution across space but not across frequency, different schemes have been devised to construct PSD maps, for instance by exploiting the sparsity of power distributions over space and frequency with a basis expansion model [2, 15] or by leveraging the framework of kernel-based learning [4]. Rather than mapping power, other families of methods construct channel-gain maps using Kriged Kalman filtering [16], non-parametric regression in reproducing kernel Hilbert spaces (RKHSs) [17], low rank and sparsity [18], or hidden Markov random fields [19].

All the aforementioned schemes require accurate knowledge of the sensor locations. For this reason, they will be collectively referred to as *location-based (LocB) cartography*. However, location is seldom known in practice and therefore must be estimated from features such as the received signal strength, the time (difference) of arrival (T(D)oA), or the direction of arrival (DoA) of positioning pilot signals transmitted by satellites (e.g. in GPS) or terrestrial base stations (e.g. in LTE or WiFi [20]) [21, 22]. Unfortunately, accurate location estimates are often not available in practice due to propagation phenomena affecting those pilot signals such as multipath, which limits the applicability of existing cartography techniques, especially in indoor and dense urban scenarios. To see the intuition behind this observation, Figs. B.1a and B.1b respectively show the x and y coordinates of the location estimates obtained by applying a state-of-the-art localization algorithm to TDoA measurements of 5 pilot signals received in free space (details of the specific simulation setting can be found in Sec. B.5). On the other hand, Figs. B.1c and B.1d depict the same estimates but in an indoor propagation scenario. As observed, the estimates in the second case are neither accurate nor smooth across space, which precludes any reasonable estimate of a spectrum map based on them.

To counteract this difficulty, there are three main types of indoor positioning systems [23]: (i) Those based on ultra-wideband (UWB) [24–26], which require a dedicated infrastructure and relatively high costs, e.g. synchronized anchor nodes in the area where the map has to be constructed. Therefore, localization cannot be carried out in an area where such hardware is not present. (ii) Other indoor positioning systems are based on fingerprinting [23, 27, 28], which involves a manual collection and storage of a dataset. This dataset may comprise the measured power of multiple beacons at a set of known locations. Note that this process is time consuming and typically expensive because a human or robot should physically go through several known locations to take measurements. Furthermore, if there are significant changes in the propagation environment, these methods would require the acquisition of a new dataset. (iii) There exist other indoor positioning systems that combine UWB or fingerprinting with ultrasound [29] or RFID [30]. Thus, they inherit the limitations of (i) and (ii) and require furthermore special sensors and/or line-of-sight propagation conditions. To sum up, all existing cartography schemes require accurate location information, which is not available in dense multipath and indoor scenarios when there are no special localization infrastructure or fingerprinting datasets.

The main contribution of this paper is to address this limitation by proposing the framework of *location-free (LocF) cartography*. The key observation is that inaccurate location estimates introduce significant errors in spectrum map estimation. To bypass this limitation, the proposed approach obtains spectrum maps indexed directly by (or as a function of) features of the received pilot signals. Although many algorithms can be devised within this framework, the present paper develops an algorithm based on kernel-based learning for the sake of exposition. This is not only because of the simplicity, flexibility, and good performance of kernel-based estimators, but also because they have well-documented merits in spectrum cartography [4, 15]. Similarly, the discussion focuses on constructing power maps, but the proposed paradigm carries over to other metrics such as PSD. Remarkably, as a byproduct of skipping the localization step, the resulting cartography algorithm is typically computationally less expensive than its LocB counterparts and does not require additional localization infrastructure or the costly creation of fingerprinting datasets. The second main contribution is a design of pilot signal features tailored to multipath environments. The third contribution is a special technique to accommodate scenarios where a sensor can only extract a subset of those features due to low signal-to-noise ratio (SNR). Finally, the proposed LocF cartography scheme is studied through Monte Carlo simulations in realistic propagation environments. As expected, the proposed scheme outperforms LocB cartography in multipath scenarios, but traditional LocB approaches are still preferable when accurate location estimates are available.

The rest of this paper is structured as follows: Sec. B.2 describes the system model, states the problem, and reviews LocB cartography. Sec. B.3 introduces LocF cartography along with the proposed map estimation algorithm, whereas Sec. B.4 deals with feature design. Numerical tests are presented in Sec. B.5, and conclusions in Sec. B.6.

Notation: Scalars are denoted by lowercase letters. Bold uppercase (lowercase) letters denote matrices (column vectors), \mathbf{I}_N is the $N \times N$ identity matrix and $\mathbf{1}$ is the vector of all ones of appropriate dimension. The symbol $j := \sqrt{-1}$ is the imaginary unit, $(\cdot)^*$ stands for the complex conjugate, while $*$ denotes convolution. Furthermore, operators $(\cdot)^\top$ and $\|\cdot\|_F$ represent transposition and the Frobenius norm, respectively.

B.2 Problem Formulation and LocB Cartography

This section formulates the general spectrum cartography problem and reviews the basics of LocB cartography.

The goal is to determine the power $p(\mathbf{x})$ of a certain channel, termed *channel-to-map (C2M)*, at every location $\mathbf{x} \in \mathcal{X}$ of a geographical region of interest $\mathcal{X} \subset \mathbb{R}^d$, with $d = 2$ or 3 . For example, this C2M can be an uplink or downlink channel of a cellular network as well as a radio or TV broadcasting channel. To this end, a collection of sensors gather N measurements at locations $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$ not necessarily known. The noisy measurement of the power $p(\mathbf{x}_n)$ at location \mathbf{x}_n will be represented as \tilde{p}_n . Since the sensors collect measurements at multiple locations in \mathcal{X} , the number of measurements may be significantly greater than the number of sensors.

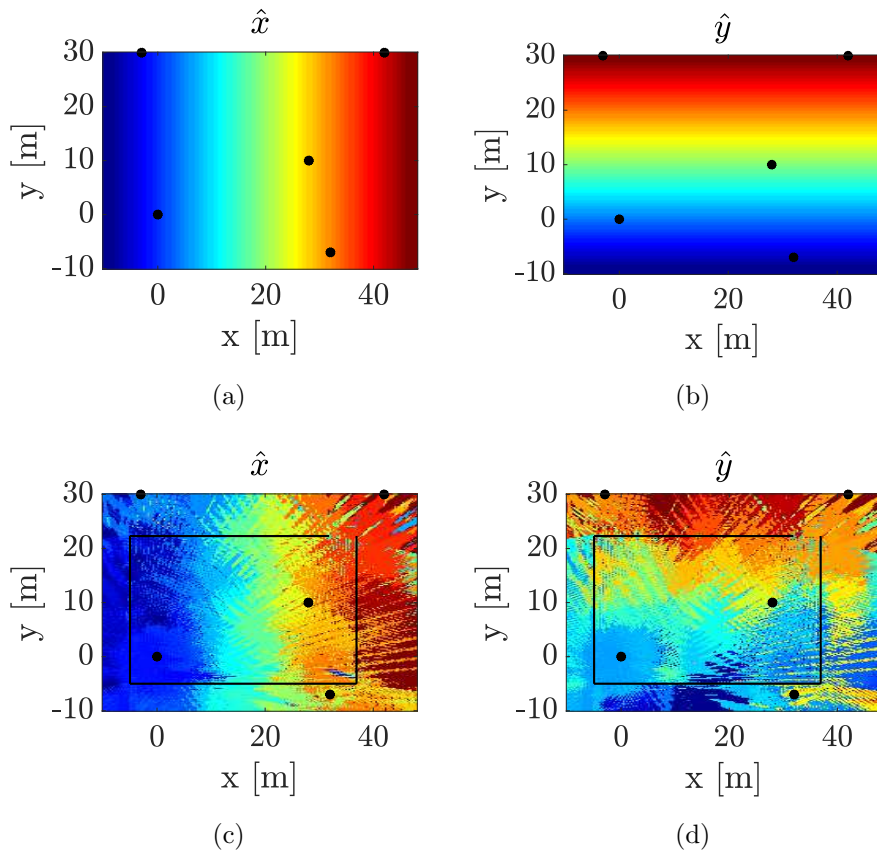


Figure B.1: Estimation of spatial coordinates using TDoA: (a)-(b) in free space, (c)-(d) indoor where the solid black lines represent the walls of the building; the black dots represent the locations of the anchor base stations. The color of each pixel represents the value of the estimated location coordinate at each point in the 150×150 grid area. Because location estimates in (a)-(b) coincide with the true locations, they can act as colorbars to the estimates in (c)-(d).

In LocB cartography [1–4, 8–12, 14, 15, 17–19], a fusion center is ideally given pairs $\{(\mathbf{x}_n, \tilde{p}_n)\}_{n=1}^N$, which include the exact sensor locations $\{\mathbf{x}_n\}_{n=1}^N$, and obtains a function estimate $\hat{p}(\mathbf{x})$ that provides the power of the C2M at any query location $\mathbf{x} \in \mathcal{X}$. With this function, a node at location \mathbf{x} can determine the power of the C2M if it knows \mathbf{x} . In practice, however, location is typically unknown and hence the sensor at the n -th measurement point must estimate \mathbf{x}_n by relying on pilot signals $\{y_{l,n}[k]\}_{l=1}^L$, where $y_{l,n}[k]$ denotes the k -th sample of the pilot signal transmitted by the l -th base station¹ and received at the n -th measurement point. For convenience, form the $L \times K$ matrix \mathbf{Y}_n whose (l, k) -th entry is $y_{l,n}[k]$. Note that these pilot signals are generally transmitted through a separate channel, not necessarily the C2M. However, both channels may coincide, as it occurs in certain cellular communication standards.

¹Although the discussion assumes for simplicity that the pilot signals are transmitted by terrestrial base stations, the proposed scheme can also be applied when these pilot signals are transmitted by satellites.

From \mathbf{Y}_n , the sensor at the n -th measurement point obtains the estimate $\hat{\mathbf{x}}_n := \hat{\mathbf{x}}(\mathbf{Y}_n)$ of \mathbf{x}_n by means of some localization algorithm [21, 22]. A fusion center then uses $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$ to obtain an estimate $\hat{p}(\mathbf{x})$ of the function $p(\mathbf{x})$. Therefore, if the location estimates $\{\hat{\mathbf{x}}_n\}_{n=1}^N$ are noisy, so will be $\hat{p}(\mathbf{x})$. If a node at an unknown query location wishes to determine the power of the C2M, it will use the pilot signals \mathbf{Y} to obtain an estimate $\hat{\mathbf{x}} := \hat{\mathbf{x}}(\mathbf{Y})$ of its location and will evaluate the map estimate as $\hat{p}(\hat{\mathbf{x}})$. In this case, \mathbf{Y} is a matrix whose (l, k) -th entry is given by the k -th sample of the l -th pilot signal $y_l[k]$ at the query location. Thus, such an estimation has two sources of error: first, the location estimation error in $\hat{\mathbf{x}}$ and, second, the map estimation error in $\hat{p}(\mathbf{x})$.

Remark 2 *One may argue that a node can determine the power of the C2M at its location more efficiently by measuring it rather than by receiving the pilot signals, applying a localization algorithm, and evaluating the map. Whereas this may be the case for a single C2M, if the aim is to determine the PSD, the power of many C2Ms, or the impulse response, then the associated measurement time may be prohibitive, which favors the adoption of spectrum cartography approaches.*

B.3 Location-Free Cartography

This section proposes LocF cartography, which bypasses the localization step involved in all existing cartography approaches. To this end, the LocF cartography problem is formulated as a function estimation task in Sec. B.3.1 and solved via kernel-based learning in Sec. B.3.2.

B.3.1 Map Estimate as a Function Composition

As detailed in the previous section, existing spectrum cartography techniques are heavily impaired by localization errors since the maps they construct are functions of noisy location estimates. The main idea of the proposed framework is to bypass such a dependence.

To this end, it is worth interpreting LocB cartography from a more abstract perspective. As detailed in Sec. B.2, the LocB map estimate is of the form $\hat{p}(\hat{\mathbf{x}})$ with $\hat{\mathbf{x}} := \hat{\mathbf{x}}(\mathbf{Y})$ denoting the output of the selected localization algorithm when the pilot signals are given by $\mathbf{Y} \in \mathcal{Y}$. Thus, this estimate can be seen as a function of \mathbf{Y} , i.e. $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) := \hat{p}(\hat{\mathbf{x}}(\mathbf{Y}))$, which can be expressed schematically as:

$$\begin{array}{ccccc} \mathcal{Y} & \xrightarrow{\hat{\mathbf{x}}} & \mathcal{X} & \xrightarrow{\hat{p}} & \mathbb{R} \\ \mathbf{Y} & \longrightarrow & \hat{\mathbf{x}}(\mathbf{Y}) & \longrightarrow & \hat{p}(\hat{\mathbf{x}}(\mathbf{Y})). \end{array} \quad (\text{B.1})$$

As mentioned in Sec. B.2, existing (LocB) cartography approaches obtain an estimate \hat{p} of p using the data $\{(\hat{\mathbf{x}}(\mathbf{Y}_n), \tilde{p}_n)\}_{n=1}^N$ for instance by searching for a function in an RKHS [4, 13, 14]. When $\hat{\mathbf{x}}(\mathbf{Y})$ is a reasonable estimate of the location \mathbf{x} at which \mathbf{Y} has been observed, such a LocB approach works well. However, due to multipath propagation effects impacting the pilot signals in \mathbf{Y} , $\hat{\mathbf{x}}(\mathbf{Y})$ may be very different from \mathbf{x} , which drastically hinders the estimation of p . Thus, in those cases where the location

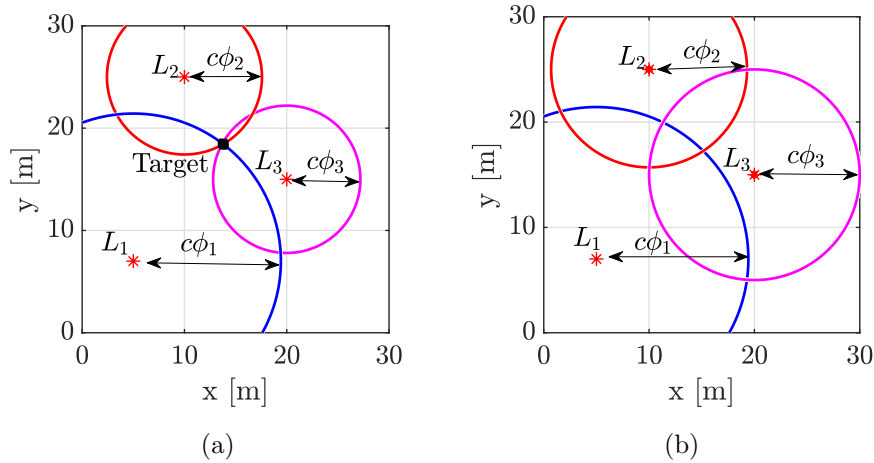


Figure B.2: Multi-lateration using ToA measurements with circles as possible sensor locations: (a) consistent ToA with the sought sensor location being the intersection of the circles (black square) and (b) inconsistent ToA measurements. The red stars represent the locations of the anchor base stations.

estimates $\{\hat{\mathbf{x}}(\mathbf{Y}_n)\}_{n=1}^N$ are noisy, the resulting estimate \hat{p} , and consequently $\hat{p}_{\mathbf{Y}}$, will be correspondingly noisy.

Since the source of such an error is the dependency of $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) = \hat{p}(\hat{\mathbf{x}}(\mathbf{Y}))$ on the estimated location $\hat{\mathbf{x}}(\mathbf{Y})$, one could think of bypassing this dependence by directly estimating $\hat{p}_{\mathbf{Y}}$ as a general function of \mathbf{Y} :

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\hat{p}_{\mathbf{Y}}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \hat{p}_{\mathbf{Y}}(\mathbf{Y}). \end{aligned} \tag{B.2}$$

When pursuing an estimate of this general form, $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$ would not be confined to depend on \mathbf{Y} only through the estimated location. However, finding such an estimate given $\{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$ by searching over a generic class of functions such as an RKHS would be extremely challenging due the so-called *curse of dimensionality* [31, 32]. To intuitively understand this phenomenon, note that the number of input variables of function $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$ is LK , typically in the order of hundreds or thousands. Since learning a multivariate function up to a reasonable accuracy generally requires that the number of data points be several orders of magnitude larger than the number of input variables, this approach would need N to be significantly larger than LK , and therefore prohibitively large.

To summarize, the structure imposed by (B.2) is too generic, whereas the one imposed by (B.1) is too restrictive. To attain a sweet spot in this trade-off, it is worth decomposing $\hat{\mathbf{x}}(\mathbf{Y})$ as detailed next. Recall that $\hat{\mathbf{x}}(\mathbf{Y})$ is the result of applying a localization algorithm to the pilot signals \mathbf{Y} . For most existing algorithms, $\hat{\mathbf{x}}(\mathbf{Y})$ can be thought of as the composition of two functions: a function $\phi : \mathcal{Y} \rightarrow \mathcal{F} \subset \mathbb{R}^M$ that obtains M features from \mathbf{Y} , such as T(D)oA or DoA, and a function $\hat{l} : \mathcal{F} \rightarrow \mathcal{X}$, that provides a location estimate

$\hat{\mathbf{l}}(\phi)$ given a feature vector $\phi \in \mathcal{F}$. In this case, $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$ can be decomposed as:

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\phi} \mathcal{F} && \xrightarrow{\hat{\mathbf{l}}} \mathcal{X} && \xrightarrow{\hat{p}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \phi(\mathbf{Y}) && \longrightarrow \hat{\mathbf{l}}(\phi(\mathbf{Y})) && \longrightarrow \hat{p}(\hat{\mathbf{l}}(\phi(\mathbf{Y}))). \end{aligned} \quad (\text{B.3})$$

Observe that the reason why the location estimate $\hat{\mathbf{x}}(\mathbf{Y}) = \hat{\mathbf{l}}(\phi(\mathbf{Y}))$ is inaccurate in multipath environments is because the algorithm that evaluates $\hat{\mathbf{l}}$ adopts a model where there is a certain “agreement” among features $\phi(\mathbf{Y})$. To see this, consider Fig. B.2, which illustrates the task of estimating the location of a sensor in an area with $L = 3$ base stations. The features in $\phi \in \mathbb{R}^M$, with $M = L = 3$, used in this example are noiseless ToA features. For each pilot signal, there is a circle centered at the base station and whose radius equals c times the ToA, where c is the speed of light. Thus, when there is no multipath, the ToA features are accurate and the sensor to be located must lie in the intersection of the three circles, as shown in Fig. B.2a. Thus, the localization algorithm (embodied in $\hat{\mathbf{l}}$) just needs to return the location at which these circles intersect. However, in multipath environments, the ToA features obtained from \mathbf{Y} do not generally equal the time it takes for an electromagnetic wave to propagate from the corresponding base station to the sensor. As a result, the aforementioned circles will not generally intersect; see Fig. B.2b. In other words, the expected agreement among features is absent and, hence, the localization algorithm will return an inaccurate estimate of the position.

In view of these arguments, the key idea in this paper is to pursue estimates $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$ of the form:

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\phi} \mathcal{F} && \xrightarrow{\hat{d}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \phi(\mathbf{Y}) && \longrightarrow \hat{d}(\phi(\mathbf{Y})). \end{aligned} \quad (\text{B.4})$$

In this setting, the problem is find an estimate $\hat{d}(\phi)$ given $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$, where $\phi_n := \phi(\mathbf{Y}_n)$. By following this approach, the estimated map $\hat{p}(\mathbf{Y}) = \hat{d}(\phi(\mathbf{Y}))$ does not involve a high number of inputs as in (B.2) and does not depend on the location estimate as in (B.1). For the latter reason, this approach will be referred to as *LocF cartography*. Since this approach does not need the agreement among entries of $\phi(\mathbf{Y})$ illustrated in Fig. B.2b, it is expected to outperform traditional spectrum cartography methods when such an agreement is not present, as occurs in multipath environments.

B.3.2 Kernel-based Power Map Learning

This section applies kernel-based learning to provide an algorithm capable of learning the function \hat{d} introduced in Sec. B.3.1.

Given pairs $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$, where $\phi_n := \phi(\mathbf{Y}_n)$, the problem can be informally stated as finding a function \hat{d} that satisfies two conditions: CO1) \hat{d} fits the data, that is, $\hat{d}(\phi_n) \approx \tilde{p}_n$, $n = 1, \dots, N$; and CO2) \hat{d} generalizes well to unseen data, i.e., if a new pair $(\phi_{N+1}, \tilde{p}_{N+1})$ is received, then $\hat{d}(\phi_{N+1}) \approx \tilde{p}_{N+1}$. A popular approach to solve the aforementioned function learning problem is kernel-based learning, mainly due to its simplicity, universality, and good performance [33]. Furthermore, multiple works have demonstrated the merits of this framework for spectrum cartography; see Sec. B.1.

The first step when attempting to learn a function is to specify in which family of functions \hat{d} must be sought. In kernel-based learning, one seeks \hat{d} in a set known as a *reproducing-kernel Hilbert space* (RKHS), which is given by:

$$\mathcal{H} := \left\{ d : d(\boldsymbol{\phi}) = \sum_{i=1}^{\infty} \alpha_i \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}'_i), \boldsymbol{\phi}'_i \in \mathcal{F}, \alpha_i \in \mathbb{R} \right\}, \quad (\text{B.5})$$

where $\kappa : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$ is a symmetric and positive definite function known as *reproducing kernel* [34]. Although kernel methods can use any reproducing kernel, a common choice is the so-called Gaussian *radial basis function* $\kappa(\boldsymbol{\phi}, \boldsymbol{\phi}') := \exp[-\|\boldsymbol{\phi} - \boldsymbol{\phi}'\|^2/(2\sigma^2)]$, where $\sigma > 0$ is a parameter selected by the user. As any Hilbert space, \mathcal{H} has an associated inner product and norm. For an RKHS function $d(\boldsymbol{\phi}) = \sum_{i=1}^{\infty} \alpha_i \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}'_i)$, the latter is given by:

$$\|d\|_{\mathcal{H}}^2 := \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_j \kappa(\boldsymbol{\phi}'_i, \boldsymbol{\phi}'_j). \quad (\text{B.6})$$

Kernel-based learning typically solves a problem of the form:

$$\hat{d} = \arg \min_{d \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\tilde{p}_n, \boldsymbol{\phi}_n, d(\boldsymbol{\phi}_n)) + \omega(\|d\|_{\mathcal{H}}), \quad (\text{B.7})$$

where \mathcal{L} is a loss function quantifying the deviation between the observations $\{\tilde{p}_n\}_{n=1}^N$ and the predictions $\{d(\boldsymbol{\phi}_n)\}_{n=1}^N$ returned by a candidate d ; and ω is an increasing function. The first term in (B.7) promotes function estimates satisfying CO1. The second term promotes estimates satisfying CO2 by limiting overfitting. Intuitively, $\|\cdot\|_{\mathcal{H}}$ captures a certain form of smoothness that limits the variability of d .

Although there exist different candidate functions for \mathcal{L} and ω in kernel-base learning, typical choices are $\mathcal{L}(\tilde{p}_n, \boldsymbol{\phi}_n, d(\boldsymbol{\phi}_n)) = (\tilde{p}_n - d(\boldsymbol{\phi}_n))^2$ and $\omega(\|d\|_{\mathcal{H}}) = \lambda \|d\|_{\mathcal{H}}^2$, where $\lambda > 0$ is a regularization parameter that balances smoothness and goodness of fit. For this choice, \hat{d} is termed *kernel ridge regression* estimate [33, Ch. 4], and is the one used in our experiments for simplicity. The goal is therefore to solve (B.7). However, since \mathcal{H} is generally infinite dimensional, (B.7) cannot be directly solved. Fortunately, one can invoke the *representer theorem* [34], which states that the solution to (B.7) is of the form:

$$\hat{d}(\boldsymbol{\phi}) = \sum_{n=1}^N \alpha_n \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}_n), \quad (\text{B.8})$$

for some $\{\alpha_n\}_{n=1}^N$. Although the representer theorem does not provide $\{\alpha_n\}_{n=1}^N$, these coefficients can be obtained by substituting (B.8) into (B.7) and solving the resulting problem with respect to them. Applying this procedure for kernel ridge regression results in the problem:

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \frac{1}{N} \|\tilde{\boldsymbol{p}} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}, \quad (\text{B.9})$$

where $\boldsymbol{\alpha} := [\alpha_1, \dots, \alpha_N]^\top$, $\tilde{\boldsymbol{p}} := [\tilde{p}_1, \dots, \tilde{p}_N]^\top$, and \mathbf{K} is a positive-definite $N \times N$ matrix whose (n, n') -th entry is $\kappa(\boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'})$. Problem (B.9) can be readily solved in closed-form

as $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}$. The estimate \hat{d} solving (B.7) for kernel ridge regression can be recovered by substituting the resulting $\{\alpha_n\}_{n=1}^N$ into (B.8). To obtain the predicted power of the C2M at a query location \mathbf{x} where the pilot signals are given by \mathbf{Y} , one just evaluates the LocF estimate $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) = \hat{d}(\boldsymbol{\phi}(\mathbf{Y}))$.

B.4 Location-Free Features

As described in Sec. B.3.1, LocB cartography algorithms learn a function of the location estimate. In the machine learning terminology, the *features* are the spatial coordinates of the sensor locations. On the other hand, the features used by LocF cartography are the entries of $\boldsymbol{\phi}(\cdot)$. In principle, $\boldsymbol{\phi}(\mathbf{Y})$ could be set to contain the same features as the ones used by $\hat{\mathbf{l}}(\cdot)$; see Sec. B.3. However, it is generally preferable to use features specifically tailored to LocF cartography. This section accomplishes the design of these features in several steps.

B.4.1 Feature Extraction

In Sec. B.3.1, $\boldsymbol{\phi}(\mathbf{Y})$ comprised M features used by typical localization algorithms, e.g. T(D)oA or DoA. The key observation is that, although these features are appropriate for localization, a different set of features may be preferable for LocF cartography. To come up with a natural feature design, this section first reviews the features used by typical localization algorithms (hence for LocB cartography) and analyzes their limitations. Inspired by this analysis, a novel feature extraction approach is proposed. To simplify the exposition, the scenario where sensors are synchronized with the base stations is presented first. A more practical setup, where this synchronization is not required, will be considered next.

B.4.1.1 Sensors are Synchronized with Base Stations

The received pilot signal is generally modeled as:

$$y_{l,n}[k] := a_l[k] * h_{l,n}[k] + w_{l,n}[k], \quad (\text{B.10})$$

where $a_l[k]$ is the k -th sample of the l -th transmitted pilot signal, $h_{l,n}[k]$ is the discrete-time channel impulse response between the l -th base station and the sensor at the n -th location, and $w_{l,n}[k]$ is the noise term. The discrete-time impulse response $h_{l,n}[k]$ is obtained next from its analog counterpart $h_{l,n}(t)$, which follows the conventional multipath channel model with $P_{l,n}$ components:

$$h_{l,n}(t) = \sum_{p=1}^{P_{l,n}} \alpha_{l,n}^{(p)} \delta(t - t_{l,n}^{(p)}), \quad (\text{B.11})$$

where $\delta(\cdot)$ is the Dirac delta distribution and $\alpha_{l,n}^{(p)} \in \mathbb{R}$ and $t_{l,n}^{(p)}$ are respectively the amplitude and delay of the p -th path. After up-conversion to the carrier frequency f_c , the pilot signal of the l -th base station is transmitted and received by the sensor at the

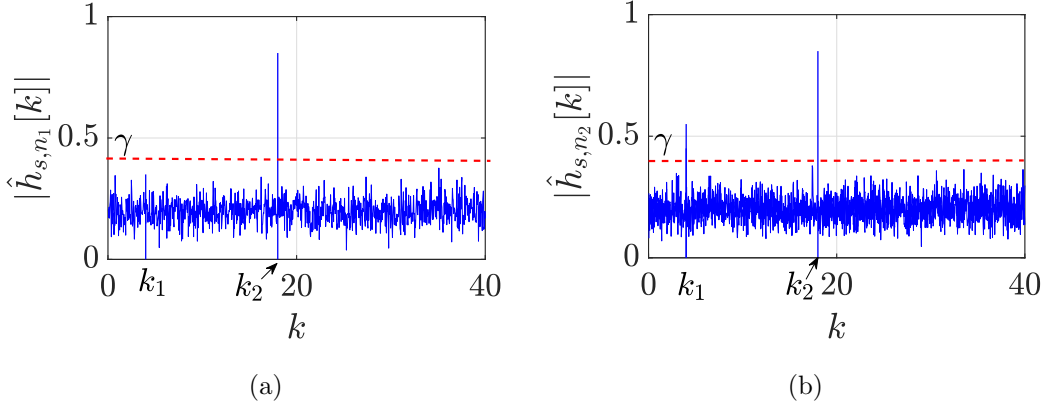


Figure B.3: Extraction of ToA from digital impulse response measured at two points that are spatially close. In (a), the ToA estimate is proportional to k_2 ; whereas in (b), the ToA estimate is proportional to k_1 .

n -th measurement point, which bandpass-filters with bandwidth B , down-converts, and samples at the Nyquist rate $T = 1/B$. Therefore, the received noiseless samples are given by $y_{l,n}[k]$ in (B.10), where [35, 36]:

$$h_{l,n}[k] = \sum_{p=1}^{P_{l,n}} \alpha_{l,n}^{(p)} e^{-j2\pi f_c t_{l,n}^{(p)}} \operatorname{sinc} \left(k - \frac{t_{l,n}^{(p)}}{T} \right). \quad (\text{B.12})$$

In view of these expressions, one of the most natural estimators for the ToA $\tau_{l,n} := t_{l,n}^{(1)}$ is:

$$\hat{\tau}_{l,n} := T \cdot \min\{k : |\hat{h}_{l,n}[k]| \geq \gamma\}, \quad (\text{B.13})$$

where $\hat{h}_{l,n}[k]$ is an estimate of $h_{l,n}[k]$ and γ is typically set as a function of the signal-to-noise ratio [26].

It will be argued next that such a ToA feature does not evolve smoothly over space in presence of multipath, and therefore, this may negatively impact estimation performance, as occurs with LocB cartography; see discussion about Fig. B.1 in Sec. B.1. For simplicity, assume that $a_l[k] = \delta[k]$, where $\delta[k]$ is the Kronecker delta. In this case, one can directly estimate $h_{l,n}[k]$ as $\hat{h}_{l,n}[k] = y_{l,n}[k] = h_{l,n}[k] + w_{l,n}[k]$, which is a noisy version of $h_{l,n}[k]$. To see the impact of multipath, consider a simple example where the measurement points \mathbf{x}_{n_1} and \mathbf{x}_{n_2} lie close to each other and the channel impulse responses are given by $\hat{h}_{l,n_1}[k] = \alpha_{l,n_1}^{(1)} \delta[k - k_{l,n_1}^{(1)}] + \alpha_{l,n_1}^{(2)} \delta[k - k_{l,n_1}^{(2)}] + w_{l,n_1}[k]$ and $\hat{h}_{l,n_2}[k] = \alpha_{l,n_2}^{(1)} \delta[k - k_{l,n_2}^{(1)}] + \alpha_{l,n_2}^{(2)} \delta[k - k_{l,n_2}^{(2)}] + w_{l,n_2}[k]$. Due to their spatial proximity, it follows that:

$$\alpha_{l,n_1}^{(1)} \approx \alpha_{l,n_2}^{(1)}, \quad \alpha_{l,n_1}^{(2)} \approx \alpha_{l,n_2}^{(2)}, \quad (\text{B.14a})$$

$$k_{l,n_1}^{(1)} \approx k_{l,n_2}^{(1)} \approx k_1, \quad k_{l,n_1}^{(2)} \approx k_{l,n_2}^{(2)} \approx k_2, \quad (\text{B.14b})$$

for some k_1 and k_2 . Assuming for simplicity that the effects of noise are negligible, if

$|\alpha_{l,n_1}^{(1)}| < \gamma < |\alpha_{l,n_2}^{(2)}|$ and $\gamma < |\alpha_{l,n_2}^{(1)}|$, then the ToA estimates are:

$$\begin{aligned}\hat{\tau}_{n_1} &:= T \cdot \min\{k : |\hat{h}_{l,n_1}[k]| \geq \gamma\} = Tk_{l,n_1}^{(2)} \approx Tk_2, \\ \hat{\tau}_{n_2} &:= T \cdot \min\{k : |\hat{h}_{l,n_2}[k]| \geq \gamma\} = Tk_{l,n_2}^{(1)} \approx Tk_1.\end{aligned}$$

This scenario is illustrated in Fig. B.3. Despite how close their locations and observed impulse responses are, the ToA estimates at locations \mathbf{x}_{n_1} and \mathbf{x}_{n_2} can be quite different, which establishes that the ToA estimate in (B.13) is not a smooth function of the spatial location.

Since this non-smoothness negatively affects the performance of the proposed LocF cartography estimator (and since the latter does not need ToA estimates that are proportional to the distance, as occurs in LocB cartography), a promising candidate for feature would be the *center of mass* (CoM) of the estimated impulse response:

$$\text{CoM}_{l,n} := \frac{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2 k}{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2},$$

where K is the number of samples. To see why such a feature evolves smoothly over space, suppose that the effects of noise are negligible and note that this CoM feature applied to the channel impulse responses in the previous example yields:

$$\begin{aligned}\text{CoM}_{l,n_1} &= \frac{k_{l,n_1}^{(1)} |\alpha_{l,n_1}^{(1)}|^2 + k_{l,n_1}^{(2)} |\alpha_{l,n_1}^{(2)}|^2}{|\alpha_{l,n_1}^{(1)}|^2 + |\alpha_{l,n_1}^{(2)}|^2}, \\ \text{CoM}_{l,n_2} &= \frac{k_{l,n_2}^{(1)} |\alpha_{l,n_2}^{(1)}|^2 + k_{l,n_2}^{(2)} |\alpha_{l,n_2}^{(2)}|^2}{|\alpha_{l,n_2}^{(1)}|^2 + |\alpha_{l,n_2}^{(2)}|^2}.\end{aligned}$$

From (B.14), it follows that $\text{CoM}_{l,n_1} \approx \text{CoM}_{l,n_2}$, which indicates that the CoM is indeed a feature that evolves smoothly over space, and therefore preferable for LocF cartography. In this case, the feature vector at the n -th sensor location becomes $\phi_n = [\text{CoM}_{1,n}, \dots, \text{CoM}_{L,n}]^\top$.

B.4.1.2 Sensors are not Synchronized with Base Stations

Since synchronization requires more expensive equipment and becomes challenging in multipath scenarios, TDoA estimates are generally preferred for localization. TDoA estimates are typically obtained by extracting the lag corresponding to the maximum cross-correlation of a pair of received pilot signals [37]. Assuming zero-mean, the cross-correlation between two pilot signals received by the sensor at the n -th location is defined as:

$$c_{l,l',n}[i] := \mathbb{E}\{y_{l,n}[k]y_{l',n}^*[k-i]\} \quad \text{with } l \neq l'. \quad (\text{B.15})$$

With $a_l[k] = a_{l'}[k]$ a white process with power σ_a^2 and uncorrelated with $w_{l,n}[k]$ and $w_{l',n}[k]$, also uncorrelated with each other, it can be easily seen that:

$$c_{l,l',n}[i] = \sigma_a^2 (h_{l,n}[i] * h_{l',n}^*[-i]).$$

A common estimate of the TDoA $\Delta_{l,l',n}$ is (see e.g. [37]):

$$\hat{\Delta}_{l,l',n} = T \cdot \arg \max_i \{ |\hat{c}_{l,l',n}[i]| \}, \quad (\text{B.16})$$

where $\hat{c}_{l,l',n}[i]$ is an estimate of $c_{l,l',n}[i]$. To see the intuition behind this estimator, note that $\hat{h}_{l,n}[k] = \alpha_{l,n}^{(1)} \delta[k - k_{l,n}^{(1)}]$ and $\hat{h}_{l',n}[k] = \alpha_{l',n}^{(1)} \delta[k - k_{l',n}^{(1)}]$ in a free-space channel with large bandwidth B . This implies that:

$$\begin{aligned} c_{l,l',n}[i] &= \sigma_a^2 \alpha_{l,n}^{(1)} \left(\alpha_{l',n}^{(1)} \right)^* \delta \left[i - \left(k_{l,n}^{(1)} - k_{l',n}^{(1)} \right) \right] \\ &= \sigma_a^2 \alpha_{l,n}^{(1)} \left(\alpha_{l',n}^{(1)} \right)^* \delta \left[i - \Delta_{l,l',n}/T \right], \end{aligned}$$

and therefore the lag of the maximum magnitude of $c_{l,l',n}[i]$ provides the TDoA in this simple scenario.

Similar arguments to those used in Sec. B.4.1.1 to conclude that the ToA estimates are not spatially smooth can also be invoked to reach the same conclusion for TDoA. Likewise, following the same rationale as in Sec. B.4.1.1, this section proposes alleviating the aforementioned issue by adopting features of the form:

$$\text{CoM}_{l,l',n} := \frac{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2 i}{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2}, \quad (\text{B.17})$$

where $\text{CoM}_{l,l',n}$ is the CoM of the cross-correlation between the l -th and l' -th pilot signals. The proposed feature has three advantages: i) it is smooth, as portrayed later in Sec. B.5.1, ii) it does not require synchronization between the localization base stations and the sensors, and iii) it does not require the knowledge of the impulse responses. With this choice, the feature vector at the n -th measurement location becomes:

$$\begin{aligned} \phi_n &= [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \\ &\quad \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^\top. \end{aligned} \quad (\text{B.18})$$

B.4.2 Cartography from a Reduced Set of Features

As argued earlier in Sec. B.3.1, learning becomes difficult when the number of input features M is high. This section develops a scheme to reduce this number of features to improve estimation performance in LocF cartography.

As stated in the previous section, in LocB cartography, the feature vectors correspond to the coordinates of the estimated location. Application of the localization algorithm represented by the function $\hat{\mathbf{l}}$ in (B.3) naturally reduces dimensionality from the original M features to just 2 or 3. On the other hand, in the case of LocF cartography, a larger number N of measurements to learn \hat{d} in (B.4) may be necessary to attain a target accuracy if M is large. This observation calls for a dimensionality reduction step that condenses the information of the feature vectors $\{\phi_n\}_{n=1}^N \subset \mathbb{R}^M$ into vectors $\{\bar{\phi}_n\}_{n=1}^N \subset \mathbb{R}^r$ of a reduced size r . Intuitively, r should be the minimum number that preserves most information while eliminating most of the noise in $\{\phi_n\}_{n=1}^N$. Even if some information is lost, the reduction in the error entailed by the fact that the function to be estimated has fewer input arguments may pay off in practice.

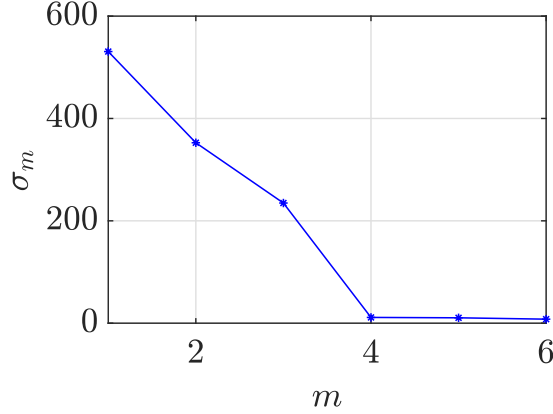


Figure B.4: Singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of Φ in non-increasing order for a multipath environment with $L = 4$ transmitters.

In the cases where the feature vectors $\{\phi_n\}_{n=1}^N$ lie close to a low-dimensional subspace, the coordinates of these vectors with respect to a basis for such a subspace may constitute a suitable reduced set of features. To see this, it is instructive to start by considering the scenario of TDoA features. Suppose, for simplicity, that the effects of noise are negligible, so that the TDoA estimates $\hat{\Delta}_{l,l',n}$ approximately equal the true TDoAs $\Delta_{l,l',n}$. Then, the rows of $\Phi := [\phi_1, \dots, \phi_N]$ are of the form $\Delta_{l,l'} := [\Delta_{l,l',1}, \Delta_{l,l',2}, \dots, \Delta_{l,l',N}]^\top$. If $\tau_l := [\tau_{l,1}, \dots, \tau_{l,N}]^\top$ collects the ToA from the l -th base station to all sensor locations, then it clearly holds that $\Delta_{l,l'} = \tau_l - \tau_{l'}$. Consequently, $\Delta_{1,l} - \Delta_{1,l'} = \tau_1 - \tau_l - (\tau_1 - \tau_{l'}) = \tau_{l'} - \tau_l = \Delta_{l',l}$, which implies that all rows of Φ are linear combinations of the $L - 1$ rows $\{\Delta_{1,l}\}_{l=2}^L$. Thus, the rank of Φ is at most $L - 1$ or, equivalently, the vectors $\{\phi_n\}_{n=1}^N$ lie in a subspace of dimension $L - 1$. When effects of noise are noticeable, one would expect that the vectors $\{\phi_n\}_{n=1}^N$ lie *close* to a subspace of dimension $L - 1$.

Similarly, one can expect that when the entries of the vectors $\{\phi_n\}_{n=1}^N$ are given by (B.17), these vectors also lie close to a low-dimensional subspace since CoM features are proportional to the TDoAs in absence of multipath; see Sec. B.4.1. This phenomenon can be illustrated through simulation (see Sec. B.5 for more details). Fig. B.4 depicts the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$ of Φ in non-increasing order for a multipath environment described in Sec. B.5 with $L = 4$. As expected, roughly $r = L - 1 = 3$ directions capture almost all the energy of the rows of Φ .

When a set of random vectors lie close to a subspace, an appealing approach for dimensionality reduction is principal component analysis (PCA) [31, Ch. 12], which obtains the reduced feature vectors by projecting the input data vectors onto the subspace that preserves most of the energy. Since in this paper no probabilistic assumptions have been introduced on $\{\phi_n\}_{n=1}^N$, the typical formulation of PCA is not directly applicable. However, as detailed next, it is not difficult to extend this idea to the fully deterministic scenario, which furthermore provides intuition.

Assume w.l.o.g. a centered set of feature vectors, i.e., $(1/N) \sum_{n=1}^N \phi_n = \mathbf{0}$. If not centered, just subtract the mean by replacing Φ with $\Phi - (1/N)\Phi\mathbf{1}\mathbf{1}^\top$. The subspace that captures most of the energy of the observations can be determined using the singular

value decomposition (SVD) of Φ , which for $M < N$ is given by:

$$\Phi = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\top \\ \mathbf{V}_2^\top \end{bmatrix}, \quad (\text{B.19})$$

where $\Sigma_1 := \text{diag}\{\sigma_1, \dots, \sigma_r\}$ contains the r largest singular values of Φ , $\Sigma_2 := \text{diag}\{\sigma_{r+1}, \dots, \sigma_M\}$ contains the $M - r$ smallest, and the columns of $\mathbf{U} := [\mathbf{U}_1 \quad \mathbf{U}_2]$ (respectively $\mathbf{V} := [\mathbf{V}_1, \mathbf{V}_2]$) are the left (right) singular vectors of Φ . Clearly, if the data vectors $\{\phi_n\}_{n=1}^N$ are multiplied by the orthogonal matrix \mathbf{U}^\top , the resulting vectors $\{\phi'_n\}_{n=1}^N$, with $\phi'_n := \mathbf{U}^\top \phi_n$, contain the same information. Thus, one can replace Φ with $\Phi' := \mathbf{U}^\top \Phi$.

By applying this transformation, which can be thought of as a generalized rotation, most of the energy of Φ' is concentrated in its first r rows. To see this, note that the energy of the first r rows of Φ' is given by:

$$\begin{aligned} \|\mathbf{U}_1^\top \Phi\|_F^2 &= \|\Sigma_1 \mathbf{V}_1^\top\|_F^2 = \text{Tr}(\Sigma_1 \mathbf{V}_1^\top \mathbf{V}_1 \Sigma_1^\top) \\ &= \text{Tr}(\Sigma_1 \Sigma_1^\top) = \|\Sigma_1\|_F^2 = \sum_{m=1}^r \sigma_m^2, \end{aligned}$$

whereas the energy of the last $M - r$ rows of Φ' is given by:

$$\|\mathbf{U}_2^\top \Phi\|_F^2 = \|\Sigma_2\|_F^2 = \sum_{m=r+1}^M \sigma_m^2.$$

When $r = L - 1$, since the rows of Φ lie approximately in a subspace of dimension r , it follows that $\sigma_m \approx 0$ for $m > r$. Therefore $\sum_{m=1}^r \sigma_m^2 \gg \sum_{m=r+1}^M \sigma_m^2$ and, hence, $\|\mathbf{U}_1^\top \Phi\|_F^2 \gg \|\mathbf{U}_2^\top \Phi\|_F^2$. Equivalently, most of the energy of the vectors $\{\phi'_n\}_{n=1}^N$ is concentrated in their first r entries. This observation suggests using the first r entries of the vectors $\{\phi'_n\}_{n=1}^N$ as features, while discarding the rest. That is, the reduced dimensionality feature vectors will be given by $\{\bar{\phi}_n\}_{n=1}^N$, where $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n$. Note that $\bar{\phi}_n$ is just the vector of coordinates of ϕ_n with respect to the basis composed of the columns of \mathbf{U}_1 .

The number r of entries of the new feature vectors $\{\bar{\phi}_n\}_{n=1}^N$ may be potentially much smaller than M and can therefore boost estimation performance meaningfully. For instance, when $\{\phi_n\}_{n=1}^N$ are given by (B.18), this reduction is from $M = L(L - 1)/2$ features to $r = L - 1$ features.

In scenarios of very strong multipath, the rows of Φ may not lie close to any subspace of dimension $L - 1$. In those cases, it may be worth choosing a value of r greater than $L - 1$. A possibility is to specify a fraction $\eta \in [0, 1]$ of the energy of Φ that must be kept in $\bar{\Phi} := \mathbf{U}_1^\top \Phi$, and choose r to be the smallest integer that guarantees this condition, that is:

$$r = \min \left\{ r' : \frac{\sum_{\bar{m}=1}^{r'} \sigma_{\bar{m}}^2}{\sum_{m=1}^M \sigma_m^2} \geq \eta \right\}. \quad (\text{B.20})$$

To summarize, the problem of LocF cartography with the technique for reducing the set of features introduced in this section is as follows. Given the original set of measurements $\{\phi_n\}_{n=1}^N \subset \mathbb{R}^M$, one must form the matrix Φ , compute \mathbf{U}_1 from the SVD in (B.19), and

obtain the reduced features $\{\bar{\phi}_n\}_{n=1}^N \subset \mathbb{R}^r$ where $\bar{\phi}_n = \mathbf{U}_1^\top \phi_n$. Then, the function \hat{d} is obtained from the pairs $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$ using the approach in Sec. B.3.2. To evaluate the resulting map at a query location where the received pilot signals are given by \mathbf{Y} , one must simply obtain $\hat{d}(\mathbf{U}_1^\top \phi(\mathbf{Y}))$.

B.4.3 Dealing with Missing Features

Due to propagation effects, the signal-to-noise ratio of some of the received pilot signals may be too low for feature extraction. In this case, the features associated with those pilot signals may be unreliable or simply unavailable. This section develops techniques to cope with such missing features.

Let $\Omega \subset \{1, \dots, M\} \times \{1, \dots, N\}$ be such that $(m, n) \in \Omega$ iff the m -th feature is available at the n -th measurement location and define the ‘‘incomplete’’ feature matrix $\check{\Phi} \in (\mathbb{R} \cup \{\text{FiM}\})^{M \times N}$ as:

$$(\check{\Phi})_{m,n} = \begin{cases} (\phi_n)_m + \varsigma_{m,n} & \text{if } (m, n) \in \Omega \\ \text{FiM} & \text{otherwise,} \end{cases} \quad (\text{B.21})$$

where $\varsigma_{m,n}$ explicitly models error in the feature extraction and the symbol FiM represents that the corresponding feature is missing. Since the matrix $\check{\Phi}$ contains missing features, the LocF cartography scheme presented so far is not directly applicable. The missing features must be filled first. Hence, the goal is, given $\check{\Phi}$, find $\Phi \in \mathbb{R}^{M \times N}$ that agrees with $\check{\Phi}$ on Ω . A popular approach to address such a matrix completion task is via rank minimization [38]:

$$\begin{aligned} & \underset{\Phi}{\text{minimize}} \quad \text{rank}(\Phi) \\ & \text{subject to} \quad \mathcal{P}_\Omega(\Phi) = \mathcal{P}_\Omega(\check{\Phi}), \end{aligned} \quad (\text{B.22})$$

where

$$\begin{aligned} \mathcal{P}_\Omega : (\mathbb{R} \cup \{\text{FiM}\})^{M \times N} &\longrightarrow \mathbb{R}^{M \times N} \\ \check{\Phi} &\longmapsto \mathcal{P}_\Omega(\check{\Phi}), \end{aligned}$$

with

$$\left(\mathcal{P}_\Omega(\check{\Phi})\right)_{m,n} = \begin{cases} (\check{\Phi})_{m,n} & \text{if } (m, n) \in \Omega \\ 0 & \text{if } (m, n) \notin \Omega. \end{cases}$$

Although this problem is non-convex, efficient solvers exist based on convex relaxation [39, 40]. A legitimate question would be what is the minimum number of available features required to recover a reasonable reconstruction of Φ . As a guideline, a result in [41] establishes that, under certain conditions, the minimum number of available features to recover $\Phi \in \mathbb{R}^{M \times N}$ is $\mathcal{O}\left(\tilde{N} \text{rank}(\Phi) \log(\tilde{N})\right)$ where $\tilde{N} = \max(M, N)$.

Although the aforementioned rank minimization approach could, in principle, be used, it suffers from two limitations. First, it does not exploit the prior information that Φ can be well approximated by a matrix of rank r , where r is typically $L - 1$; see Fig. B.4.

Second, the constraint in (B.22) would render the reconstructed matrix sensitive to the noise $\{\varsigma_{m,n}\}_{m,n}$ present in $\check{\Phi}$. Thus, an appealing alternative to (B.22) would be:

$$\begin{aligned} \mathring{\Phi} := \underset{\Phi}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathcal{P}_\Omega(\Phi) - \mathcal{P}_\Omega(\check{\Phi})\|_F^2 \\ \text{subject to} \quad & \Phi \in \mathcal{M}_r, \end{aligned} \quad (\text{B.23})$$

where $\mathcal{M}_r := \{\Phi \in \mathbb{R}^{M \times N} : \operatorname{rank}(\Phi) = r\}$ is the smooth manifold of r -rank $M \times N$ matrices.

There exist algorithms to find local minima of the non-convex problem (B.23). One example based on manifold optimization [42] is the linear retraction-based geometric conjugate gradient (LRGeomCG) method from [43]. A less computationally expensive alternative is the singular value projection (SVP) method in [44], which is based on the traditional projected subgradient descent method.

After solving (B.23), all the columns of $\mathring{\Phi} := [\mathring{\phi}_1, \dots, \mathring{\phi}_N]$ clearly lie in a subspace of dimension r . From the arguments in Sec. B.4.2, learning the map can be improved by suppressing this redundancy. To this end, one could use the technique in Sec. B.4.2, which would obtain the reduced-dimensionality feature vectors as follows:

$$\bar{\Phi} := [\bar{\phi}_1, \dots, \bar{\phi}_N] = \mathring{U}_1^\top \mathring{\Phi}. \quad (\text{B.24})$$

Here, the columns of \mathring{U}_1 are the left singular vectors corresponding to the r largest singular values of $\mathring{\Phi}$. Nevertheless, since $\mathring{\Phi}$ has rank r , it is not necessary to obtain \mathring{U}_1 by means of an SVD. Namely, the columns of \mathring{U}_1 can be directly obtained by orthonormalizing the first r linearly independent columns of $\mathring{\Phi}$, e.g. through Gram-Schmidt.

To sum up, to estimate a map using the proposed LocF cartography in presence of missing features is as follows. First, matrix $\check{\Phi}$ is formed with the available features. Then, the completed matrix $\mathring{\Phi}$ is obtained using LRGeomCG or SVP. Next, \mathring{U}_1 is obtained through Gram-Schmidt over this completed matrix. Finally, one learns \hat{d} from $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$, where $\bar{\phi}_n$ is the n -th column of $\bar{\Phi}$ in (B.24), using the approach in Sec. B.3.2.

To evaluate the estimated map at a test location, one would require in principle the feature vector $\phi \in \mathbb{R}^M$ at that location or, alternatively, its reduced-dimensionality version $\bar{\phi} \in \mathbb{R}^r$. However, due to the phenomena described earlier, only some of the features of ϕ may be available, which can be collected in the vector $\check{\phi} \in (\mathbb{R} \cup \{\text{FiM}\})^M$. The problem now is to find the reduced-dimensionality feature vector $\bar{\phi}$ given $\check{\phi}$.

Since the columns of $\mathring{\Phi}$ lie in an r -dimensional subspace for which the columns of \mathring{U}_1 form an orthonormal basis, it is reasonable to say that the feature vector at the testing point $\phi \in \mathbb{R}^M$ also lies in that subspace, meaning that this vector can be written as $\phi = \mathring{U}_1 \bar{\phi}$ for some $\bar{\phi}$. The procedure to recover $\bar{\phi}$ depends on whether $\check{\phi}$ contains enough observed features. Let $\Omega' \subset \{1, \dots, M\}$ be such that the $m \in \Omega'$ iff the m -th feature is available in $\check{\phi}$. If $M := |\Omega'| \geq r$, one can think of finding $\bar{\phi}$ using the well-known regularized least squares (RLS) method as:

$$\begin{aligned} \hat{\phi} = \arg \min_{\bar{\phi}} \quad & \left\| \mathcal{P}_{\Omega'}(\check{\phi}) - \mathcal{P}_{\Omega'}(\mathring{U}_1 \bar{\phi}) \right\|^2 \\ & + \mu (\bar{\phi} - \bar{\phi}_{\text{avg}})^\top \mathbf{C}^{-1} (\bar{\phi} - \bar{\phi}_{\text{avg}}), \end{aligned} \quad (\text{B.25})$$

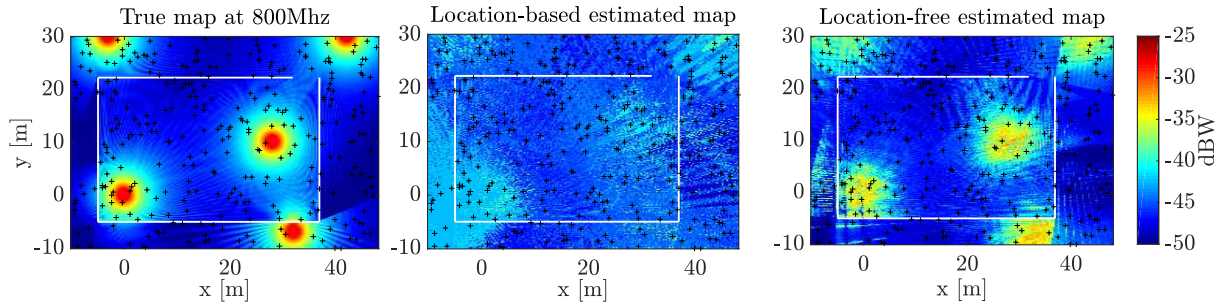


Figure B.5: (left) True map, (middle) LocB ($\lambda' = 3.3 \times 10^{-3}$, $\sigma' = 0.5$ m), and (right) LocF ($\lambda = 1.9 \times 10^{-4}$, $\sigma = 37$ m) estimated maps; $N = 300$, $L = 5$, $B = 20$ MHz, and $K = 10$. The black crosses indicate the sensor locations and the solid white lines represent the walls of the building.

where

$$\mathcal{P}_{\Omega'} : (\mathbb{R} \cup \{\text{FiM}\})^M \longrightarrow \mathbb{R}^M$$

$$\check{\phi} \longmapsto \phi, \quad (\phi)_m = \begin{cases} (\check{\phi})_m & \text{if } m \in \Omega' \\ 0 & \text{if } m \notin \Omega', \end{cases}$$

$\mu > 0$ is a regularization parameter, $\bar{\phi}_{\text{avg}}$ and $\mathbf{C} \in \mathbb{R}^{r \times r}$ are respectively the sample mean vector and covariance matrix of the coordinates of the completed features in the training phase, that is, $\bar{\phi}_{\text{avg}} = (1/N)\bar{\Phi}\mathbf{1}$ and $\mathbf{C} = (1/N)(\bar{\Phi} - \bar{\phi}_{\text{avg}}\mathbf{1}^\top)(\bar{\Phi} - \bar{\phi}_{\text{avg}}\mathbf{1}^\top)^\top$. To solve Problem (B.25), let the elements of Ω' be denoted as $\Omega' := \{m_1, \dots, m_{\check{M}}\}$. Then:

$$\hat{\phi} = \left(\mathring{\mathbf{U}}_1^\top \mathbf{S}^\top \mathbf{S} \mathring{\mathbf{U}}_1 + \mu \mathbf{C}^{-1} \right)^{-1} \left(\mathring{\mathbf{U}}_1^\top \mathbf{S}^\top \mathbf{S} \mathcal{P}_{\Omega'}(\check{\phi}) + \mu \mathbf{C}^{-1} \bar{\phi}_{\text{avg}} \right), \quad (\text{B.26})$$

where $\mathbf{S} \in \{0, 1\}^{\check{M} \times M}$ is a row selection matrix with all entries equal to zero except for the entries $(1, m_1), \dots, (\check{M}, m_{\check{M}})$, which equal to 1. Thus, $\mathbf{S} \mathcal{P}_{\Omega'}(\check{\phi}) = \mathbf{S} \mathring{\mathbf{U}}_1 \bar{\phi}$. On the other hand, if $\check{M} := |\Omega'| < r$, it is not possible to identify $\bar{\phi}$ from $\check{\phi}$. The extreme case would be when $\check{M} = 0$. A natural estimate at such point can be the spatial average of the signal power $(1/N) \sum_n \tilde{p}_n$.

B.5 Numerical tests

This section evaluates the performance of LocF cartography in presence of multipath, where localization algorithms cannot achieve accurate location estimates. To this end, the simulations are carried out in a 42×27 m structure comprising several parallel vertical planes modeling the external and internal walls of a building, the latter is located in a 60×40 m rectangular area \mathcal{X} .

All the experiments described in this paper can be reproduced with the MATLAB code which is available at the first author's homepage.

The simulation area contains L active transmitters. Some of these are positioned inside the building, others outside. Matrix $\mathbf{Y}_n \in \mathbb{C}^{L \times K}$ containing the noisy received

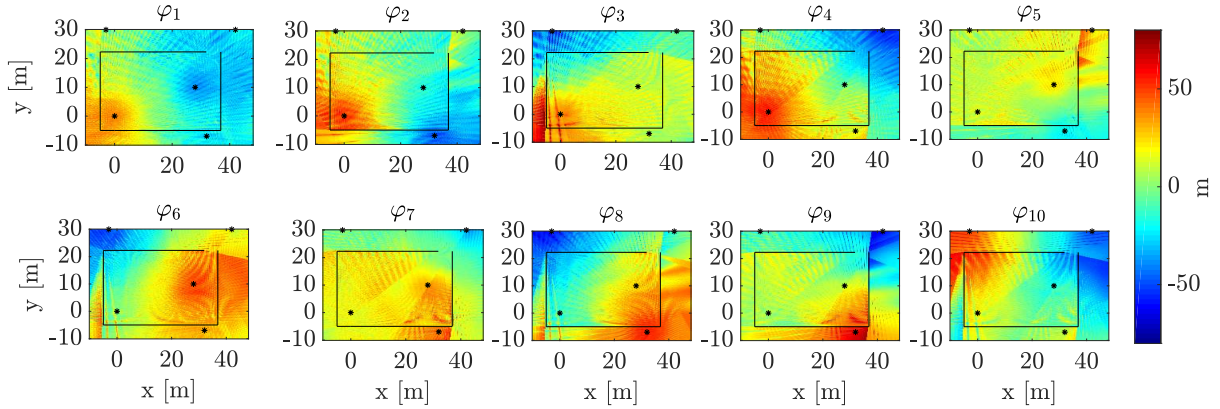


Figure B.6: Maps of the $M = 10$ LocF features with $L = 5$, $B = 20$ MHz, and $K = 10$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations.

Table B.1: Parameters used for the experiment in Fig. B.8.

	B (MHz)	50	100	200	700
	K	25	50	100	350
LocB	σ' (m)	10.1	8.9	9	7
	λ'	1.8×10^{-3}	9.1×10^{-4}	7.1×10^{-4}	2.1×10^{-4}
LocF	σ (m)	27	41	53	28
	λ	3.81×10^{-4}	6.1×10^{-5}	1.1×10^{-5}	5×10^{-4}

pilot signals is generated according to (B.10), where K is adjusted depending on B to capture all the multipath components. For simplicity, the pilot signals are given by² $a_l[k] = \delta[k]$ which implies that the rows of $\mathbf{Y}_n \in \mathbb{C}^{L \times K}$ contain the impulse responses of the bandlimited channels between the L transmitters and the n -th measurement location. The channel $h_{l,n}[k]$ is generated following (B.12) with a carrier frequency of 800 MHz and pilot channel bandwidth $B = 1/T$. The noise samples $w_{l,n}[k]$ are independent normal random variables with zero-mean and variance -70 dBm. Propagation adheres to the *Motley-Keenan multi-wall radio propagation model* [45], which accounts for the direct path, up to 5 first-order wall reflections, and up to 5 wall-to-wall second-order reflections. Remarkably, the model captures the impact of the angle of incidence on the power of the reflected ray. For simplicity, the C2M is chosen to be the channel where localization pilot signals are transmitted. In practice, this is the case in the downlink of a cellular communication system such as LTE where the base stations transmit both communication signals and localization pilots.

To ensure that the measurements are obtained in the far-field propagation region, sensor locations are spread uniformly at random over $\bar{\mathcal{X}}$, which comprises those points in \mathcal{X} lying at least 3 wavelengths away from all transmitters. Note that, although the number of sensor locations is sometimes in the order of hundreds, this does not mean that a large

²Amplitude units are such that a signal $x[k] = 1, \forall k$, has power 1 W.

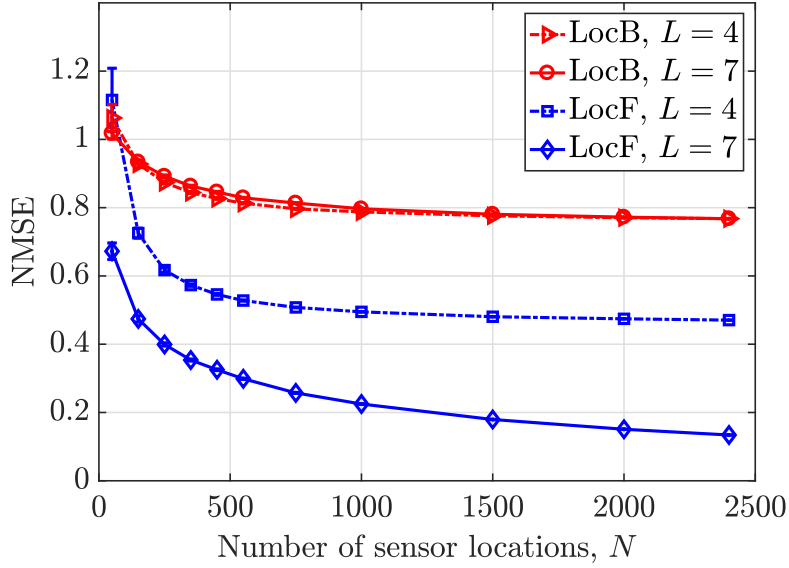


Figure B.7: Performance comparison between the LocF cartography ($\lambda = 1.9 \times 10^{-4}$, $\sigma = 37$ m) and the LocB cartography ($\lambda' = 3.3 \times 10^{-3}$, $\sigma' = 0.5$ m) with $B = 20$ MHz and $K = 10$.

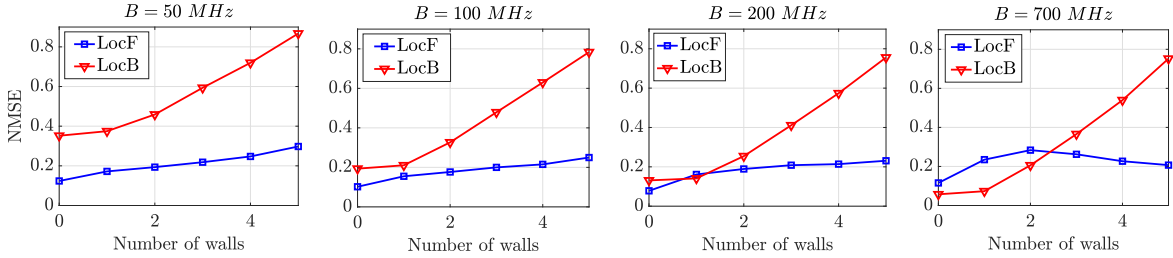


Figure B.8: LocF and LocB map NMSE as a function of the number of walls for different values of the bandwidth, $L = 5$, $N = 300$.

number of sensing devices must be used since each device may gather measurements at tens or hundreds of spatial locations. The power measurement p_n (measured in dBW) of the C2M at position \mathbf{x}_n is corrupted by additive noise ϵ_n to yield $\tilde{p}_n = p_n + \epsilon_n$, where $\{\epsilon_n\}_{n=1}^N$ are independent normal random variables with zero-mean and variance σ_ϵ^2 . This variance is such that the signal-to-noise ratio defined as $10 \log_{10}(\bar{p}^2/\sigma_\epsilon^2) \approx 40$ dB, where $\bar{p} := \int_{\bar{\mathcal{X}}} p(\mathbf{x}) d\mathbf{x} / \int_{\bar{\mathcal{X}}} d\mathbf{x}$ is the spatial average of $p(\mathbf{x})$. This SNR is considered practical since the measurement noise power σ_ϵ^2 can be driven arbitrarily close to zero in practice by averaging over a sufficiently long time window.

Quantitative evaluation will compare the normalized mean square error (NMSE) defined as:

$$\text{NMSE} = \frac{\mathbb{E}\{|p(\mathbf{x}) - \hat{p}_{\mathbf{Y}}(\mathbf{Y}(\mathbf{x}), \mathcal{T})|^2\}}{\mathbb{E}\{|p(\mathbf{x}) - \bar{p}|^2\}}, \quad (\text{B.27})$$

where $\hat{p}_{\mathbf{Y}}(\mathbf{Y}(\mathbf{x}), \mathcal{T})$ (measured in dBW) denotes the result of evaluating the map constructed from the training set $\mathcal{T} := \{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$ at the location \mathbf{x} , where $\mathbf{Y}(\mathbf{x})$ comprises

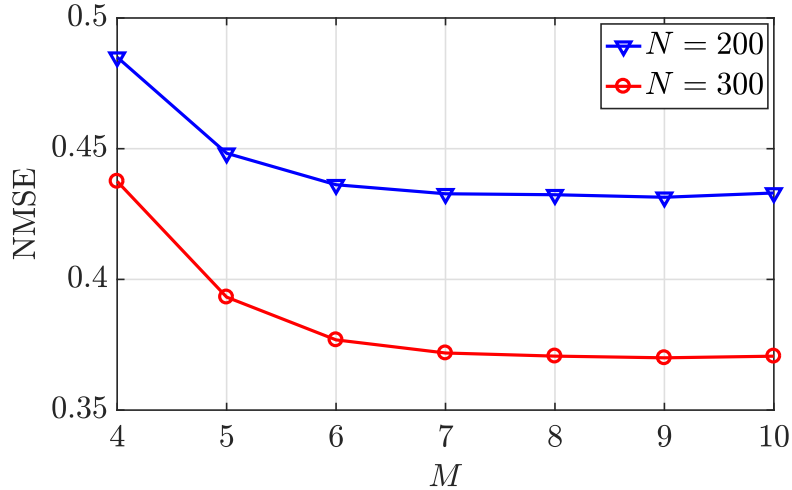


Figure B.9: LocF estimated map NMSE for different values of number of features M and sensor locations N , with $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.9 \times 10^{-4}$, and $\sigma = 37$ m.

the received pilot signals at \mathbf{x} . The denominator in (B.27) normalizes the square error of the considered algorithm by the error incurred by the best data-agnostic estimator, which estimates the spatial average \bar{p} at all points. Thus, the adopted performance metric is higher than traditional NMSE, meaning that it is more challenging to obtain lower values. Furthermore $\mathbb{E}\{\cdot\}$ denotes the expectation over the sensor locations and noise.

B.5.1 LocF vs. LocB

To avoid the need for synchronization between transmitters and sensors, the LocF algorithm utilizes the features in (B.17), which additionally provide robustness to multipath and evolve smoothly over space; see Sec. B.4.1. Since this center of mass can be thought of as a lag, it is scaled by the sampling period T and speed of light c to obtain the corresponding range difference, i.e.:

$$\phi_n := Tc [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^\top. \quad (\text{B.28})$$

Using these features, the LocF algorithm uses the kernel ridge regression technique in Sec. B.3.2 with Gaussian radial basis functions with parameter σ . The reason is that this *universal kernel* is capable of approximating arbitrary continuous functions that vanish at infinity [46]. On the other hand, for LocB cartography, the feature vector $\phi_n = \hat{\mathbf{x}}_n \in \mathbb{R}^2$ comprises estimates of the spatial coordinates of the n -th sensor location obtained by the *iterative re-weighting squared range difference-least squares* (IRWSRD-LS) algorithm [47], which features state-of-the-art localization performance. This algorithm is applied over TDoA features extracted from $\{\mathbf{Y}_n\}_{n=1}^N$ through (B.16). At the n -th sensor location, these features $\{\hat{\Delta}_{1,l',n}\}_{l'=2}^L$ comprise the TDoA between a reference base station and the remaining $L-1$ base stations. Enlarging this set by including TDoA measurements $\hat{\Delta}_{l,l',n}$ with $l \neq 1$ would not be beneficial for the estimation performance as discussed in [48].

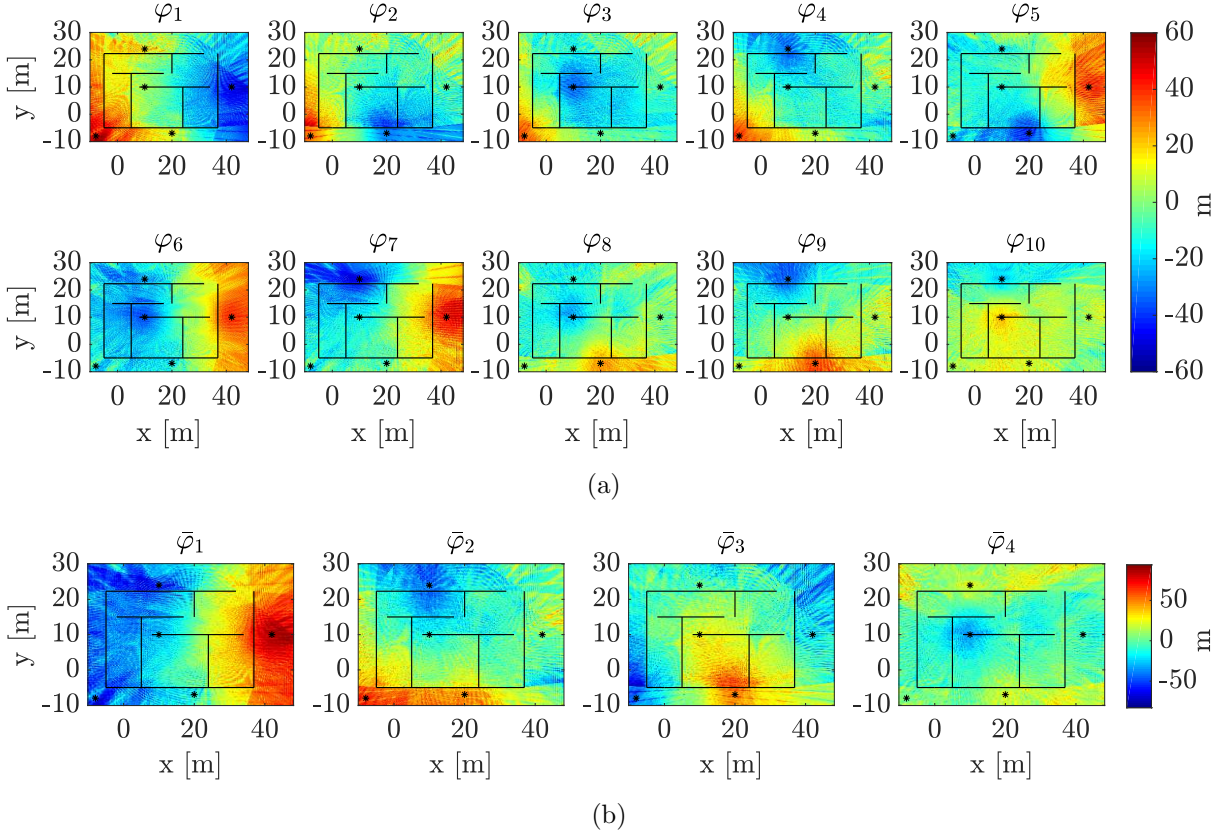


Figure B.10: Maps of: (a) all the $M = 10$ features and (b) $r = 4$ reduced features with $L = 5$, $B = 20$ MHz, and $K = 10$. The solid black lines represent the walls of the building and the black stars represent the transmitter locations. The maps are obtained by representing the value of the feature at every location in \mathcal{X} .

The reason is the redundancy inherent to TDoA features described in Sec. B.4.2. To ensure a fair comparison, LocB utilizes the same function learning algorithm as LocF; see Sec. B.3.2. Specifically, given $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$, the map is estimated as $\hat{p}(\hat{\mathbf{x}}) = \boldsymbol{\kappa}'^\top(\hat{\mathbf{x}})\hat{\boldsymbol{\beta}}$ where $\boldsymbol{\kappa}'(\hat{\mathbf{x}}) := [\kappa'(\hat{\mathbf{x}}, \hat{\mathbf{x}}_1), \dots, \kappa'(\hat{\mathbf{x}}, \hat{\mathbf{x}}_N)]^\top$, $\hat{\boldsymbol{\beta}} := (\mathbf{K}' + \lambda' N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}$, and \mathbf{K}' is an $N \times N$ matrix with (n, n') -th entry $\kappa'(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_{n'})$ and κ' is a Gaussian radial basis function with parameter σ' . In this way, this benchmark LocB algorithm coincides with those in [4, 13] when a power map must be estimated on a single frequency and with a single kernel. In all experiments, the values of λ , λ' , σ , and σ' used by the LocF and LocB schemes were tuned to approximately yield the lowest NMSE.

Fig. B.5 (left) depicts the true map generated through the multi-wall model, where the black crosses indicate the sensor locations and the solid white lines represent the walls of the building. The middle and right panels respectively show the LocB and LocF map estimates, obtained by placing a query sensor at every location. It is observed that the quality of the LocF estimate is considerably higher than that of the LocB estimate. The cause for the poor performance of the LocB algorithm is that the location estimates evolve in a non-smooth fashion across space, and attempting to learn the C2M from such non-smooth features is more challenging; see Figs. B.1c and B.1d and the discussion in Sec. B.1. To illustrate how the LocF approach alleviates this issue, Fig. B.6 depicts the

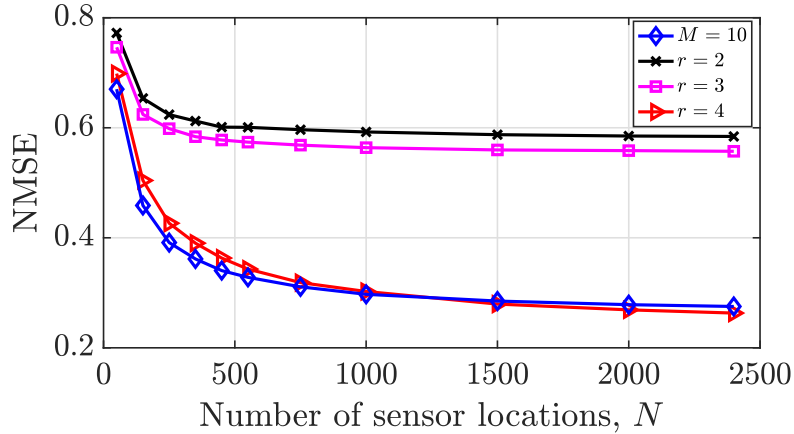


Figure B.11: Estimated map NMSE with reduced features for different r and without reduced features; $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.6 \times 10^{-3}$, and $\sigma = 25$ m.

features used by the LocF estimator across \mathcal{X} . Specifically, if $\phi(\mathbf{x})$ denotes the feature vector, obtained as in (B.28) for location \mathbf{x} , then the m -th panel titled φ_m in Fig. B.6 corresponds to the m -th entry of $\phi(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{X}$. It is observed that the evolution of these proposed features across space is significantly smoother than the one in Figs. B.1c and B.1d. A quantitative comparison is provided in Fig. B.7, which shows the NMSE as a function of the number of sensor locations N for $L = 4$ and 7 transmitters. The error bars delimit intervals of 6 standard deviations of the NMSE across the 200 independent Monte Carlo runs. It is observed that, with high significance, the proposed LocF cartography scheme outperforms its LocB counterpart for both values of L provided that the number of measurement locations is roughly larger than 150.

The rest of the section studies the impact of multipath on the LocF and LocB cartography approaches by varying the number of walls. Fig. B.8 shows the NMSE as a function of the number of walls for different values of B . The parameters used for both LocF and LocB schemes are listed in Table B.1. The NMSE is obtained by also averaging over wall locations, which are confined to be in the positions of the walls in Fig. B.6 plus an additional wall that divides the room in two.

As expected, for all the simulated values of B , the performance of both LocF and LocB schemes is degraded (yet more severely in LocB) as the number of walls increases. Moreover, the performance of the LocB improves significantly with the bandwidth, since a higher bandwidth allows a more accurate estimation of the TDoA. This is because multipath components arriving within a time interval of length $T = 1/B$ cannot be resolved; see Sec. B.4.1 and references therein. As intuition predicts, when multipath is sufficiently low and the bandwidth is sufficiently high, LocB cartography outperforms LocF. It is remarkable that LocF cartography exhibits robustness to multipath since the NMSE remains approximately constant even for a significant increase of multipath.

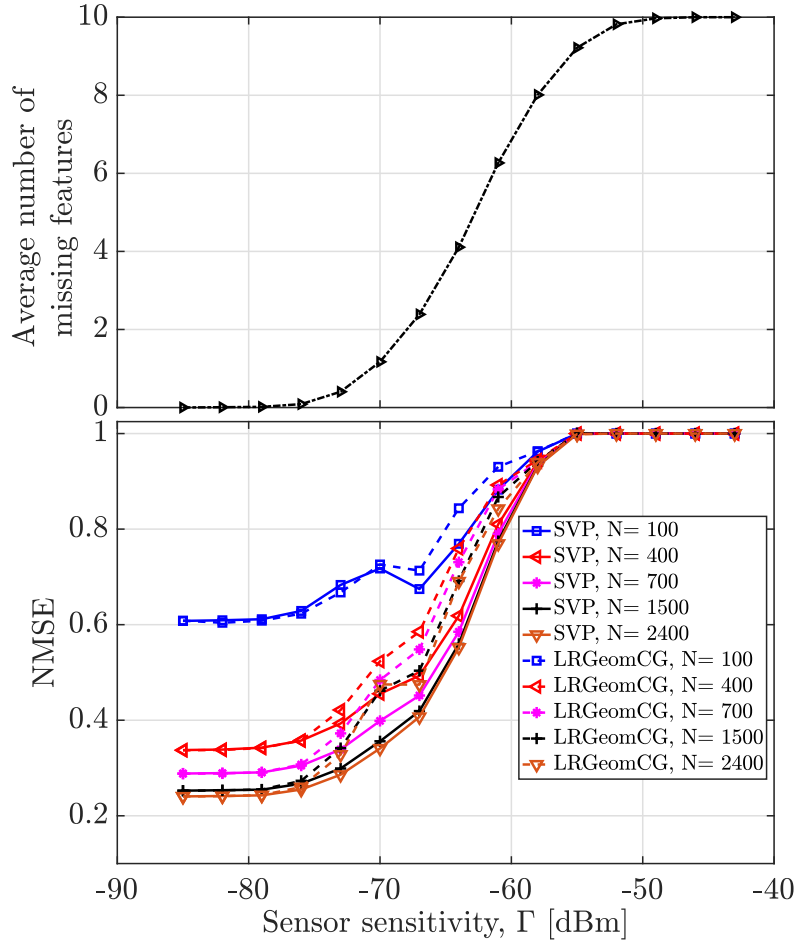


Figure B.12: (top) Average number of missing features and (bottom) estimated map NMSE, both as a function of Γ with $L = 5$, $B = 20$ MHz, $K = 10$, $\lambda = 1.9 \times 10^{-4}$, $\mu = 5.42$, and $\sigma = 37$ m.

B.5.2 Feature Design

This section provides empirical support for the findings in Sec. B.4.2. From now on, all experiments will involve only the LocF estimator. The first experiment investigates the impact of the number of features, which in all previous simulations was equal to $M = L(L - 1)/2$. To this end, Fig. B.9 shows the NMSE as a function of the number M of features for two different numbers N of sensor locations. The expectation operators in (B.27) also average with respect to all choices of M features out of the $L(L - 1)/2$. As observed, the NMSE improves from $M = 4$ to roughly $M = 7$ features, and remains approximately the same for $M \geq 7$. Although this effect may look counter-intuitive at first glance, this is a common phenomenon in machine learning related to the bias-variance trade-off [32] and the curse of dimensionality [31,32]; see Sec. B.3.1. Clearly, this effect motivates the feature dimensionality reduction techniques proposed in Sec. B.4.2. The rest of this section corroborates the merits of such techniques. A more challenging scenario with more walls will be considered. The first step is to determine the number of reduced features to be used. It can be seen that $r = 4$ in (B.20) retains at least $\eta = 99\%$ of the variance of the features in all tested scenarios. Thus, in principle, a

map can be learned using the reduced features $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n \in \mathbb{R}^4$ without meaningfully sacrificing estimation performance. Before corroborating that this is actually the case, it is instructive to visualize the aforementioned reduced features across space. Fig. B.10a portrays the maps of the $M = 10$ original features, which correspond to the entries of $\phi(\mathbf{x})$; see Sec. B.5.1. On the other hand, the panels of Fig. B.10b depict the reduced features over space, i.e., the 4 entries of the vector $\bar{\phi}(\mathbf{x}) := \mathbf{U}_1^\top \phi(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{X}$. These figures reveal that the reduced features inherit the spatial smoothness of the original features.

To quantify the impact of reducing the dimensionality of the feature vectors, Fig. B.11 compares the NMSE of the LocF map estimate that relies on the original features ($M = 10$) with the one that relies on the reduced features ($r = 2, 3, 4$). As observed, using just the 4 reduced features attains a similar performance to the estimator built on the 10 original features. This is expected given the bias-variance trade-off mentioned earlier. At this point, it might seem that the effects observed in Fig. B.9 contradict those of Fig. B.11 since in the former the NMSE is lower when 10 features are used relative to the case where only 4 are used. However, that should not be concluded since the features in Fig. B.9 correspond to the entries of ϕ_n (see (B.28)) whereas the features in Fig. B.11 correspond to the entries of $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n$.

B.5.3 LocF cartography with Missing Features

This section assesses the performance of the techniques developed in Sec. B.4.3 to cope with missing features.

A feature will be deemed missing at a given sensor location if the received power of at least one of the two associated pilot signals is below a sensitivity threshold Γ . The top panel of Fig. B.12 depicts the average number of missing features as a function of Γ . The average is taken with respect to the sensor locations and noise. The bottom panel of Fig. B.12 shows the LocF map NMSE also as a function of Γ . The matrix completion problem in (B.23) is solved with both SVP and LRGeomCG; the implementation for the latter is the one provided in the ManOpt toolbox [49]. For higher values of N , the performance of both algorithms is clearly strongly determined by the average number of missing features. SVP seems to outperform LRGeomCG in terms of NMSE. Besides, the computation time of SVP is roughly half the one of LRGeomCG.

B.6 Conclusions

Location-free (LocF) cartography has been proposed as an alternative to classical location-based (LocB) schemes, which suffer a strong performance degradation when multipath impairs the propagation of localization pilot signals. The central idea is to learn a map as a function of certain features of the localization pilot signals. Building upon this approach, kernel-ridge regression was applied to estimate power maps from these features. Practical issues addressed in the paper include feature design, dimensionality reduction, and dealing with missing features. Simulations corroborate the merits of LocF cartography relative to LocB alternatives. Future research will include mapping other channel metrics such

as power spectral density (PSD) and channel gain, as well as developing distributed and online extensions.

References

- [1] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.
- [4] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.
- [5] S. Grimoud, S. B. Jemaa, B. Sayrac, and E. Moulines, “A REM enabled soft frequency reuse scheme,” in *Proc. IEEE Global Commun. Conf.*, Miami, FL, Dec. 2010, pp. 819–823.
- [6] E. Dall’Anese, S.-J. Kim, G. B. Giannakis, and S. Pupolin, “Power control for cognitive radio networks under channel uncertainty,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Aug. 2011.
- [7] W.C.M.V. Beers and J.P.C. Kleijnen, “Kriging interpolation in simulation: A survey,” in *Proc. IEEE Winter Simulation Conf.*, Washington, D. C., Dec. 2004, vol. 1, pp. 113–121.
- [8] G. Boccolini, G. Hernandez-Penalosa, and B. Beferull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.
- [9] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [10] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, “Joint link learning and cognitive radio sensing,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.

- [11] S.-J. Kim and G. B. Giannakis, "Cognitive radio spectrum prediction using dictionary learning," in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [12] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, "Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [13] J.-A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via kernel-based learning," *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [14] M. Hamid and B. Beferull-Lozano, "Non-parametric spectrum cartography using adaptive radial basis functions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [15] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, "Group-lasso on splines for spectrum cartography," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [16] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.
- [17] D. Romero, D. Lee, and G. B. Giannakis, "Blind channel gain cartography," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Greater Washington, D. C., Dec. 2016, pp. 1110–1115.
- [18] D. Lee, S.-J. Kim, and G. B. Giannakis, "Channel gain cartography for cognitive radios leveraging low rank and sparsity," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5953–5966, Jun. 2017.
- [19] D. Lee, D. Berberidis, and G. B. Giannakis, "Adaptive Bayesian channel gain cartography," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Calgary, Canada, Apr. 2018, pp. 3555–3558.
- [20] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen, "Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on WiMAX networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 283–294, Aug. 2009.
- [21] P. S. Naidu, *Distributed Sensor Arrays: Localization*, CRC Press, 2017.
- [22] A. Bensky, *Wireless Positioning Technologies and Applications*, Artech House, 2016.
- [23] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybernetics C., Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

- [24] InfSoft, “Ultra-wideband,” [Online]. Available: <https://www.ultrawideband.io/en/technology.php>.
- [25] L. Yang and G. B. Giannakis, “Ultra-wideband communications: An idea whose time has come,” *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 26–54, Nov. 2004.
- [26] D. Dardari, C.-C. Chong, and M. Win, “Threshold-based time-of-arrival estimators in UWB dense multipath channels,” *IEEE Trans. Commun.*, vol. 56, no. 8, pp. 1366–1378, Aug. 2008.
- [27] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Computer Networks*, vol. 47, no. 6, pp. 825–845, Apr. 2005.
- [28] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, “On indoor position location with wireless LANs,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Lisboa, Portugal, Sep. 2002, vol. 2, pp. 720–724.
- [29] HP, “SmartLOCUS,” [Online]. Available: <https://www.rfidjournal.com>.
- [30] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID,” in *Proc. IEEE Int. Conf. Pervasive Computing Commun.*, Fort Worth, TX, Mar. 2003, pp. 407–415.
- [31] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. Springer, 2006.
- [32] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 2007.
- [33] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
- [34] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *Proc. Comput. Learning Theory*, Amsterdam, The Netherlands, Jul. 2001, pp. 416–426.
- [35] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [36] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*, California Technical Publishing, 1997.
- [37] N. E. Gemayel, S. Koslowski, F. K. Jondral, and J. Tschan, “A low cost TDOA localization system: Setup, challenges and results,” in *Proc. Workshop Pos. Navigation Commun.*, Dresden, Germany, Mar. 2013, pp. 1–4.
- [38] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, 2002.
- [39] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.

- [40] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM J. Opt.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.
- [41] E. J. Candès and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, Apr. 2010.
- [42] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2009.
- [43] B. Vandereycken, “Low-rank matrix completion by riemannian optimization,” *SIAM J. Opt.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.
- [44] P. Jain, R. Meka, and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” in *Proc. Advances in Neural Inf. Proces. Syst.*, Vancouver, Canada, Dec. 2010, pp. 937–945.
- [45] S. Hosseinzadeh, H. Larijani, and K. Curtis, “An enhanced modified multi wall propagation model,” in *Proc. IEEE Global Internet of Things Summit*, Geneva, Switzerland, Jun. 2017, pp. 1–4.
- [46] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal kernels,” *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006.
- [47] D. Ismailova and W.-S. Lu, “Improved least-squares methods for source localization: An iterative re-weighting approach,” in *Proc. IEEE Int. Conf. Dig. Signal Process.*, Singapore, Jul. 2015, pp. 665–669.
- [48] R. Kaune, “Accuracy studies for TDOA and TOA localization,” in *Proc. IEEE Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 408–415.
- [49] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “ManOpt, a MATLAB toolbox for optimization on manifolds,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1455–1459, Apr. 2014.

Appendix C

Paper C

Title: Data-Driven Spectrum Cartography via Deep Completion Autoencoders

Authors: Yves Teganya and Daniel Romero

Affiliation: Dept. of Information and Communication Technology, University of Agder, P. O. Box 509, NO-4898 Grimstad, Norway

Conference: *IEEE International Conference on Communications*, Dublin, Ireland, Jun. 2020, pp. 1–7

Copyright ©: IEEE

Data-Driven Spectrum Cartography via Deep Completion Autoencoders

Yves Teganya and Daniel Romero

Abstract — Spectrum maps, which provide RF spectrum metrics such as power spectral density for every location in a geographic area, find numerous applications in wireless communications such as interference control, spectrum management, resource allocation, and network planning to name a few. Spectrum cartography techniques construct these maps from a collection of measurements collected by spatially distributed sensors. Due to the nature of the propagation of electromagnetic waves, spectrum maps are complicated functions of the spatial coordinates. For this reason, model-free approaches have been preferred. However, all existing schemes rely on some interpolation algorithm unable to learn from data. This paper proposes a novel approach to spectrum cartography where propagation phenomena are learned from data. The resulting algorithms can therefore construct a spectrum map from a significantly smaller number of measurements than existing schemes since the spatial structure of shadowing and other phenomena is previously learned from maps in other environments. Besides the aforementioned new paradigm, this is also the first work to perform spectrum cartography with deep neural networks. To exploit the manifold structure of spectrum maps, a deep network architecture is proposed based on completion autoencoders.

Keywords— Spectrum cartography, deep learning, cognitive radio, completion autoencoders.

C.1 Introduction

Spectrum cartography constructs maps of RF channel metrics such as received signal power, interference power, power spectral density (PSD), electromagnetic absorption, or channel gain; see e.g. [1–3]. Besides applications like source localization [2] or radio tomography [4, 5], spectral maps find a myriad of applications in wireless communications such as network planning, interference coordination, power control, spectrum management, resource allocation, handoff procedure design, dynamic spectrum access, and cognitive radio [6–8]. Spectrum maps are constructed from measurements acquired by spectrum sensors or mobile devices.

Most approaches are based on some interpolation algorithm. For example, power maps have been constructed through kriging [1,9], dictionary learning [10,11], compressive sensing [3], Bayesian models [12], matrix completion [13], and kernel methods [14, 15]. PSD maps have also been constructed by exploiting the sparsity of power across space and frequency [2] as well as by applying thin-plate spline regression [16] and kernel-based learning [8, 17]. Metrics other than power and PSD have also been mapped in the literature. For example, [5, 18, 19] are capable of constructing channel gain maps.

Unfortunately, none of the existing approaches can learn from data. This means that they fail to learn the characteristics of the propagation phenomena and, therefore, a substantial performance improvement is expected if such knowledge can be incorporated.

To address this limitation, the first contribution of this paper is a data-driven paradigm for spectrum cartography. Specifically, it proposes learning the spatial features of the relevant propagation phenomena such as shadowing, reflection, and diffraction using a data set of past measurements. Intuitively, leveraging these learned features can significantly reduce the number of measurements required to attain a target performance. This aspect is critical since all measurements need to be collected in a sufficiently short time since the mapped metric is subject to temporal variations in real-world scenarios. The second contribution comprises a spectrum cartography algorithm to construct PSD maps relying on a deep neural network. Although several approaches for applying this class of networks are discussed, the most natural one relies on a spatial discretization of the area of interest. The resulting tensor completion task is addressed by means of a *completion network* architecture with an encoder-decoder structure that capitalizes on the observation that spectrum maps lie close to a low-dimensional manifold embedded in a high-dimensional space. Our experiments reveal that the performance of such algorithm beats the state-of-the-art alternatives. Finally, all code, trained networks, and the data set constructed for this work will be posted at the authors' web sites.

The novelty of this paper is twofold. First, this is the first work to propose a data-driven spectrum cartography approach. Second, this is the first work to propose a deep learning algorithm for spectrum cartography.

The rest of this paper is organized as follows. Sec. C.2 describes the problem of PSD cartography. Sec. C.3 presents the aforementioned data-driven spectrum cartography paradigm and proposes a deep neural network architecture based on completion autoencoders. Simulations and conclusions are respectively provided in Secs. C.4 and C.5. Unfortunately, due to space requirements, we had to omit many insightful explanations and a large number of experiments that further support the proposed approach.

Notation: $|\mathcal{A}|$ denotes the cardinality of set \mathcal{A} . $[\mathbf{A}]_{i,j}$ is the (i, j) -th entry of matrix \mathbf{A} , whereas $[\mathbf{B}]_{i,j,k}$ is the (i, j, k) -th entry of tensor \mathbf{B} . Finally, \mathbf{A}^\top is the transpose of matrix \mathbf{A} .

C.2 Model and Problem Formulation

Consider L sources located in a geographical region of interest $\mathcal{X} \subset \mathbb{R}^2$ and operating on a certain frequency band. Let $\Upsilon_l(f)$ denote the transmit PSD of the l -th source and let $H_l(\mathbf{x}, f)$ represent the frequency response of the channel between the l -th source and a sensor with an isotropic antenna located at $\mathbf{x} \in \mathcal{X}$. For simplicity, assume that small-scale fading has been averaged out; see also Remark 5. Both $\Upsilon_l(f)$ and $H_l(\mathbf{x}, f)$ are assumed to remain constant over time, a realistic assumption provided that the measurements described below are collected in an interval of shorter length than the channel coherence time and time scale of changes in $\Upsilon_l(f)$.

If the L signals are uncorrelated, the PSD at $\mathbf{x} \in \mathcal{X}$ is

$$\Psi(\mathbf{x}, f) = \sum_{l=1}^L \Upsilon_l(f) |H_l(\mathbf{x}, f)|^2 + v(\mathbf{x}, f)$$

with $v(\mathbf{x}, f)$ the noise PSD of a generic sensor at location \mathbf{x} , which models thermal noise, background radiation noise, and interference from remote sources. A certain number of devices, such as mobile users in a cellular communication network or spectrum sensors, collect PSD measurements $\{\tilde{\Psi}(\mathbf{x}_n, f)\}_{n=1}^N$ at N locations $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$ and finite set of frequencies $f \in \mathcal{F}$; see also Remark 6. These measurements can be obtained using e.g. periodograms or spectral analysis methods such as the Bartlett or Welch method [20].

These measurements are sent to a fusion center, which may be e.g. a base station, a mobile user, or a cloud server, depending on the application. Given $\{(\mathbf{x}_n, \tilde{\Psi}(\mathbf{x}_n, f)), n = 1, \dots, N, f \in \mathcal{F}\}$, the fusion center must obtain an estimate $\hat{\Psi}(\mathbf{x}, f)$ of $\Psi(\mathbf{x}, f)$ at every location $\mathbf{x} \in \mathcal{X}$ and frequency $f \in \mathcal{F}$. In spectrum cartography, function $\Psi(\mathbf{x}, f)$ is typically referred to as the *true map*, whereas $\hat{\Psi}(\mathbf{x}, f)$ is the *estimated map* or *map estimate*. The algorithm or rule that provides a map estimate, which in this paper is a neural network, is termed *map estimator*. The challenge is to exploit the spatial structure of propagation phenomena so that the estimation error, quantified e.g. as $\sum_f \int_{\mathcal{X}} |\Psi(\mathbf{x}, f) - \hat{\Psi}(\mathbf{x}, f)|^2 d\mathbf{x}$, is minimized for a certain N or, alternatively, the minimum N required to guarantee a target estimation error is minimized.

To the best of our knowledge, all existing approaches to spectrum cartography are based on interpolation algorithms that do not learn from data. In contrast, the next section develops a novel data-driven methodology that learns the aforementioned structure from a data set.

Remark 3 *Sensors must determine their locations $\{\mathbf{x}_n\}_n$ with an error sufficiently small relative to the scale of spatial variability of $\Psi(\mathbf{x}, f)$ across \mathcal{X} . Thus, estimating small-scale fading is more challenging than estimating shadowing since the coherence distance of the former is comparable to the wavelength and typical communication bands of interest have wavelengths in the order of centimeters.*

Remark 4 *The number of measurement locations may be significantly larger than the number of sensors if the sensors move. Measurements collected at different locations may be useful to estimate a spectrum map so long as the difference between measurement instants is small relative to the time scale of the variations of the PSD map. The latter*

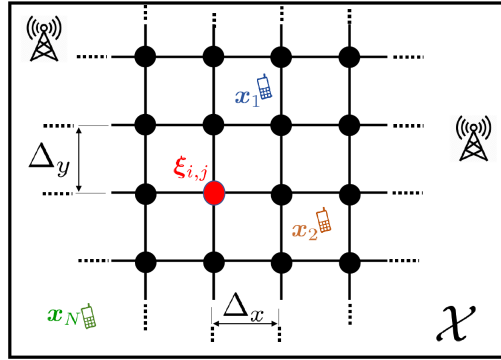


Figure C.1: Model setup and area discretization.

is highly dependent on the specific application. For example, one expects significant variations in DVB-T bands to occur in the scale of several months, whereas PSD maps in LTE bands may change in the scale of milliseconds due to power control, mobility, and interference.

C.3 Proposed Data-Driven Cartography

This section introduces a data-driven paradigm for spectrum cartography and develops a deep learning algorithm that abides by this principle. To this end, Sec. C.3.1 starts by reformulating the problem at hand as a tensor-completion task amenable to application of deep neural networks. Subsequently, Sec. C.3.2 addresses unique aspects of tensor/matrix completion via deep learning. Finally, Secs. C.3.3 and C.3.4 respectively describe how a deep neural network can be trained to learn the spatial structure of propagation phenomena and how this task can be addressed via the notion of *completion autoencoders*.

C.3.1 Spectrum Cartography as a Tensor Completion Task

Observe that the value of N depends on the number and movement of the sensors relative to the time-scale of temporal changes in $\Psi(\mathbf{x}, f)$; cf. Sec. C.2. In principle, one could think of using a separate map estimator for each possible value of N . Each estimator could be relatively simple since it would always take the same number of inputs. However, such an approach would be highly inefficient in terms of memory, computation, and prone to erratic behavior since each estimator is trained with a different data set. Thus, it is more practical to rely on a single estimator that can accommodate arbitrary values of N . A customary approach in deep learning for coping with inputs of variable lengths is through recurrent neural networks [21], [22, Ch. 10]. Unfortunately, besides the difficulties of training these networks, it is unclear how such an approach could effectively exploit spatial information. For this reason, the selected approach in this work is to reformulate the cartography problem as a tensor completion task amenable to a solution based on a feedforward architecture [22, Ch. 6].

To this end, one must discretize \mathcal{X} , a trick already applied in radio tomographic imaging [23, 24] and spectrum cartography [13]. To introduce the appropriate notation,

it will be briefly outlined next. Define an $N_y \times N_x$ rectangular grid over \mathcal{X} , as depicted in Fig. C.1. This grid comprises points $\boldsymbol{\xi}_{i,j}$ evenly spaced by Δ_x and Δ_y along the x - and y -axes respectively, that is, the (i, j) -th grid point is given by $\boldsymbol{\xi}_{i,j} := [i\Delta_x, j\Delta_y]^\top$, with $i = 1, \dots, N_y$, $j = 1, \dots, N_x$. For future usage, define $\mathcal{A}_{i,j} \subset \{1, \dots, N\}$ as the set containing the indices of the measurement locations assigned to the (i, j) -th grid point by the criterion of minimum distance, i.e., $n \in \mathcal{A}_{i,j}$ iff $\|\boldsymbol{\xi}_{i,j} - \mathbf{x}_n\| \leq \|\boldsymbol{\xi}_{i',j'} - \mathbf{x}_n\| \forall i', j'$ with $i' \neq i$ and $j' \neq j$.

This grid induces a discretization of $\Psi(\mathbf{x}, f)$ along the \mathbf{x} variable. One can therefore collect the true PSD values at the grid points in matrix $\boldsymbol{\Psi}(f) \in \mathbb{R}^{N_y \times N_x}$, $f \in \mathcal{F}$, whose (i, j) -th entry is given by $[\boldsymbol{\Psi}(f)]_{i,j} = \Psi(\boldsymbol{\xi}_{i,j}, f)$. By letting $\mathcal{F} = \{f_1, \dots, f_{N_f}\}$, it is also possible to stack these matrices along the third dimension to form the tensor $\boldsymbol{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\boldsymbol{\Psi}]_{i,j,n_f} = \Psi(\boldsymbol{\xi}_{i,j}, f_{n_f})$, $n_f = 1, \dots, N_f$. For short, the term *true map* will either refer to $\Psi(\mathbf{x}, f)$ or $\boldsymbol{\Psi}$.

Similarly, one can collect the measurements in a tensor of the same dimensions. Informally, if the grid is sufficiently fine (Δ_x and Δ_y are sufficiently small), it holds that $\mathbf{x}_n \approx \boldsymbol{\xi}_{i,j} \forall n \in \mathcal{A}_{i,j}$ and, correspondingly, $\Psi(\mathbf{x}_n, f) \approx \Psi(\boldsymbol{\xi}_{i,j}, f) \forall n \in \mathcal{A}_{i,j}$. It follows that,

$$\Psi(\boldsymbol{\xi}_{i,j}, f) \approx \frac{1}{|\mathcal{A}_{i,j}|} \sum_{n \in \mathcal{A}_{i,j}} \Psi(\mathbf{x}_n, f)$$

whenever $|\mathcal{A}_{i,j}| \geq 1$. Therefore, it makes sense to aggregate all the measurements assigned to $\boldsymbol{\xi}_{i,j}$ as¹

$$\tilde{\Psi}(\boldsymbol{\xi}_{i,j}, f) := \frac{1}{|\mathcal{A}_{i,j}|} \sum_{n \in \mathcal{A}_{i,j}} \tilde{\Psi}(\mathbf{x}_n, f).$$

Conversely, when $|\mathcal{A}_{i,j}| = 0$, there are no measurements associated with $\boldsymbol{\xi}_{i,j}$, in which case one says that there is a *miss* at $\boldsymbol{\xi}_{i,j}$. Upon letting $\Omega \subset \{1, \dots, N_y\} \times \{1, \dots, N_x\}$ be such that $(i, j) \in \Omega$ iff $|\mathcal{A}_{i,j}| > 0$, all (possibly aggregated) measurements $\tilde{\Psi}(\boldsymbol{\xi}_{i,j}, f)$ can be collected in $\tilde{\boldsymbol{\Psi}}(f) \in \mathbb{R}^{N_y \times N_x}$, defined as

$$[\tilde{\boldsymbol{\Psi}}(f)]_{i,j} = \begin{cases} \tilde{\Psi}(\boldsymbol{\xi}_{i,j}, f) & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

Note that misses have been filled with zeroes, but other values can be used.

When $(i, j) \in \Omega$, the values of $[\tilde{\boldsymbol{\Psi}}(f)]_{i,j}$ and $[\boldsymbol{\Psi}(f)]_{i,j}$ differ due to the error introduced by the spatial discretization as well as due to the measurement error incurred when measuring $\Psi(\mathbf{x}_n, f)$, $n \in \mathcal{A}_{i,j}$. The latter is caused mainly by the finite time devoted by sensors to take measurements.

As before, the matrices $\tilde{\boldsymbol{\Psi}}(f)$, $f = 1, \dots, N_f$ can be stacked along the 3rd dimension to form $\tilde{\boldsymbol{\Psi}} \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\tilde{\boldsymbol{\Psi}}]_{i,j,n_f} = [\tilde{\boldsymbol{\Psi}}(f_{n_f})]_{i,j}$. For short, this tensor will be referred to as the *sampled map*.

The cartography problem stated in Sec. C.2 can now be reformulated as, given Ω and $\tilde{\boldsymbol{\Psi}}$, estimate $\boldsymbol{\Psi}$.

¹For simplicity, the notation implicitly assumes that $\mathbf{x}_n \neq \boldsymbol{\xi}_{i,j} \forall n, i, j$, but this is not a requirement.

C.3.2 Feedforward Completion Networks

The previous section reformulated the spectrum cartography problem as a tensor completion task. Since conventional neural networks cannot directly accommodate input misses and set-valued inputs like Ω , this section explores the possibilities and motivates the adopted approach.

But before that, a quick refresh on deep learning is in order. A feedforward deep neural network is a function $p_{\mathbf{w}}$ that can be expressed as the composition $p_{\mathbf{w}}(\Phi) = p_{\mathbf{w}_L}^{(L)}(p_{\mathbf{w}_{L-1}}^{(L-1)}(\dots p_{\mathbf{w}_1}^{(1)}(\Phi)))$ of *layer* functions $p_{\mathbf{w}_l}^{(l)}$, where Φ is the input. Although there is no commonly agreed definition of layer function, it is typically formed by concatenating simple scalar-valued functions termed *neurons* that implement a linear function followed by a simple non-linear function known as activation [22]. The term *neuron* stems from the resemblance between these functions and certain simple functional models for natural neurons. Similarly, there is no consensus on which values of L qualify for $p_{\mathbf{w}}$ to be considered a *deep* neural network. With vector \mathbf{w}_l containing the parameters of the l -th layer, the parameters of the entire network can be collected in $\mathbf{w} := [\mathbf{w}_1^\top, \dots, \mathbf{w}_L^\top]^\top \in \mathbb{R}^{N_w}$. These parameters are *learned* using a *training set* in a process termed *training*.

The rest of this section as well as Sec. C.3.3 carefully delineate how a deep neural network can be trained to perform data-driven spectrum cartography. Occasional references to works in areas such as collaborative filtering and image inpainting will provide insight and motivate the design decisions. On the other hand, Sec. C.3.4 will address the design of $p_{\mathbf{w}}$.

Although the training set construction is detailed in Sec. C.3.3, suppose by now that a set of T *training examples* $\{(\tilde{\Psi}_t, \Omega_t)\}_{t=1}^T$ is given. Here, $\{\tilde{\Psi}_t\}_t$ is a collection of sampled maps acquired in different environments and Ω_t the corresponding sampling set.

The desired estimator should obtain Ψ as a function of $\tilde{\Psi}$ and Ω . But regular neural networks cannot directly accommodate set-valued inputs and missing entries. For this reason, [25] proposes filling the missing entries in $\tilde{\Psi}$ by solving

$$\begin{aligned} \underset{\{\chi_t\}_t, \mathbf{w}}{\text{minimize}} \quad & \frac{1}{T} \sum_{t=1}^T \|\mathcal{P}_{\Omega_t}(\chi_t - p_{\mathbf{w}}(\chi_t))\|_F^2, \\ & [\chi_t]_{i,j,n_f} = [\tilde{\Psi}_t]_{i,j,n_f} \quad \forall n_f, \forall (i,j) \in \Omega_t, \end{aligned} \quad (\text{C.1})$$

where $\|\mathbf{A}\|_F^2 := \sum_{i,j,n_f} [\mathbf{A}]_{i,j,n_f}^2$ is the Frobenius norm of tensor \mathbf{A} and $\mathcal{P}_{\Omega}(\mathbf{A})$ is defined as $[\mathcal{P}_{\Omega}(\mathbf{A})]_{i,j,n_f} = [\mathbf{A}]_{i,j,n_f}$ if $(i,j) \in \Omega$ and $[\mathcal{P}_{\Omega}(\mathbf{A})]_{i,j,n_f} = 0$ otherwise. The map estimate produced by this method is directly the minimizer χ_t of (C.1). Observe that if there exists a value of \mathbf{w} for which $p_{\mathbf{w}}$ becomes the identity map, i.e. $\chi = p_{\mathbf{w}}(\chi)$, $\forall \chi$, then the optimum of (C.1) is attained regardless of the value of the entries $[\chi_t]_{i,j,n_f}$, $(i,j) \notin \Omega_t$, which would render this estimator useless. Thus, some form of *capacity/complexity control* is necessary [26]. For instance, one can (i) impose constraints on \mathbf{w} , (ii) add a regularization term to the objective function, or (iii) limit capacity through the design of the network architecture. Approach (iii) will be discussed further in Sec. C.3.4. To simplify the exposition, expressions in this paper will not display constraints or regularizers, but it is understood that the user may include them if necessary.

After $\mathbf{w} = \hat{\mathbf{w}}$ has been obtained by applying (C.1) with sufficiently large T , one can complete further tensors $\tilde{\Psi}_t$ by setting \mathbf{w} in (C.1) to this learned vector $\hat{\mathbf{w}}$ and optimize only with respect to $\{\chi_t\}_t$, which is computationally simpler.

The number of optimization variables in (C.1) is $N_w + N_x N_y N_f T$, where N_w is the length of \mathbf{w} . This number is prohibitive for high T , as required for training deep neural networks. Besides, even with the aforementioned simplified approach that only optimizes with respect to $\{\chi_t\}_t$, a large number of forward and backward backpropagation passes [22, Ch. 6] are required to estimate each map. Thus, this approach is not suitable for real-time implementation, as required in spectrum cartography applications.

To alleviate this limitation, a simple alternative would be to just feed $\tilde{\Psi}$ to the neural network and train by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t - p_{\mathbf{w}}(\tilde{\Psi}_t) \right) \right\|_F^2. \quad (\text{C.2})$$

Although the missing entries were filled with zeros in Sec. C.3.1, one can alternatively use other real numbers. After (C.2) is solved, $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\tilde{\Psi})$, which requires a single forward pass. Besides, solving (C.2) involves just N_w optimization variables. However, because the completion step $p_{\mathbf{w}}(\tilde{\Psi})$ does not involve Ω , poor performance is expected since the network cannot distinguish missing entries from measurements close to the filling value.

In the application at hand, one could circumvent this limitation by expressing the entries of $\tilde{\Psi}$ in natural power units (e.g. Watt) and filling the misses with a negative number such as -1. Unfortunately, the usage of finite-precision arithmetic would introduce large errors in the map estimates and is problematic in our experience. For this reason, expressing $\tilde{\Psi}$ in logarithmic units such as dBm is preferable. However, the problem of distinguishing missing entries persists since logarithmic units are not confined to be non-negative.

A more practical approach is to complement the input of the network with a binary mask that indicates which entries are observed, as proposed in the image inpainting literature [27]. In this case, the binary mask $\mathbf{M}_{\Omega} \in \{0, 1\}^{N_y \times N_x}$ associated with the sampling set Ω is given by $[\mathbf{M}_{\Omega}]_{i,j} = 1$ if $(i, j) \in \Omega$ and $[\mathbf{M}_{\Omega}]_{i,j} = 0$ otherwise.

To simplify notation, let $\check{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f + 1}$ denote a tensor obtained by concatenating $\tilde{\Psi}$ and \mathbf{M}_{Ω} along the third dimension. The neural network can therefore be trained as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\check{\Psi}_t - p_{\mathbf{w}}(\check{\Psi}_t) \right) \right\|_F^2 \quad (\text{C.3})$$

and, afterwards, a tensor $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\check{\Psi})$. Then, this scheme is simple to train, inexpensive to test, and exploits information about the location of the misses.

C.3.3 Learning in Real-World Scenarios

A key novelty in this paper is to obtain map estimators by learning from data. This section describes how to construct a suitable training set in the application at hand.

The first consideration pertains to ill-conditioning issues arising when the number of frequencies N_f in \mathcal{F} is large, as will typically be the case. Suppose that the first layer of p_w is fully connected and has N_N neurons. Its total number of parameters becomes $(N_y N_x N_f + 1)N_N$ plus possibly additional parameters of the activation functions. Other layers will experience the same issue to different extents. Since T must be comparable to the number of unknowns to train the network effectively, the impact of a large N_f is to drastically limit the number of layers or neurons that can be used.

Previous approaches in spectrum cartography experienced similar issues, which were often addressed by the introduction of parametric models along the frequency domain; see e.g. [8, 16]. Although such an approach can be similarly adopted in the present work, thereby reducing the number of *channels* at the neural network input from $N_f + 1$ to a much smaller number, it will be argued next that directly separating the problem across frequencies may be preferable when training a deep neural network. The idea is that propagation phenomena at similar frequencies are expected to be similar. Building upon this principle, p_w can operate separately at each frequency f . This means that training can be accomplished through

$$\underset{w}{\text{minimize}} \frac{1}{TN_f} \sum_{t=1}^T \sum_{f \in \mathcal{F}} \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t(f) - p_w(\tilde{\Psi}_t(f)) \right) \right\|_F^2, \quad (\text{C.4})$$

where $\tilde{\Psi}_t(f) \in \mathbb{R}^{N_y \times N_x \times 2}$ is a tensor with first frontal slab given by $\tilde{\Psi}_t(f)$ and second frontal slab given by \mathbf{M}_{Ω_t} .

Observe that the number of variables is now reduced by a factor of N_f whereas the “effective” number of training examples has been multiplied by N_f ; cf. number of summands in (C.4). This is a drastic improvement especially when N_f takes values such as 512 or 1024, as customary in spectral analysis. Thus, such a frequency separation allows an increase in the number of neurons per layer or (typically more useful [22, Ch. 5]) the total number of layers for a given T . Although such a network would not exploit structure across the frequency domain, the fact that it would be better trained is likely to counteract this limitation in many setups.

The next step is to construct the data set, for which three approaches are discussed next:

C.3.3.1 Synthetic Training Data

Since collecting a large number of training maps may be slow or expensive, one can instead generate maps using a mathematical model or simulator that captures the structure of the propagation phenomena; see e.g. [28]. Fitting p_w to data generated by that model could, in principle, yield an estimator that effectively exploits the path loss and shadowing structure. The idea is therefore to generate T maps $\{\Psi_t(\mathbf{x}, f)\}_{t=1}^T$ together with T sampling sets $\{\Omega_t\}_{t=1}^T$. Afterwards, $\{\tilde{\Psi}_t\}_{t=1}^T$ and $\{\check{\Psi}_t\}_{t=1}^T$ can be formed as described earlier. It is possible to add artificially generated noise to the synthetic measurements in $\check{\Psi}_t$ to model the effect of measurement error. This would train the network to counteract the impact of such error, along the lines of denoising autoencoders [22, Ch. 14]. The advantage of this approach is that one has access to the ground truth, i.e., one can use

the true maps Ψ_t as *targets*. Specifically, the neural network can be trained on the data $\{(\check{\Psi}_t, \Psi_t)\}_{t=1}^T$ by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{TN_f} \sum_{t=1}^T \sum_{f \in \mathcal{F}} \|\Psi_t(f) - p_{\mathbf{w}}(\check{\Psi}_t(f))\|_F^2. \quad (\text{C.5})$$

If the model or simulator is sufficiently close to the reality, completing a real-world map $\check{\Psi}(f)$ as $p_{\mathbf{w}}(\check{\Psi}(f))$ should produce an accurate estimate.

C.3.3.2 Real Training Data

In practice, real maps may be available for training. However, in most cases, it will not be possible to collect measurements at all grid points before the map changes. Besides, it is not possible to obtain the entries of Ψ but only measurements of it. This means that a real training set is of the form $\{\check{\Psi}_t, t = 1, \dots, T\}$.

For training, one can plug this data directly into (C.4). However, $p_{\mathbf{w}}$ may then focus on learning just the values $\{[\check{\Psi}_t(f)]_{i,j}, (i,j) \in \Omega_t\}$, as would happen e.g. when $p_{\mathbf{w}}$ is the identity mapping. To counteract this trend, one can use one part of the measurements as the input and another part as the output (target). For each t , construct the Q_t pairs of (not necessarily disjoint) subsets $\Omega_{t,q}^{(I)}, \Omega_{t,q}^{(O)} \subset \Omega_t, q = 1, \dots, Q_t$, e.g. by drawing a given number of elements of Ω_t uniformly at random without replacement. Using these subsets, subsample $\check{\Psi}_t(f)$ to yield $\check{\Psi}_{t,q}^{(I)}(f) := \mathcal{P}_{\Omega_{t,q}^{(I)}}(\check{\Psi}_t(f))$ and $\check{\Psi}_{t,q}^{(O)}(f) := \mathcal{P}_{\Omega_{t,q}^{(O)}}(\check{\Psi}_t(f))$. With these $TN_f \sum_t Q_t$ training examples, one can think of training as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{TN_f \sum_t Q_t} \sum_{f \in \mathcal{F}} \sum_{t=1}^T \sum_{q=1}^{Q_t} \left\| \mathcal{P}_{\Omega_{t,q}^{(O)}} \left(\check{\Psi}_{t,q}^{(O)}(f) - p_{\mathbf{w}} \left(\check{\Psi}_{t,q}^{(I)}(f) \right) \right) \right\|_F^2, \quad (\text{C.6})$$

where $\check{\Psi}_{t,q}^{(I)}(f)$ has $\check{\Psi}_{t,q}^{(I)}(f)$ and $\mathbf{M}_{\Omega_{t,q}^{(I)}}$ as frontal slabs.

C.3.3.3 Hybrid Training

In practice, one expects to have real data, but only in a limited amount. It makes sense to apply the notion of *transfer learning* [22, Ch. 15] as follows: first, learn an initial parameter vector $\hat{\mathbf{w}}$ by solving (C.5) with synthetic data. Second, solve (C.6) with real data, but using $\hat{\mathbf{w}}$ as initialization for the optimization algorithm. The impact of choosing this initialization is that the result of solving (C.6) in the second step will be closer to a “better” local optimum than if a worse initialization were adopted.

C.3.4 Deep Completion Autoencoders

This section proposes a deep neural network architecture based on *convolutional autoencoders* [29].

A (conventional) autoencoder [22, Ch. 12] is a neural network $p_{\mathbf{w}}$ composed of two parts, an encoder $\epsilon_{\mathbf{w}}$ and a decoder $\delta_{\mathbf{w}}$, which satisfy $p_{\mathbf{w}}(\Phi) = \delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\Phi)) \forall \Phi$. The output of the encoder $\lambda := \epsilon_{\mathbf{w}}(\Phi) \in \mathbb{R}^{N_\lambda}$ is referred to as the *code* or vector of *latent variables* and is of a typically much lower dimension than the input Φ . An autoencoder is trained so

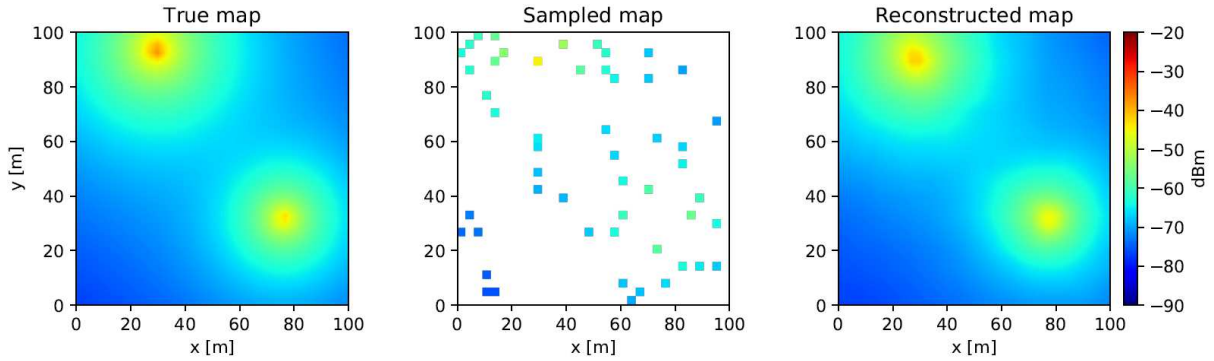


Figure C.2: Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying grid points $\{\xi_{i,j}\}$ with $|\mathcal{A}_{i,j}| > 0$, and (right) estimated map.

that $\delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\Phi)) \approx \Phi \nabla \Phi$, which forces the encoder to compress the information in Φ into the N_λ variables in λ . The selection of N_λ will be addressed later.

A *completion* autoencoder adheres to the same principles as conventional autoencoders except for the fact that the encoder must determine the latent variables from a subset of the entries of the input. If a mask is used, the transfer function must satisfy $\Phi \approx \delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\mathcal{P}_\Omega(\Phi), \mathbf{M}_\Omega)) \nabla \Phi$ and for a sampling set Ω that preserves sufficient information for reconstruction. If Ω does not satisfy this requirement, then reconstructing Φ is impossible regardless of the technique used. In the application at hand and with the notation introduced in previous sections, the above expression becomes $\tilde{\Psi} \approx \delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\check{\Psi}))$.

As indicated earlier, autoencoders are useful only when most of the information in the input can be condensed in N_λ variables, i.e., when the possible inputs lie close to a manifold of dimension N_λ . To see that this is the case in spectrum cartography, an illustrating toy example is presented next. Suppose that there are two sources, each one with a fixed (yet possibly different) power, that can be placed at arbitrary positions in \mathcal{X} and suppose that propagation occurs in free space. All possible spectrum maps in this setup are defined by $N_\lambda = 4$ quantities, which correspond to the x and y coordinates of the two sources. Fig. C.2 illustrates this effect, where the left panel of Fig. C.2 depicts a true map Ψ and the right panel shows its estimate using a completion autoencoder with $N_\lambda = 4$. The quality of the estimate clearly supports the aforementioned manifold hypothesis. Details about the network and simulation setup are provided in Sec. C.4. In a real-world scenario, there may be more than two sources, their transmit power may not always be the same, and there are shadowing effects, which means that $N_\lambda \geq 4$ will be required.

Since space limitations prevent us from detailing every design decision, the rest of this section will be confined to outline the main aspects of the architecture developed in this work and summarized in Fig. C.3.

The encoder mainly comprises convolutional and pooling layers. The motivation for convolutional layers is three-fold: (i) relative to fully connected layers, they severely reduce the number of parameters to train and, consequently, the amount of data required. Despite this drastic reduction, (ii) convolutional layers are still capable of exploiting the spatial structure of maps and (iii) they result in shift-invariant transfer functions, a desirable property in the application at hand since moving the sources in a certain direction

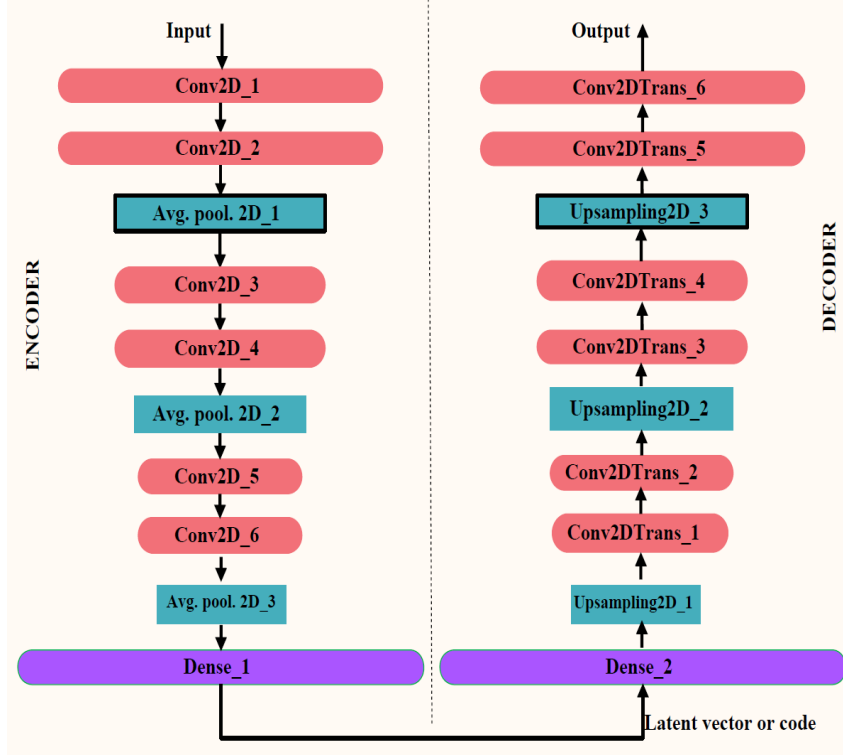


Figure C.3: Autoencoder architecture.

Table C.1: Parameters of the proposed network.

Layers	Parameters
Conv2D/ Conv2DTranspose	Kernel size = 3×3 , stride = 1, activation = PLReLU, 64 filters
AveragePooling2D	Pool size = 2, stride = 2
Upsampling2D	Up-sampling factor = 2, bilinear interpolation
Dense	64 neurons (encoder), 1024 neurons (decoder)

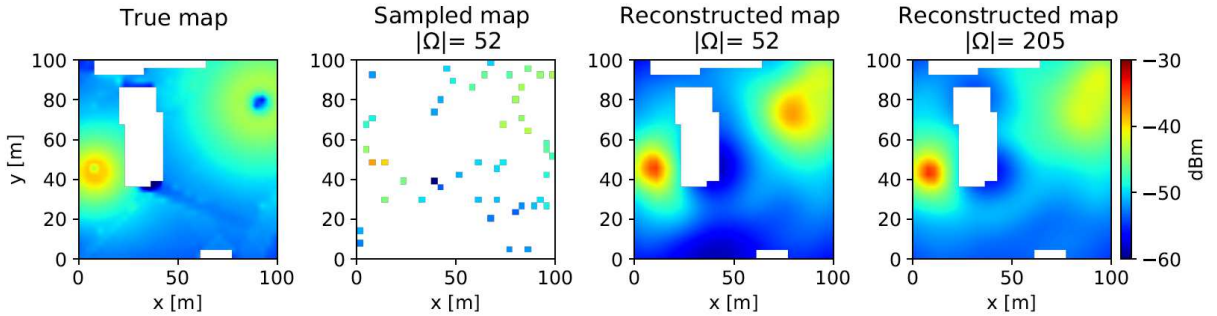


Figure C.4: Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the grid points $\{\xi_{i,j}\}$ where $|\mathcal{A}_{i,j}| > 0$; (center right) and (right): estimated maps. White areas represent buildings.

must be corresponded by the same movement in the estimated map. These layers compute

$$[\Phi^{(O)}]_{i,j,c_{out}} = \sum_{c_{in}=1}^{C_{in}} \sum_{u=-k}^k \sum_{v=-k}^k [\mathbf{F}_{c_{out}}]_{u,v,c_{in}} [\Phi^{(I)}]_{i-u,j-v,c_{in}},$$

where $\Phi^{(O)}$ is the output tensor, $\Phi^{(I)}$ is the input tensor, and $F_{c_{\text{out}}}$ is the c_{out} -th filter (or kernel), which is of size $2k + 1 \times 2k + 1$. Layer indices were omitted in order not to overload notation. The activation functions used here are parametric *leaky rectified linear units* (PLReLUs) [30] whose leaky parameter is also trained.

On the other hand, *average pooling* layers down-sample the outputs of convolutional layers, thereby condensing the information gradually in fewer features. Additionally, pooling features are approximately shift invariant as well [22, Ch. 9].

The last layer of the encoder is *fully-connected*. Since the previous layers were constrained to be convolutional or pooling layers, a final fully-connected layer is included in the encoder so that the latent variables can capture arbitrary relations among the shift invariant features obtained by the output of the second-to-last layer.

As usual in autoencoders, the decoder follows a “reverse” architecture relative to the encoder. Wherever the encoder has a convolutional layer, the decoder has a corresponding *convolution transpose* layer [31], sometimes called “deconvolutional” layer. Likewise, the pooling layers of the encoder are matched with *up-sampling* layers, which use bilinear interpolation in the architectures that we investigated. Finally, the fully connected layer of the encoder is paired with a fully connected layer in the decoder. The overall network architecture is summarized in Fig. C.3 and Table C.1.

C.4 Numerical Experiments

This section validates the proposed framework and network architecture through numerical experiments. Due to space limitations, the focus is on the most fundamental cartographic aspects, where the main novelty resides. Thus, \mathcal{F} is set to the singleton $\mathcal{F} = \{900 \text{ MHz}\}$. \mathcal{X} is a square area of side 100 m, discretized into a grid with $N_y = N_x = 32$. The two considered transmitters have height 1.5 m and transmit power 11 and 7 dBm over a bandwidth of 5 MHz.

Two classes of maps are generated. First, $T = 4 \cdot 10^5$ maps are obtained where the two transmitters are placed uniformly at random and where propagation adheres to the Gudmundson model [32] with pathloss exponent 3, gain at unit distance -30 dB, and shadowing correlation $E\{H_l(\mathbf{x}_1, f)H_l(\mathbf{x}_2, f)\} = \sigma_{\text{sh}}^2 0.95^{\|\mathbf{x}_1 - \mathbf{x}_2\|}$ with $\sigma_{\text{sh}}^2 = 10 \text{ dB}^2$. Sensors are distributed uniformly at random without replacement across the grid points. A separate set of maps is generated using Remcom’s Wireless InSite software in an urban scenario. Sensors are distributed uniformly at random without replacement across the grid points that lie on the streets. To better observe the impact of propagation phenomena, $v(\mathbf{x}, f)$ is set to 0. Each measurement $\tilde{\Psi}(\boldsymbol{\xi}_{i,j}, f)$ is obtained by adding zero-mean Gaussian noise with standard deviation 1 dB to $\Psi(\boldsymbol{\xi}_{i,j}, f)$, $(i, j) \in \Omega$.

The network proposed in Sec. C.3.4 is implemented in TensorFlow and trained using the ADAM solver with learning rate 10^{-4} . Due to lack of space, only one training approach can be analyzed, in this case (C.5) with $\{(\tilde{\Psi}_t, \Psi_t)\}_{t=1}^T$ the Gudmundson data set. The algorithm is compared against the state-of-the-art competitors described next, whose parameters were tuned to approximately optimize their performance in the second experiment. (i) The kriging algorithm in [1] with regularization parameter 10^{-5} and Gaussian

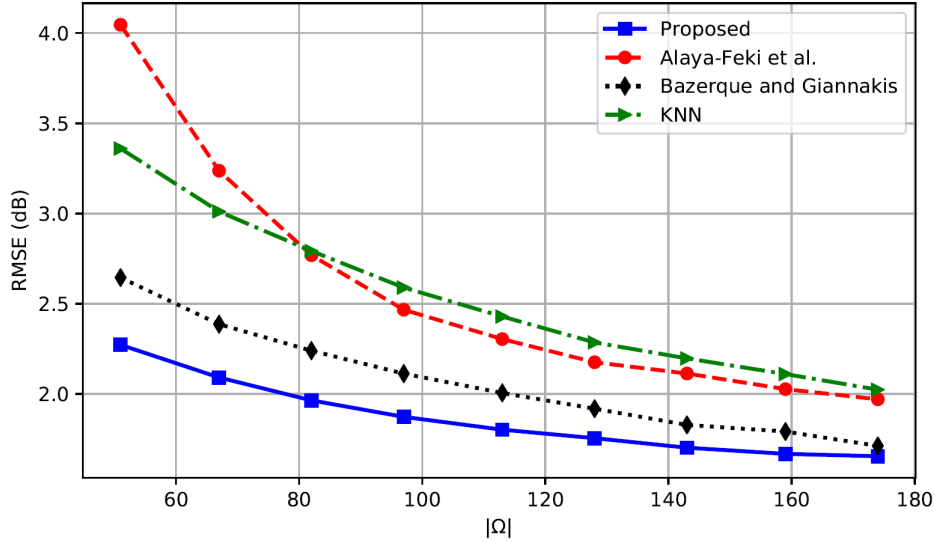


Figure C.5: Comparison with state-of-the-art alternatives. Even though the parameters of the competing algorithms were tuned for this specific experiment, the proposed network offers a markedly better performance.

radial basis functions with parameter $\sigma_K := 3\sqrt{\Delta_y N_y \Delta_x N_x / |\Omega|}$, which is approximately 3 times the mean distance between two points at which measurements have been collected. (ii) The multikernel algorithm in [17] with 20 Laplacian kernels with parameter uniformly spaced between $[0.1\sigma_K, \sigma_K]$ and regularization parameter 10^{-4} . As a benchmark, (iii) the K -nearest neighbors (KNN) algorithm with $K = 5$ is also shown.

The first experiment shows an estimated map using the proposed algorithm. The first panel of Fig. C.4 depicts the true map, which was generated using the Remcom data set. The second panel shows $\tilde{\Psi}$ whereas the third and fourth show map estimates using different numbers of measurements. Observe that with just $|\Omega| = 52$ measurements, the estimate is already of a high quality. Note that details due to diffraction or the directivity of the antennas are not reconstructed because the Gudmundson model used to train the network does not capture them and therefore the network did not learn these features. This illustrates the need for training over data sets that model the reality as close as possible.

The second experiment compares the root mean square error

$$\text{RMSE} = \sqrt{\frac{\mathbb{E}\{\|\Psi - \hat{\Psi}\|_F^2\}}{N_y N_x}},$$

of the aforementioned algorithms, where Ψ is the true map, drawn at random from the Gudmundson data set, $\hat{\Psi}$ is the estimated map, and $\mathbb{E}\{\cdot\}$ denotes expectation over maps, noise, and sensor locations. From Fig. C.5, the proposed scheme performs approximately a 20 % better than the next competing alternative. The parameters of the competing algorithms were tuned for this specific experiment, so their performance as in Fig. C.5 is optimistic. In practice one must expect a greater performance gap.

C.5 Conclusions

Learning propagation features from data yields spectrum cartography algorithms that require fewer measurements to attain a target performance. Deep neural networks can bring this idea into practice and offer a performance that beats the state-of-the-art. Future work will design more sophisticated network architectures relying on larger data sets.

References

- [1] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.
- [4] N. Patwari and P. Agrawal, “Effects of correlated shadowing: Connectivity, localization, and RF tomography,” in *Int. Conf. Info. Process. Sensor Networks*, St. Louis, MO, Apr. 2008, pp. 82–93.
- [5] D. Romero, Donghoon Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, 2018.
- [6] S. Grimoud, S. B. Jemaa, B. Sayrac, and E. Moulines, “A REM enabled soft frequency reuse scheme,” in *Proc. IEEE Global Commun. Conf.*, Miami, FL, Dec. 2010, pp. 819–823.
- [7] E. Dall’Anese, S.-J. Kim, G. B. Giannakis, and S. Pupolin, “Power control for cognitive radio networks under channel uncertainty,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Aug. 2011.
- [8] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.
- [9] G. Boccolini, G. Hernandez-Penalzoza, and B. Bekerull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.

- [10] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, “Joint link learning and cognitive radio sensing,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.
- [11] S.-J. Kim and G. B. Giannakis, “Cognitive radio spectrum prediction using dictionary learning,” in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [12] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, “Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [13] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [14] M. Hamid and B. Beferull-Lozano, “Non-parametric spectrum cartography using adaptive radial basis functions,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [15] Y. Teganya, D. Romero, L. M. Lopez-Ramos, and B. Beferull-Lozano, “Location-free spectrum cartography,” *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4013–4026, Aug. 2019.
- [16] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, “Group-lasso on splines for spectrum cartography,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [17] J.-A. Bazerque and G. B. Giannakis, “Nonparametric basis pursuit via kernel-based learning,” *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [18] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.
- [19] D. Lee, D. Berberidis, and G. B. Giannakis, “Adaptive Bayesian channel gain cartography,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Calgary, Canada, Apr. 2018, pp. 3555–3558.
- [20] P. Stoica and R. L. Moses, “Spectral analysis of signals,” 2005.
- [21] D. P. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*, John Wiley & Sons, Inc., 2001.
- [22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.
- [23] D. Romero, D. Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, Apr. 2018.

- [24] B. R. Hamilton, X. Ma, R. J. Baxley, and S. M. Matechik, “Propagation modeling for radio frequency tomography in wireless networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 1, pp. 55–65, Feb. 2014.
- [25] J. Fan and T. Chow, “Deep learning based matrix completion,” *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.
- [26] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 2007.
- [27] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Trans. Graphics*, vol. 36, no. 4, pp. 107, Jul. 2017.
- [28] M. C Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of communication systems: modeling, methodology and techniques*, Springer Science & Business Media, 2006.
- [29] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos,” *Pattern Recognition Letters*, vol. 105, pp. 13–22, Apr. 2018.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE Int. Conf. Comput. Vision*, Washington, DC, Dec. 2015, pp. 1026–1034.
- [31] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.
- [32] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electron. Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

Appendix D

Paper D

Title: Deep Completion Autoencoders for Radio Map Estimation

Authors: Yves Teganya and Daniel Romero

Affiliation: Dept. of Information and Communication Technology, University of Agder, P. O. Box 509, NO-4898 Grimstad, Norway

Journal: *IEEE Transactions on Wireless Communications*, Under review

Deep Completion Autoencoders for Radio Map Estimation

Yves Teganya and Daniel Romero

Abstract—Radio maps provide metrics such as power spectral density for every location in a geographic area and find numerous applications such as UAV communications, interference control, spectrum management, resource allocation, and network planning to name a few. Radio maps are constructed from measurements collected by spectrum sensors distributed across space. Since radio maps are complicated functions of the spatial coordinates due to the nature of electromagnetic wave propagation, model-free approaches are strongly motivated. Nevertheless, all existing schemes rely on interpolation algorithms unable to learn from data. In contrast, this paper proposes a novel approach in which the spatial structure of propagation phenomena such as shadowing is learned beforehand from a data set with measurements in other environments. Relative to existing schemes, a significantly smaller number of measurements is therefore required to estimate a map with a prescribed accuracy. As an additional novelty, this is also the first work to estimate radio maps using deep neural networks. Specifically, a deep completion autoencoder architecture is developed to effectively exploit the manifold structure of this class of maps.

Keywords— Radio maps, spectrum cartography, deep learning, completion autoencoders, electromagnetic wave propagation.

D.1 Introduction

Spectrum cartography comprises a collection of techniques to construct radio maps, which provide channel metrics such as received signal power, interference power, power spectral density (PSD), electromagnetic absorption, or channel gain across a geographic area; see e.g. [1–4]. Besides applications related to localization [2, 5] and radio tomography [6, 7], radio maps find a myriad of applications in wireless communications such as network planning, interference coordination, power control, spectrum management, resource allocation, handoff procedure design, dynamic spectrum access, and cognitive radio; see e.g. [8–10]. More recently, radio maps have been widely recognized as an enabling technology for UAV communications because they allow autonomous UAVs to account for communication constraints when planning a mission; see e.g. [11–14].

Radio maps are estimated from measurements acquired by spectrum sensors or mobile devices. Most approaches build upon some interpolation algorithm. For example, power maps have been constructed through kriging [1, 15–17], compressive sensing [3], dictionary learning [18, 19], matrix completion [20], Bayesian models [21], radial basis functions [22, 23], and kernel methods [24]. PSD map estimators have been developed using sparse learning [2], thin-plate spline regression [25], kernel-based learning [10, 26],

and tensor completion [27, 28]. Related approaches have been adopted in [7, 29–31] to propose channel-gain map estimators.

Unfortunately, none of the existing approaches can learn from data, which suggests that their estimation performance can be significantly improved along this direction.¹

To this end, the first contribution of this work is a data-driven paradigm for radio map estimation. The idea is to learn the spatial structure of relevant propagation phenomena such as shadowing, reflection, and diffraction using a data set of past measurements in different environments. Intuitively, learning how these phenomena evolve across space can significantly reduce the number of measurements required to achieve a given estimation accuracy. This is a critical aspect since the time interval in which measurements are collected needs to be sufficiently short relative to the temporal variations of the target map in real-world scenarios (coherence time). The second contribution comprises a PSD map estimation algorithm based on a deep neural network. To cope with the variable number of measurements, a tensor completion task is formulated based on a spatial discretization of the area of interest and addressed by means of a *completion network* with an encoder-decoder architecture. This structure is motivated by the observation that radio maps lie close to a low-dimensional manifold embedded in a high-dimensional space. Extensive experiments using a realistic data set obtained with Remcom’s Wireless InSite simulator reveal that the proposed algorithm markedly outperforms state-of-the-art radio map estimators. This data set will be posted along with the code and trained neural networks at the authors’ web sites.

The novelty of this work is twofold: (i) it is the first to propose data-driven spectrum cartography; (ii) it is the first to propose a deep learning algorithm for radio map estimation.

The rest of this paper is organized as follows. Sec. D.2 formulates the problem of PSD map estimation. Sec. D.3 introduces the aforementioned data-driven radio map estimation paradigm and proposes a deep neural network architecture based on completion autoencoders. Simulations and conclusions are respectively provided in Secs. D.4 and D.5.

Notation: $|\mathcal{A}|$ denotes the cardinality of set \mathcal{A} . Bold uppercase (lowercase) letters denote matrices or tensors (column vectors), $[\mathbf{a}]_i$ is the i -th entry of vector \mathbf{a} , $[\mathbf{A}]_{i,j}$ is the (i, j) -th entry of matrix \mathbf{A} , and $[\mathbf{B}]_{i,j,k}$ is the (i, j, k) -th entry of tensor \mathbf{B} . Finally, \mathbf{A}^\top is the transpose of matrix \mathbf{A} .

D.2 PSD map Estimation Problem

This section formulates the problem of *PSD map* estimation. The problem where *power maps* must be estimated can be recovered as a special case of *PSD map* estimation in a single frequency.

¹The conference version of this work [32] presents the core ideas here. Relative to [32], the present paper contains improved neural network architectures (including fully convolutional networks), a methodology for PSD estimation with basis expansion models, and extensive empirical validation and comparison with existing algorithms through a realistic data set. Some of the ideas in [33] are similar to those in [32]. However, since the former was submitted after our paper [32], it is regarded as parallel work.

Consider L transmitters, or sources, located in a geographic region of interest $\mathcal{X} \subset \mathbb{R}^2$ and operating in a certain frequency band. Let $\Upsilon_l(f)$ denote the transmit PSD of the l -th source and let $H_l(\mathbf{x}, f)$ represent the frequency response of the channel between the l -th source and a receiver with an isotropic antenna at location $\mathbf{x} \in \mathcal{X}$. Both $\Upsilon_l(f)$ and $H_l(\mathbf{x}, f)$ are assumed to remain constant over time; see Remark 6.

If the L signals are uncorrelated, the PSD at $\mathbf{x} \in \mathcal{X}$ is

$$\Psi(\mathbf{x}, f) = \sum_{l=1}^L \Upsilon_l(f) |H_l(\mathbf{x}, f)|^2 + v(\mathbf{x}, f), \quad (\text{D.1})$$

where $v(\mathbf{x}, f)$ models thermal noise, background radiation noise, and interference from remote sources. A certain number of devices with sensing capabilities, e.g. user terminals in a cellular network, collect PSD measurements $\{\tilde{\Psi}(\mathbf{x}_n, f)\}_{n=1}^N$ at N locations $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$ and at a finite set of frequencies $f \in \mathcal{F}$; see also Remark 6. These frequency measurements can be obtained using e.g. periodograms or spectral analysis methods such as the Bartlett or Welch method [34].

These measurements are sent to a fusion center, which may be e.g. a base station, a mobile user, or a cloud server, depending on the application. Given $\{(\mathbf{x}_n, \tilde{\Psi}(\mathbf{x}_n, f)), n = 1, \dots, N, f \in \mathcal{F}\}$, the problem that the fusion center needs to solve is to find an estimate $\hat{\Psi}(\mathbf{x}, f)$ of $\Psi(\mathbf{x}, f)$ at every location $\mathbf{x} \in \mathcal{X}$ and frequency $f \in \mathcal{F}$. Function $\Psi(\mathbf{x}, f)$ is typically referred to as the *true map*, whereas $\hat{\Psi}(\mathbf{x}, f)$ is the *map estimate*. An algorithm that produces $\hat{\Psi}(\mathbf{x}, f)$ is termed *map estimator*.

A natural error metric is the energy $\sum_f \int_{\mathcal{X}} |\Psi(\mathbf{x}, f) - \hat{\Psi}(\mathbf{x}, f)|^2 d\mathbf{x}$. One can quantify the performance of a map estimator in terms of the expectation of this error for a given N or, equivalently, in terms of the minimum N required to guarantee that the expected error is below a prescribed bound.

Remark 5 *The channel $H_l(\mathbf{x}, f)$ is usually decomposed into path loss, shadowing, and fast fading components. Whereas path loss and shadowing typically vary in a scale of meters, fast fading changes in a scale comparable to the wavelength. Since contemporary wireless communication systems utilize wavelengths in the order of millimeters or centimeters, estimating this fast fading component would require knowing the sensor locations $\{\mathbf{x}_n\}_n$ with an accuracy in the order of millimeters, which is not possible e.g. with current global navigation satellite systems (GNSSs). Thus, it is customary to assume that the effects of fast fading have been averaged out and, hence, $H_l(\mathbf{x}, f)$ captures only path loss and shadowing. This is especially well-motivated in scenarios where sensors acquire measurements while moving.*

Remark 6 *$\Upsilon_l(f)$ and $H_l(\mathbf{x}, f)$ can be assumed constant over time so long as the measurements are collected within an interval of shorter length than the channel coherence time and time scale of changes in $\Upsilon_l(f)$. The latter is highly dependent on the specific application. For example, one expects that significant variations in DVB-T bands occur in a scale of several months, whereas $\Upsilon_l(f)$ in LTE bands may change in a scale of milliseconds. In any case, a sensor that moves may collect multiple measurements over this interval where $\Upsilon_l(f)$ and $H_l(\mathbf{x}, f)$ remain approximately constant, which could render the number of measurements significantly larger than the number of sensors.*

D.3 Data-Driven Radio Map Estimation

All existing map estimators rely on interpolation algorithms that do not learn from data. However, it seems natural that an algorithm can be trained to learn how to solve the problem in Sec. D.2 using a record of past measurements, possibly in other geographic regions. Specifically, besides $\mathcal{D}:=\{(\mathbf{x}_n, \tilde{\Psi}(\mathbf{x}_n, f)), \mathbf{x}_n \in \mathcal{X}, f \in \mathcal{F}, n = 1, \dots, N\}$, a number of measurement records of the form $\mathcal{D}_t:=\{(\mathbf{x}_{nt}, \tilde{\Psi}_t(\mathbf{x}_{nt}, f)), \mathbf{x}_{nt} \in \mathcal{X}_t, f \in \mathcal{F}, n = 1, \dots, N_t\}$, $t = 1, \dots, T$, may be available, where \mathcal{D}_t contains N_t measurements collected in the geographic area \mathcal{X}_t ; see Sec. D.3.5. With this additional data, a better performance is expected when estimating $\Psi(\mathbf{x}, f)$.

The rest of this section develops deep learning estimators that address this data-aided formulation. To this end, Sec. D.3.1 starts by reformulating the problem at hand as a tensor completion task amenable to application of deep neural networks. Subsequently, Sec. D.3.2 addresses unique aspects of tensor/matrix completion via deep learning. Sec. D.3.3 discusses how to exploit structure in the frequency domain. Finally, Secs. D.3.4 and D.3.5 respectively describe how to learn the spatial structure of propagation phenomena via the notion of *completion autoencoders* and how these networks can be trained in real-world scenarios.

D.3.1 Map Estimation as a Tensor Completion Task

Observe that N and N_t depend on the number and movement of the sensors relative to the time-scale of temporal variations in $\Psi(\mathbf{x}, f)$ and $\Psi_t(\mathbf{x}, f)$, respectively; cf. Remark 6. In principle, one could think of using a separate map estimator for each possible value of N . Each estimator could be relatively simple since it would always take the same number of inputs. However, such an approach would be highly inefficient in terms of memory, computation, and prone to erratic behavior since each estimator would have different parameters or be trained with a different data set. Thus, it is more practical to rely on a single estimator that can accommodate an arbitrary number of measurements.

Given their well-documented merits in a number of tasks, deep neural networks constitute a sensible framework to develop radio map estimators. However, regular feedforward neural networks cannot directly accommodate inputs of variable size. To bypass this difficulty, the approach pursued here relies on a spatial discretization amenable to application of feedforward architectures [35, Ch. 6]. Similar discretizations have been applied in [20, 27, 36, 37].

To introduce the appropriate notation, this discretization is briefly outlined for \mathcal{D} ; the extension to \mathcal{D}_t follows the same lines. Define an $N_y \times N_x$ rectangular grid over \mathcal{X} , as depicted in Fig. D.1. This grid comprises points $\boldsymbol{\xi}_{i,j}$ evenly spaced by Δ_x and Δ_y along the x - and y -axes respectively, that is, the (i, j) -th grid point is given by $\boldsymbol{\xi}_{i,j} := [j\Delta_x, i\Delta_y]^\top$, with $i = 1, \dots, N_y$, $j = 1, \dots, N_x$. For future usage, define $\mathcal{A}_{i,j} \subset \{1, \dots, N\}$ as the set containing the indices of the measurement locations assigned to the (i, j) -th grid point by the criterion of minimum distance, i.e., $n \in \mathcal{A}_{i,j}$ iff $\|\boldsymbol{\xi}_{i,j} - \mathbf{x}_n\| \leq \|\boldsymbol{\xi}_{i',j'} - \mathbf{x}_n\| \forall i', j'$.

This grid induces a discretization of $\Psi(\mathbf{x}, f)$ along the \mathbf{x} variable. One can therefore collect the true PSD values at the grid points in matrix $\boldsymbol{\Psi}(f) \in \mathbb{R}^{N_y \times N_x}$, $f \in \mathcal{F}$, whose

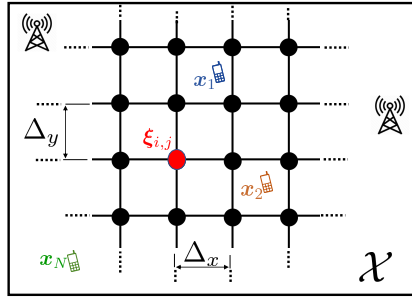


Figure D.1: Model setup and area discretization.

(i, j) -th entry is given by $[\Psi(f)]_{i,j} = \Psi(\xi_{i,j}, f)$. By letting $\mathcal{F} = \{f_1, \dots, f_{N_f}\}$, it is also possible to concatenate these matrices to form the tensor $\Psi \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\Psi]_{i,j,n_f} = \Psi(\xi_{i,j}, f_{n_f})$, $n_f = 1, \dots, N_f$. For short, the term *true map* will either refer to $\Psi(\mathbf{x}, f)$ or Ψ .

Similarly, one can collect the measurements in a tensor of the same dimensions. Informally, if the grid is sufficiently fine (Δ_x and Δ_y are sufficiently small), it holds that $\mathbf{x}_n \approx \xi_{i,j} \forall n \in \mathcal{A}_{i,j}$ and, correspondingly, $\Psi(\mathbf{x}_n, f) \approx \Psi(\xi_{i,j}, f) \forall n \in \mathcal{A}_{i,j}$. It follows that,

$$\Psi(\xi_{i,j}, f) \approx \frac{1}{|\mathcal{A}_{i,j}|} \sum_{n \in \mathcal{A}_{i,j}} \Psi(\mathbf{x}_n, f)$$

whenever $|\mathcal{A}_{i,j}| > 0$. Therefore, it makes sense to aggregate the measurements assigned to $\xi_{i,j}$ as²

$$\tilde{\Psi}(\xi_{i,j}, f) := \frac{1}{|\mathcal{A}_{i,j}|} \sum_{n \in \mathcal{A}_{i,j}} \tilde{\Psi}(\mathbf{x}_n, f).$$

Conversely, when $|\mathcal{A}_{i,j}| = 0$, there are no measurements associated with $\xi_{i,j}$, in which case one says that there is a *miss* at $\xi_{i,j}$. Upon letting $\Omega \subset \{1, \dots, N_y\} \times \{1, \dots, N_x\}$ be such that $(i, j) \in \Omega$ iff $|\mathcal{A}_{i,j}| > 0$, all aggregated measurements $\tilde{\Psi}(\xi_{i,j}, f)$ can be collected into $\tilde{\Psi}(f) \in \mathbb{R}^{N_y \times N_x}$, defined as

$$[\tilde{\Psi}(f)]_{i,j} = \begin{cases} \tilde{\Psi}(\xi_{i,j}, f) & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

Note that misses have been filled with zeroes, but other values could have been used.

When $(i, j) \in \Omega$, the values of $[\tilde{\Psi}(f)]_{i,j}$ and $[\Psi(f)]_{i,j}$ differ due to the error introduced by the spatial discretization as well as due to the measurement error incurred when measuring $\Psi(\mathbf{x}_n, f)$, $n \in \mathcal{A}_{i,j}$. The latter is caused mainly by the finite time devoted by sensors to take measurements, their movement, localization errors, and possible variations of $\Psi(\mathbf{x}_n, f)$ over time.

As before, the matrices $\tilde{\Psi}(f)$, $f = 1, \dots, N_f$ can be concatenated to form $\tilde{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\tilde{\Psi}]_{i,j,n_f} = [\tilde{\Psi}(f_{n_f})]_{i,j}$. For short, this tensor will be referred to as the *sampled map*. With this notation, the cartography problem stated in Sec. D.2 will be approximated as estimating Ψ given Ω and $\tilde{\Psi}$.

²For simplicity, the notation implicitly assumes that $\mathbf{x}_n \neq \xi_{i,j} \forall n, i, j$, but this is not a requirement.

D.3.2 Completion Networks for Radio Map Estimation

The data in the problem formulation at the end of Sec. D.3.1 cannot be handled by plain feedforward neural networks since they cannot directly accommodate input misses and set-valued inputs like Ω . This section explores how to bypass this difficulty.

But before that, a swift refresh on deep learning is in order. A feedforward deep neural network is a function $p_{\mathbf{w}}$ that can be expressed as the composition $p_{\mathbf{w}}(\Phi) = p_{\mathbf{w}_L}^{(L)}(p_{\mathbf{w}_{L-1}}^{(L-1)}(\dots p_{\mathbf{w}_1}^{(1)}(\Phi)))$ of *layer* functions $p_{\mathbf{w}_l}^{(l)}$, where Φ is the input. Although there is no commonly agreed definition of layer function, it is typically formed by concatenating simple scalar-valued functions termed *neurons* that implement a linear function followed by a non-linear function known as activation [35]. The term *neuron* stems from the resemblance between these functions and certain simple functional models for natural neurons. Similarly, there is no general agreement on which values of L qualify for $p_{\mathbf{w}}$ to be regarded a *deep* neural network, but in practice L may range from tens to thousands. With vector \mathbf{w}_l containing the parameters, or weights, of the l -th layer, the parameters of the entire network can be stacked as $\mathbf{w} := [\mathbf{w}_1^\top, \dots, \mathbf{w}_L^\top]^\top \in \mathbb{R}^{N_w}$. These parameters are *learned* using a *training set* in a process termed *training*.

The rest of this section designs $p_{\mathbf{w}_1}^{(1)}$ to cope with missing data, whereas Secs. D.3.3 and D.3.4 will address the design of the other layers. Throughout, the *training examples* will be represented by $\{(\tilde{\Psi}_t, \Omega_t)\}_{t=1}^T$, where $\tilde{\Psi}_t$ and Ω_t are obtained from \mathcal{D}_t by applying the procedure described in Sec. D.3.1.

The desired estimator should obtain Ψ as a function of $\tilde{\Psi}$ and Ω . Following one of the early approaches to accommodate misses in deep neural networks [38], one could think of solving

$$\begin{aligned} \underset{\{\chi_t\}_t, \mathbf{w}}{\text{minimize}} \quad & \frac{1}{T+1} \sum_{t=1}^{T+1} \|\mathcal{P}_{\Omega_t}(\chi_t - p_{\mathbf{w}}(\chi_t))\|_F^2, \\ & [\chi_t]_{i,j,n_f} = [\tilde{\Psi}_t]_{i,j,n_f} \quad \forall n_f, \forall (i,j) \in \Omega_t, \end{aligned} \quad (\text{D.2})$$

where $\tilde{\Psi}_{T+1} := \tilde{\Psi}$ and $\Omega_{T+1} := \Omega$ correspond to the *test measurements*, $\|\mathbf{A}\|_F^2 := \sum_{i,j,n_f} [\mathbf{A}]_{i,j,n_f}^2$ is the squared Frobenius norm of tensor \mathbf{A} , and $\mathcal{P}_{\Omega}(\mathbf{A})$ is defined by $[\mathcal{P}_{\Omega}(\mathbf{A})]_{i,j,n_f} = [\mathbf{A}]_{i,j,n_f}$ if $(i,j) \in \Omega$ and $[\mathcal{P}_{\Omega}(\mathbf{A})]_{i,j,n_f} = 0$ otherwise. The map estimate produced by (D.2) method would be $\hat{\Psi} := \chi_{T+1}$.

In (D.2) and other expressions throughout, regularizers and capacity-control constraints are not explicitly shown to simplify the notation, but they can be readily accommodated.

The number of optimization variables in (D.2) is $N_w + N_x N_y N_f (T+1)$, where N_w is the length of \mathbf{w} . Unfortunately, this number is prohibitive for a large T , as required for training deep neural networks. Besides, a large number of forward and backward backpropagation passes [35, Ch. 6] are required to estimate each map. Thus, this approach is not suitable for real-time implementation, as usually required in radio map estimation.

To alleviate this limitation, a simple alternative would be to directly feed $\tilde{\Psi}$ to the neural network and train by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t - p_{\mathbf{w}}(\tilde{\Psi}_t) \right) \right\|_F^2. \quad (\text{D.3})$$

Although the missing entries were filled with zeros in Sec. D.3.1, one can alternatively use other real numbers. After (D.3) is solved, $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\tilde{\Psi})$, which requires a single forward pass. Besides, solving (D.3) involves just N_w optimization variables.

However, because the completion $p_{\mathbf{w}}(\tilde{\Psi})$ does not account for Ω , poor performance is expected since the network cannot distinguish missing entries from measurements close to the filling value. In the application at hand, one could circumvent this limitation by expressing the entries of $\tilde{\Psi}$ in natural power units (e.g. Watt) and filling the misses with a negative number such as -1. Unfortunately, the usage of finite-precision arithmetic would introduce large errors in the map estimates and is problematic in our experience. For this reason, expressing $\tilde{\Psi}$ in logarithmic units such as dBm is preferable. However, in that case, filling misses with negative numbers would not solve the aforementioned difficulty since logarithmic units are not confined to take non-negative values. Hence, a preferable alternative is to complement the input map with a binary mask that indicates which entries are observed, as proposed in the image inpainting literature [39]. Specifically, a mask $\mathbf{M}_{\Omega} \in \{0, 1\}^{N_y \times N_x}$ can be used to represent Ω by setting $[\mathbf{M}_{\Omega}]_{i,j} = 1$ if $(i, j) \in \Omega$ and $[\mathbf{M}_{\Omega}]_{i,j} = 0$ otherwise. To simplify notation, let $\check{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f + 1}$ denote a tensor obtained by concatenating $\tilde{\Psi}$ and \mathbf{M}_{Ω} along the third dimension. The neural network can therefore be trained as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \left\| \mathcal{P}_{\Omega_t} \left(\check{\Psi}_t - p_{\mathbf{w}}(\check{\Psi}_t) \right) \right\|_F^2, \quad (\text{D.4})$$

and, afterwards, a tensor $\tilde{\Psi}$ can be completed just by evaluating $p_{\mathbf{w}}(\check{\Psi})$. Then, this scheme is simple to train, inexpensive to test, and exploits information about the location of the misses.

Remark 7 *The introduction of a binary mask to indicate the sensor locations suggests an approach to accommodate additional side information that may assist in map estimation. For example, one can append an additional mask $\mathbf{M}' \in \mathbb{R}^{N_y \times N_x}$ where $[\mathbf{M}']_{i,j}$ indicates, for instance, the height of obstacles such as buildings at $\xi_{i,j}$ or the kind of propagation terrain (e.g. urban, suburban, etc) where $\xi_{i,j}$ lies. In this case, tensor $\check{\Psi} \in \mathbb{R}^{N_y \times N_x \times N_f + 1}$ can be replaced with its augmented version of size $N_y \times N_x \times (N_f + N_m)$ obtained by concatenating $N_m - 1$ such masks to $\tilde{\Psi}$ and \mathbf{M}_{Ω} . Another (possibly complementary) approach is to combine multiple masks into a single matrix. For example, suppose that all measurements are taken outdoors and let $\mathcal{B} \subset \{1, \dots, N_y\} \times \{1, \dots, N_x\}$ be such that $(i, j) \in \mathcal{B}$ iff $\xi_{i,j}$ is inside a building. Then, the information in \mathcal{B} and Ω can be combined into $\mathbf{M}_{\Omega, \mathcal{B}} \in \{0, 1, -1\}^{N_y \times N_x}$, where $[\mathbf{M}_{\Omega, \mathcal{B}}]_{i,j} = 1$ if $(i, j) \in \Omega$, $[\mathbf{M}_{\Omega, \mathcal{B}}]_{i,j} = -1$ if $(i, j) \in \mathcal{B}$, and $[\mathbf{M}_{\Omega, \mathcal{B}}]_{i,j} = 0$ otherwise. Masks of this kind can be similarly concatenated to $\tilde{\Psi}$ to form an augmented tensor $\check{\Psi}$. The rest of the paper will use symbol $\check{\Psi}$ to refer to the result of concatenating $\tilde{\Psi}$ with the available masks.*

Remark 8 *The proposed deep learning framework offers an additional advantage: the tensors in the objective functions throughout (e.g. (D.3), (D.4)) can be expressed in dB units. This is not possible in most existing approaches, which rely on convex solvers. Consequently, existing algorithms would focus on fitting large power values and will neglect*

errors at those locations with low power values. Given its greater practical significance, it will be assumed throughout that all tensors are expressed in dB units before evaluating the Frobenius norms.

D.3.3 Exploiting Structure in the Frequency Domain

In practice, different degrees of prior information may be available when estimating a PSD map. Sec. D.3.3.1 will address the scenario in which no such information is available, whereas Sec. D.3.3.2 will develop an output layer that exploits a common form of prior information available in real-world applications.

D.3.3.1 No Prior Information

It will be first argued that the plain training approach in (D.4) is likely to be ill-posed in practical scenarios when the network does not enforce or exploit any structure in the frequency domain. To see this, suppose that the number of frequencies N_f in \mathcal{F} is significant, e.g. 512 or 1024 as would typically occur in practice, and consider a fully connected first layer $p_{\mathbf{w}_1}^{(1)}$ with N_N neurons. Its total number of parameters becomes $(N_y N_x (N_f + N_m) + 1) N_N$ plus possibly additional parameters of the activation functions. Other layers will experience the same issue to different extents. Since T must be comparable to the number of parameters to train a network effectively, a large N_f would drastically limit the number of layers or neurons that can be used for a given T .

In this case, it may be preferable to separate the problem across frequencies by noting that propagation effects at similar frequencies are expected to be similar. Building upon this principle, $p_{\mathbf{w}}$ can operate separately at each frequency f . This means that training can be accomplished through

$$\underset{\mathbf{w}}{\text{minimize}} \frac{1}{TN_f} \sum_{t=1}^T \sum_{f \in \mathcal{F}} \left\| \mathcal{P}_{\Omega_t} \left(\tilde{\Psi}_t(f) - p_{\mathbf{w}}(\tilde{\Psi}_t(f)) \right) \right\|_F^2, \quad (\text{D.5})$$

where the input $\tilde{\Psi}_t(f) \in \mathbb{R}^{N_y \times N_x \times (1+N_m)}$ is formed by concatenating $\tilde{\Psi}_t(f)$ and N_m masks; see Remark 7.

Observe that the number of variables is roughly reduced by a factor of N_f , whereas the “effective” number of training examples has been multiplied by N_f ; cf. number of summands in (D.5). This is a drastic improvement especially for moderate values of N_f . Thus, such a frequency separation allows an increase in the number of neurons per layer or (typically more useful [35, Ch. 5]) the total number of layers for a given T . Although such a network would not exploit structure across the frequency domain, the fact that it would be better trained is likely to counteract this limitation in many setups.

D.3.3.2 Output Layers for Parametric PSD Expansions

Real-world communication systems typically adhere to standards that specify transmission masks by means of carrier frequencies, channel bandwidth, roll-off factors, number of OFDM subcarriers, guard bands, location and power of pilot subcarriers, and so on.

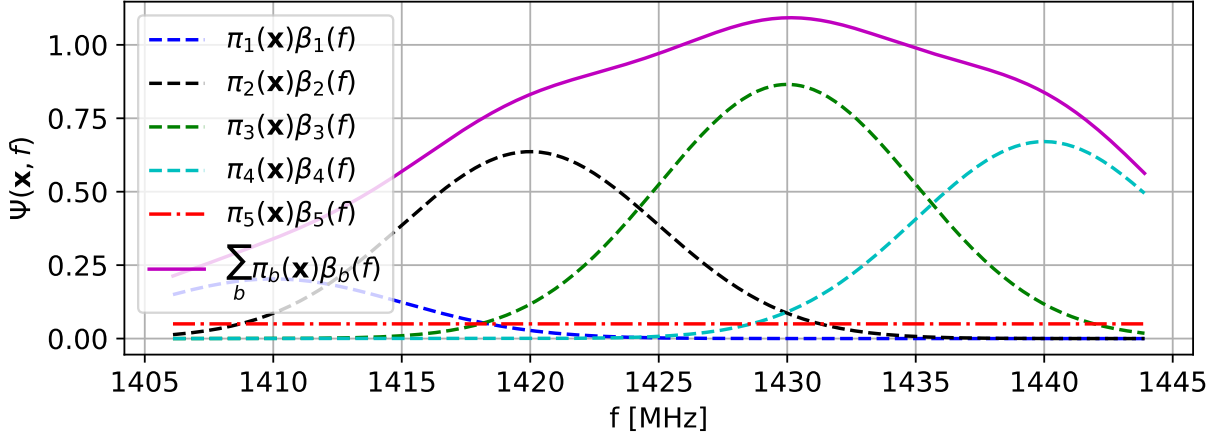


Figure D.2: PSD $\Psi(\mathbf{x}, f)$ at location \mathbf{x} using a basis expansion model with Gaussian functions.

It seems, therefore, reasonable to capitalize on such prior information for radio map estimation by means of a basis expansion model in the frequency domain like the one in [10, 40, 41]. Even when the frequency form of the transmit PSD is unknown, a basis expansion model is also motivated due to its capacity to approximate any PSD to some extent; e.g. [2, 25].

Under a basis expansion model, the transmit PSD of each source is expressed as

$$\Upsilon_l(f) = \sum_{b=1}^{B-1} \pi_{lb} \beta_b(f), \quad (\text{D.6})$$

where π_{lb} denotes the expansion coefficients and $\{\beta_b(f)\}_{b=1}^{B-1}$ is a collection of $B-1$ given basis functions such as raised-cosine or Gaussian functions. Without loss of generality, the basis functions are normalized so that $\int_{-\infty}^{\infty} \beta_b(f) df = 1$. In this way, if $\beta_b(f)$ is the PSD of the b -th channel, as possibly specified by a standard, then π_{lb} denotes the power transmitted by the l -th source in the b -th channel. Substituting (D.6) into (D.1), the PSD at $\mathbf{x} \in \mathcal{X}$ reads as

$$\Psi(\mathbf{x}, f) = \sum_{l=1}^L \sum_{b=1}^{B-1} \pi_{lb} \beta_b(f) |H_l(\mathbf{x}, f)|^2 + v(\mathbf{x}, f).$$

Now assume that $|H_l(\mathbf{x}, f)|^2$ remains approximately constant over the support of each basis function, i.e., $|H_l(\mathbf{x}, f)|^2 \approx |H_{lb}(\mathbf{x})|^2$ for all f in the support of $\beta_b(f)$. This is a reasonable assumption for narrowband $\beta_b(f)$; if it does not hold, one can always split $\beta_b(f)$ into multiple basis functions with a smaller frequency support until the assumption holds. Then, the PSD at \mathbf{x} can be written as

$$\Psi(\mathbf{x}, f) = \sum_{b=1}^{B-1} \pi_b(\mathbf{x}) \beta_b(f) + v(\mathbf{x}, f), \quad (\text{D.7})$$

where $\pi_b(\mathbf{x}) := \sum_{l=1}^L \pi_{lb} |H_{lb}(\mathbf{x})|^2$. If $\beta_b(f)$ models the transmit PSD of the b -th channel, then $\pi_b(\mathbf{x})$ corresponds to the power of the b -th channel at \mathbf{x} .

Observe that the noise PSD $v(\mathbf{x}, f)$ can be similarly expressed in terms of a basis expansion. To simplify the exposition, suppose that $v(\mathbf{x}, f)$ is expanded with a single

term as $v(\mathbf{x}, f) \approx \pi_B(\mathbf{x})\beta_B(f)$, which in turn implies that (D.7) becomes

$$\Psi(\mathbf{x}, f) = \sum_{b=1}^B \pi_b(\mathbf{x})\beta_b(f). \quad (\text{D.8})$$

Fig. D.2 illustrates this expansion for $B = 5$ when $\{\beta_b(f)\}_{b=1}^4$ are Gaussian radial basis functions and $\beta_5(f)$ is set to be constant to model the PSD of white noise. Note that the adopted basis expansion furthermore allows estimation of the noise power $\pi_B(\mathbf{x})$ at every location, thereby solving a fundamental problem in applications such as cognitive radio [42].

With the above expansion, the tensor $\Psi \in \mathbb{R}^{N_y \times N_x \times N_f}$ introduced in Sec. D.3.1 can be expressed as $[\Psi]_{i,j,n_f} = \sum_{b=1}^B \Pi_{i,j,b} \beta_b(f_{n_f})$, where tensor $\Pi \in \mathbb{R}^{N_y \times N_x \times B}$ contains the coefficients $[\Pi]_{i,j,b} = \pi_b(\xi_{i,j})$. In a deep neural network, this structure can be naturally enforced by setting all but the last layer to obtain an estimate $\hat{\Pi}$ of Π and the last layer to produce $\hat{\Psi}$. Specifically, the neural network can be expressed schematically as:

$$\begin{array}{ccc} \mathcal{L} & \xrightarrow{\bar{p}_{\mathbf{w}}} & \mathbb{R}^{N_y \times N_x \times B} & \xrightarrow{p^{(L)}} & \mathbb{R}^{N_y \times N_x \times N_f} \\ \check{\Psi} & \longrightarrow & \hat{\Pi} & \longrightarrow & \hat{\Psi}, \end{array}$$

where $\mathcal{L} \subset \mathbb{R}^{N_y \times N_x \times (N_f + N_m)}$ is the input space, function $\bar{p}_{\mathbf{w}}(\check{\Psi}) := p_{\mathbf{w}_{L-1}}^{(L-1)}(\dots p_{\mathbf{w}_1}^{(1)}(\check{\Psi}))$ groups the first $L - 1$ layers, and $p^{(L)}$ denotes the last layer. With this notation, $\hat{\Pi} = \bar{p}_{\mathbf{w}}(\check{\Psi})$ and $\hat{\Psi} = p^{(L)}(\hat{\Pi}) \in \mathbb{R}^{N_y \times N_x \times N_f}$, where $[\hat{\Psi}]_{i,j,n_f} = \sum_{b=1}^B \hat{\Pi}_{i,j,b} \beta_b(f_{n_f})$. Observe that, as reflected by the notation, the last layer $p^{(L)}$ does not involve trainable parameters. Furthermore, notice that the number of neurons in the last trainable layer has been reduced from $N_y N_x N_f$ to $N_y N_x B$. This entails a significant reduction in the number of parameters of the network and, as discussed in Sec. D.3.3.1, contributes to improve estimation performance for a given T .

D.3.4 Deep Completion Autoencoders

The previous section addressed design aspects pertaining to the map structure in the frequency domain. In contrast, this section deals with structure across space. In particular, a deep neural network architecture based on *convolutional autoencoders* [43] will be developed.

A (conventional) autoencoder [35, Ch. 12] is a neural network $p_{\mathbf{w}}$ that can be expressed as the composition of a function $\epsilon_{\mathbf{w}}$ termed *encoder* and a function $\delta_{\mathbf{w}}$ called *decoder*, i.e., $p_{\mathbf{w}}(\Phi) = \delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\Phi)) \forall \Phi$. The output of the encoder $\lambda := \epsilon_{\mathbf{w}}(\Phi) \in \mathbb{R}^{N_\lambda}$ is referred to as the *code* or vector of *latent variables* and is of a typically much lower dimension than the input Φ . An autoencoder is trained so that $\delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\Phi)) \approx \Phi \forall \Phi$, which forces the encoder to compress the information in Φ into the N_λ variables in λ .

A *completion* autoencoder adheres to the same principles as conventional autoencoders except for the fact that the encoder must determine the latent variables from a subset of the entries of the input. If a mask is used, then $\Phi \approx \delta_{\mathbf{w}}(\epsilon_{\mathbf{w}}(\mathcal{P}_\Omega(\Phi), \mathbf{M}_\Omega)) \forall \Phi$ if the sampling set Ω preserves sufficient information for reconstruction – if Ω does not satisfy

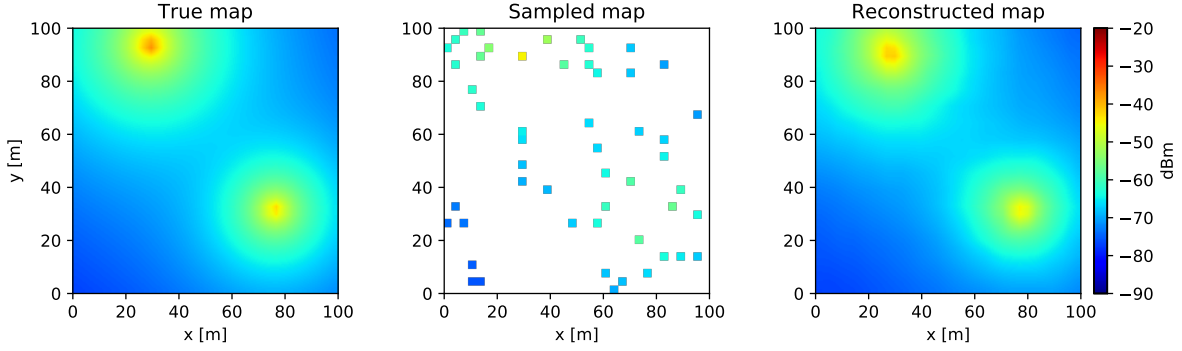


Figure D.3: Estimation with $N_\lambda = 4$ latent variables: (left) true map, (middle) sampled map portraying grid points $\{\xi_{i,j}\}$ with $|\mathcal{A}_{i,j}| > 0$, and (right) map estimate.

this requirement, then reconstructing Φ is impossible regardless of the technique used. In the application at hand and with the notation introduced in previous sections, the above expression becomes $\mathcal{P}_\Omega(\tilde{\Psi}) \approx \mathcal{P}_\Omega(\delta_w(\epsilon_w(\tilde{\Psi})))$.

As indicated earlier, autoencoders are useful only when most of the information in the input can be condensed in N_λ variables, i.e., when the possible inputs lie close to a manifold of dimension N_λ . To see that this is indeed the case in radio map estimation, an illustrating toy example is presented next. Consider two sources transmitting with a different but fixed power at arbitrary positions in \mathcal{X} and suppose that propagation occurs in free space. All possible spectrum maps in this setup can therefore be uniquely identified by $N_\lambda = 4$ quantities, namely the x and y coordinates of the two sources. Fig. D.3 illustrates this effect, where the left panel of Fig. D.3 depicts a true map Ψ and the right panel shows its estimate using the proposed completion autoencoder when $N_\lambda = 4$. Although the details about the network and simulation setup are deferred to Sec. D.4, one can already notice at this point the quality of the estimate, which clearly supports the aforementioned manifold hypothesis. In a real-world scenario, there may be more than two sources, their transmit power may not always be the same, and there are shadowing effects, which means that $N_\lambda \geq 4$ will be generally required.

The rest of this section will describe the main aspects of the architecture developed in this work and summarized in Fig. D.4. The main design decisions are supported here by arguments and intuition. Empirical support is provided in Sec. D.4.2.

The encoder mainly comprises convolutional and pooling layers. The motivation for convolutional layers is three-fold: (i) relative to fully connected layers, they severely reduce the number of parameters to train and, consequently, the amount of data required. Despite this drastic reduction, (ii) convolutional layers are still capable of exploiting the spatial structure of maps and (iii) they result in shift-invariant transfer functions, a desirable property in the application at hand since moving the sources in a certain direction must be corresponded by the same movement in the map estimate. Recall that a convolutional layer with input $\Phi^{(I)}$ and output $\Phi^{(O)}$ linearly combines 2D convolutions as

$$[\Phi^{(O)}]_{i,j,c_{\text{out}}} = \sum_{c_{\text{in}}=1}^{C_{\text{in}}} \sum_{u=-k}^k \sum_{v=-k}^k [\mathbf{F}_{c_{\text{out}}}]_{u,v,c_{\text{in}}} [\Phi^{(I)}]_{i-u,j-v,c_{\text{in}}},$$

where the c_{out} -th *kernel* $\mathbf{F}_{c_{\text{out}}}$ is of size $2k + 1 \times 2k + 1 \times C_{\text{in}}$. Layer indices were omit-

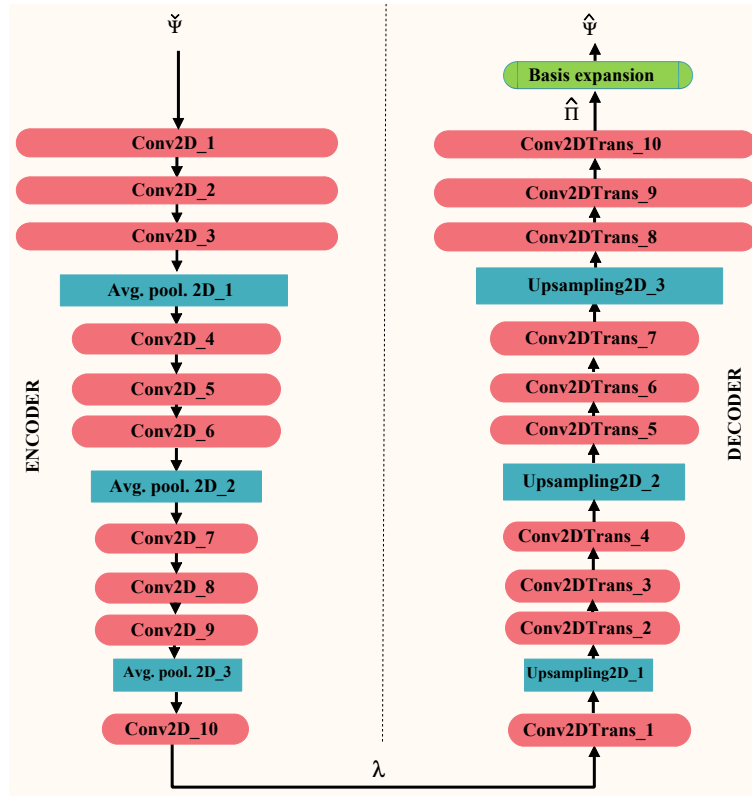


Figure D.4: Autoencoder architecture.

Table D.1: Parameters of the proposed network.

Layers	Parameters
Conv2D/ Conv2DTranspose	Kernel size = 3×3 , stride = 1, activation = PLReLU, 32 filters
AveragePooling2D	Pool size = 2, stride = 2
Upsampling2D	Up-sampling factor = 2, bilinear interpolation

ted in order not to overload notation. The adopted activation functions are parametric *leaky rectified linear units* (PReLU) [44], whose leaky parameter is also trained; see also Sec. D.4.2. *Average pooling* layers are used to down-sample the outputs of convolutional layers, thereby condensing the information gradually in fewer features while approximately preserving shift invariance [35, Ch. 9].

As usual in autoencoders, the decoder follows a “reverse” architecture relative to the encoder. Specifically, for each convolutional layer of the encoder, the decoder has a corresponding *convolution transpose* layer [45], sometimes called “deconvolutional” layer. Likewise, the pooling layers of the encoder are matched with *up-sampling* layers. A simple possibility is to implement such an upsampling operation by means of bilinear interpolation.

Observe that the proposed network, summarized in Fig. D.4 and Table D.1, is *fully convolutional*, which means that there are no fully connected layers. This not only leads to a better estimation performance due to the reduced number of parameters to train (cf.

Sec. D.4), but also enables the possibility of utilizing the same network with any value of N_x and N_y . With fully connected layers, one would generally require a different network for each pair (N_x, N_y) , which would clearly have negative implications for training.

D.3.5 Learning in Real-World Scenarios

A key novelty in this paper is to obtain map estimators by learning from data. This section describes how to construct a suitable training set in the application at hand. Specifically, three approaches are discussed:

D.3.5.1 Synthetic Training Data

Since collecting a large number of training maps may be slow or expensive, one can instead generate maps using a mathematical model or simulator that captures the structure of the propagation phenomena, such as path loss and shadowing; see e.g. [46]. Fitting p_w to data generated by that model would, in principle, yield an estimator that effectively exploits this structure. The idea is, therefore, to generate T maps $\{\Psi_t(\mathbf{x}, f)\}_{t=1}^T$ together with T sampling sets $\{\Omega_t\}_{t=1}^T$. Afterwards, $\{\tilde{\Psi}_t\}_{t=1}^T$ and $\{\check{\Psi}_t\}_{t=1}^T$ can be formed as described earlier. It is possible to add artificially generated noise to the synthetic measurements in $\tilde{\Psi}_t$ to model the effect of measurement error. This would train the network to counteract the impact of such error, along the lines of denoising autoencoders [35, Ch. 14].

The advantage of this approach is that one has access to the ground truth, i.e., one can use the true maps Ψ_t as *targets*. Specifically, the neural network can be trained on the data $\{(\check{\Psi}_t, \Psi_t)\}_{t=1}^T$ by solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{T} \sum_{t=1}^T \|\Psi_t - p_w(\check{\Psi}_t)\|_F^2. \quad (\text{D.9})$$

If the model or simulator is sufficiently close to the reality, completing a real-world map $\check{\Psi}$ as $p_w(\check{\Psi})$ should produce an accurate estimate.

D.3.5.2 Real Training Data

In practice, real maps may be available for training. However, in most cases, it will not be possible to collect measurements at all grid points within a sufficiently short time interval; see Remark 6. Besides, it is not possible to obtain the entries of Ψ but only measurements of it. This means that a real training set comprises tensors $\{\check{\Psi}_t, t = 1, \dots, T\}$ but not Ψ_t .

For training, one can plug this data directly into (D.4) or (D.5). However, p_w may then learn to fit just the observed entries $\{[\check{\Psi}_t(f)]_{i,j}, (i,j) \in \Omega_t\}$, as would happen e.g. when p_w is the identity mapping. To counteract this trend, one can adopt a sufficiently small N_λ . The downside is that estimation performance may be damaged. To bypass this difficulty, the approach proposed here is to use part of the measurements as the input and another part as the output (target). Specifically, for each t , construct Q_t pairs of (not necessarily disjoint) subsets $\Omega_{t,q}^{(I)}, \Omega_{t,q}^{(O)} \subset \Omega_t$, $q = 1, \dots, Q_t$, e.g. by drawing a given number of elements of Ω_t uniformly at random without replacement. Using these subsets,

subsample $\tilde{\Psi}_t$ to yield $\tilde{\Psi}_{t,q}^{(I)} := \mathcal{P}_{\Omega_{t,q}^{(I)}}(\tilde{\Psi}_t)$ and $\tilde{\Psi}_{t,q}^{(O)} := \mathcal{P}_{\Omega_{t,q}^{(O)}}(\tilde{\Psi}_t)$. With the resulting $\sum_t Q_t$ training instances, one can think of solving

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{\sum_t Q_t} \sum_{t=1}^T \sum_{q=1}^{Q_t} \left\| \mathcal{P}_{\Omega_{t,q}^{(O)}} \left(\tilde{\Psi}_{t,q}^{(O)} - p_{\mathbf{w}} \left(\tilde{\Psi}_{t,q}^{(I)} \right) \right) \right\|_F^2, \quad (\text{D.10})$$

where $\tilde{\Psi}_{t,q}^{(I)}$ is formed by concatenating $\tilde{\Psi}_{t,q}^{(I)}$ and $\mathbf{M}_{\Omega_{t,q}^{(I)}}$.

D.3.5.3 Hybrid Training

In practice, one expects to have real data, but only in a limited amount. It then makes sense to apply the notion of *transfer learning* [35, Ch. 15] as follows: first, learn an initial parameter vector \mathbf{w}^* by solving (D.9) with synthetic data. Second, solve (D.10) with real data, but using \mathbf{w}^* as initialization for the optimization algorithm. The impact of choosing this initialization is that the result of solving (D.10) in the second step will be generally closer to a “better” local optimum than if a random initialization were adopted. Hence, this approach combines the information of both synthetic and real data sets.

D.4 Numerical Experiments

This section validates the proposed framework and network architecture through numerical experiments. All code and data sets will be posted at the authors’ websites.

The region of interest \mathcal{X} is a square area of side 100 m, discretized into a grid with $N_y = N_x = 32$. Two data sets are constructed as described next. First, $T = 4 \cdot 10^5$ maps are generated where the two considered transmitters are placed uniformly at random in \mathcal{X} , have height 1.5 m, and transmit with power in each channel drawn uniformly at random between 5 and 11 dBm. The pathloss exponent is set to 3, whereas the gain at unit distance is -30 dB. The lognormal shadowing component adheres to the Gudmundson model [47] with correlation $\text{E} \{ H_l(\mathbf{x}_1, f) H_l(\mathbf{x}_2, f) \} = \sigma_{\text{sh}}^2 0.95^{||\mathbf{x}_1 - \mathbf{x}_2||}$, where $\sigma_{\text{sh}}^2 = 10$ dB² and $||\mathbf{x}_1 - \mathbf{x}_2||$ is the distance between \mathbf{x}_1 and \mathbf{x}_2 in meters. Measurement locations are drawn uniformly at random without replacement across the grid points. Each measurement $\tilde{\Psi}(\mathbf{x}_n, f)$ is obtained by adding zero-mean Gaussian noise with standard deviation 1 dB to $\Psi(\mathbf{x}_n, f)$.

A second data set of $T = 1.25 \cdot 10^5$ maps is generated using Remcom’s Wireless InSite software in the “urban canyon” scenario. Measurement locations are distributed uniformly at random without replacement across the grid points that lie on the streets. To average out multipath fading present in the generated maps (see Remark 5), $\Psi(\boldsymbol{\xi}_{i,j}, f)$ is replaced with $(1/|\mathcal{N}_{i,j}|) \sum_{\boldsymbol{\xi} \in \mathcal{N}_{i,j}} \Psi(\boldsymbol{\xi}, f)$, where $\mathcal{N}_{i,j}$ contains the $|\mathcal{N}_{i,j}| = 9$ grid points that lie closest to $\boldsymbol{\xi}_{i,j}$, including $\boldsymbol{\xi}_{i,j}$. A binary mask indicating the position of buildings is combined with the sample mask as indicated at the end of Remark 7. This simulator accurately captures propagation phenomena through ray tracing algorithms. Thus, this data set can be regarded as a realistic surrogate of a data set with real measurements.

The network proposed in Sec. D.3.4 with code length $N_\lambda = 64$ is implemented in TensorFlow and trained using the ADAM solver with learning rate 10^{-4} . Quantitative

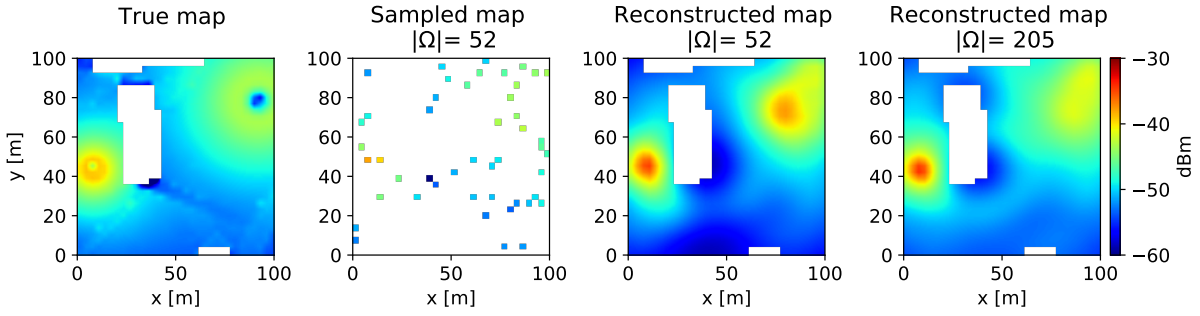


Figure D.5: Power map estimate with the proposed neural network. (left): true map, (center left): sampled map portraying the locations of the grid points $\{\xi_{i,j}\}$ where $|\mathcal{A}_{i,j}| > 0$; (center right) and (right): map estimates. White areas represent buildings.

evaluation will compare the root mean square error (RMSE), defined as:

$$\text{RMSE} = \sqrt{\frac{\mathbb{E}\{\|\Psi - \hat{\Psi}\|_F^2\}}{N_x N_y N_f}}, \quad (\text{D.11})$$

where Ψ is the true map, $\hat{\Psi}$ is the map estimate, and $\mathbb{E}\{\cdot\}$ denotes expectation over maps, noise, and sensor locations.

D.4.1 Power Map Cartography

To analyze the most fundamental radio map estimation aspects, \mathcal{F} is set here to the singleton $\mathcal{F} = \{1400 \text{ MHz}\}$ and the bandwidth to 5 MHz in both data sets. To better observe the impact of propagation phenomena, $v(\mathbf{x}, f)$ is set to 0.

The proposed algorithm is compared against a representative set of competitors, whose parameters were adjusted to approximately yield the best performance. This includes: (i) The kriging algorithm in [1] with regularization parameter 10^{-5} and Gaussian radial basis functions with parameter $\sigma_K := 5\sqrt{\Delta_y N_y \Delta_x N_x / |\Omega|}$, which is approximately 5 times the mean distance between two points at which measurements have been collected. (ii) The multikernel algorithm in [26] with regularization parameter 10^{-4} and 20 Laplacian kernels that use a parameter uniformly spaced between $[0.1\sigma_K, \sigma_K]$. (iii) Matrix completion via nuclear norm minimization [48] with regularization parameter 10^{-5} . As a benchmark, (iv) the K -nearest neighbors (KNN) algorithm with $K = 5$ is also shown.

D.4.1.1 Gudmundson Data Set

Performance is assessed next using the training approach in Sec. D.3.5.1 with $\{(\tilde{\Psi}_t, \Psi_t)\}_{t=1}^T$ given by the Gudmundson data set.

To analyze estimation of real maps when the proposed network is trained over synthetic data, the first experiment shows two map estimates when the true (test) map is drawn from the Wireless Insite data set. Specifically, the first panel of Fig. D.5 depicts the true map, the second shows $\tilde{\Psi}$, and the remaining two panels show estimates using different numbers of measurements. Observe that with just $|\Omega| = 52$ measurements, the estimate is already of a high quality. Note that details due to diffraction, multipath, and antenna

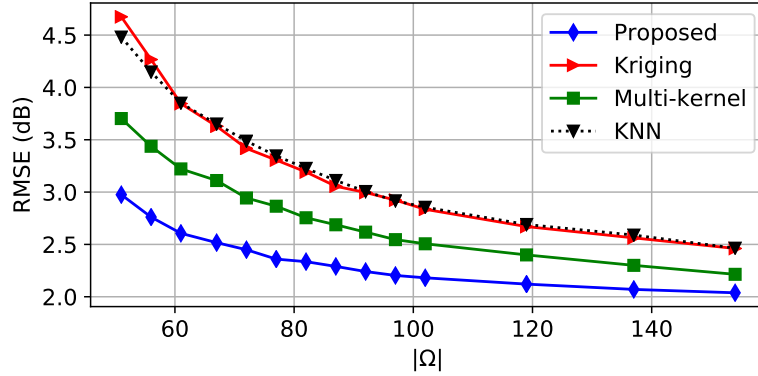


Figure D.6: Comparison with state-of-the-art alternatives. Training and testing maps drawn from the Gudmundson data set.

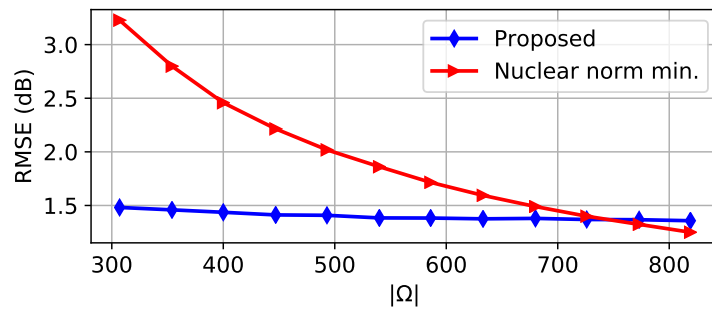


Figure D.7: Performance comparison of the proposed scheme with that of the matrix completion algorithm in [48]. Training and testing maps drawn from the Gudmundson data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$.

directivity are not reconstructed because the Gudmundson data set used to train the network does not capture these effects and, therefore, the network did not learn them.

The second experiment compares the RMSE of the proposed method with that of the competing algorithms. From Fig. D.6, the proposed scheme performs approximately a 25 % better than the next competing alternative. Due to the high RMSE of the matrix completion algorithm in [48] for the adopted range of $|\Omega|$ in Fig. D.6, its RMSE is shown in Fig. D.7 along with that of the proposed algorithm for larger values of $|\Omega|$. The proposed method still outperforms this competitor except when the number of measurements is very large, close to the total number of grid points.

D.4.1.2 Wireless Insite Data Set

To investigate how the proposed network would perform in a real-world setup, training uses the Wireless Insite data set in combination with the technique in Sec. D.3.5.2, where the sets $\Omega_{t,q}^{(I)}$ and $\Omega_{t,q}^{(O)}$ are drawn from Ω_t uniformly at random without replacement with $|\Omega_{t,q}^{(I)}| = |\Omega_{t,q}^{(O)}| = 1/2|\Omega_t|$, $q = 1, \dots, Q_t$, and $Q_t = 10 \forall t$. Fig. D.8 shows the RMSE as a function of $|\Omega|$ for the proposed scheme and competing alternatives. By the performance degradation of all four approaches relative to Fig. D.6, it follows that estimating real maps is more challenging than estimating maps in the Gudmundson data set. The performance gap is increased, where the proposed approach now performs roughly 50 % better than

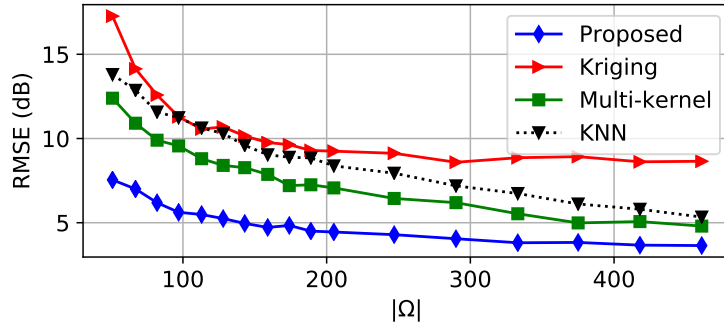


Figure D.8: Comparison with state-of-the-art alternatives. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 10$.

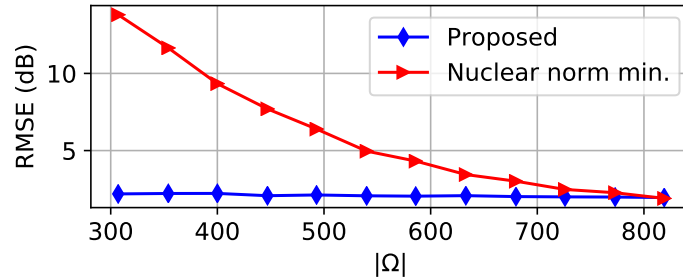


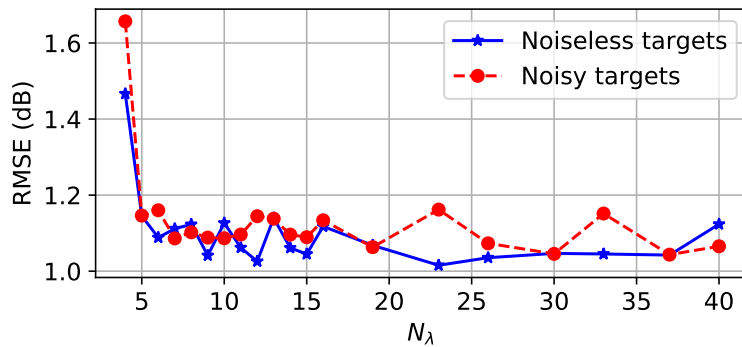
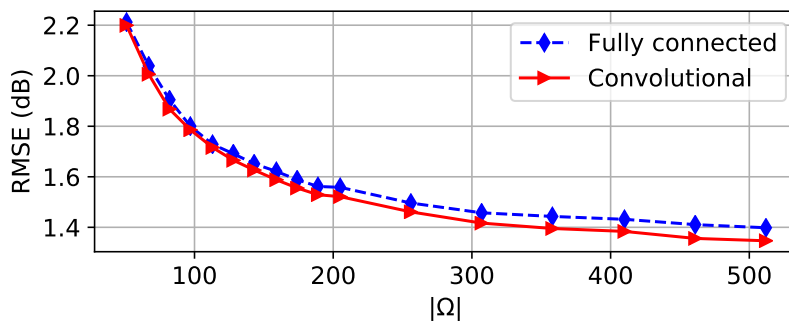
Figure D.9: Performance comparison of the proposed scheme with that of the matrix completion algorithm in [48] where the training and testing maps were obtained from the Wireless InSite data set. The number of grid points in \mathcal{X} is $N_y N_x = 1024$.

the next competing alternative. Again, the algorithm in [48] is not displayed for the same reason as in Fig. D.6. Its RMSE is shown in Fig. D.9 along with that of the proposed algorithm. The later still outperforms this competitor except when $|\Omega|$ is very large, close to $N_y N_x$, as in Fig. D.7.

D.4.2 Deep Neural Network Design

This section justifies the main design decisions regarding the proposed network. To unveil the influence of each architectural aspect, the number of convolution and convolution-transpose filters is adjusted so that the total number of parameters of the neural network N_w remains approximately the same.

The first step is to justify the choice of an autoencoder structure. To this end, Fig. D.10 complements the toy example in Fig. D.3 by plotting the RMSE as a function of the code length N_λ under two setups with pathloss propagation and fixed transmit power: i) Noisy *inputs*, noiseless *targets* in the training phase, and noisy *inputs*, noiseless *targets* in the testing phase. This corresponds to training as a denoising autoencoder; see Sec. D.3.5.1. ii) Noisy *inputs* and *targets* in the training, and noisy *inputs*, noiseless *targets* in the testing. This models how a neural network trained over real data estimates a true map. Note that the irregular behavior of the curves for $N_\lambda > 5$ owes to the fact that each N_λ corresponds to a different network, and therefore a different training process, including the initialization. As observed, the RMSE remains roughly constant for $N_\lambda > 5$, which demonstrates that the spectrum maps in this scenario lie close to a low-dimensional

Figure D.10: RMSE as a function of the code length N_λ , $|\Omega| = 104$.Figure D.11: RMSE of the proposed network as a function of $|\Omega|$ for two types of output layers for the encoder.

manifold. This justifies the autoencoder structure. Besides, training as a denoising autoencoder offers a slight performance advantage, yet it is only possible with synthetic data; see Sec. D.3.5.1. When other propagation phenomena such as shadowing need to be accounted for, $N_\lambda > 5$ is however required.

A second design consideration is whether the last layer of the encoder should be convolutional or fully connected. In the former case, the code would capture shift-invariant features, whereas greater flexibility is allowed in the latter case. This dilemma is ubiquitous in deep learning since convolutional layers constitute a special case of fully connected layers. The decision involves the trade-off between flexibility and information that can be learned with a finite number of training examples. This is investigated in Fig. D.11, which shows the RMSE as a function of the number of measurements $|\Omega|$ for these two types of layers. As observed, in the present case, fully convolutional autoencoders perform slightly better. Besides, they accommodate inputs of arbitrary N_x and N_y . For these reasons, the proposed architecture is fully convolutional.

Two more design decisions involve the number of layers L and the choice of the activation functions. Fig. D.12 shows the RMSE as a function of L with LeakyReLU and PReLU activations [44], where the latter generalize the former to allow training the leaky parameter. Recall that the number of neurons per layer is adjusted to yield approximately the same number of training parameters for all L . Thus, this figure embodies the trade-off between the number and complexity of the features extracted by the network as well as the impact of overfitting. As observed, the best performance in this case is achieved around $L = 26$ layers. Both activations yield roughly the same RMSE, yet the PReLU

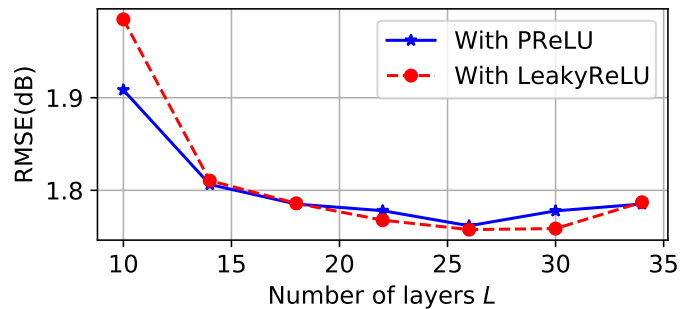


Figure D.12: Map estimate RMSE of the proposed approach as a function of the number of layers L of the autoencoder for two different activation functions, $|\Omega| = 300$.

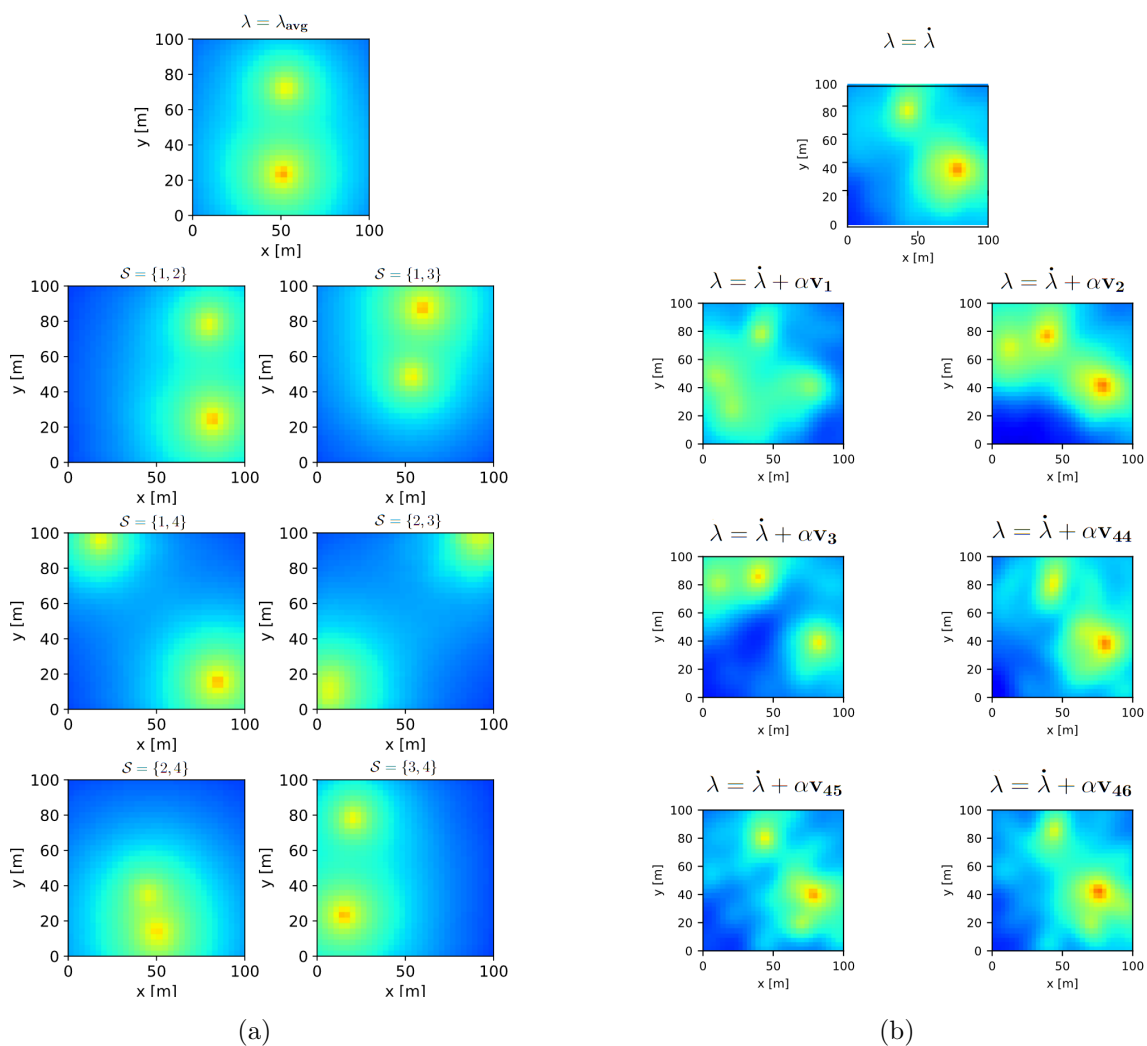


Figure D.13: Decoder outputs of autoencoder architectures with different code length and trained with different data sets: (a) $N_\lambda = 4$ with maps from the free-space propagation model, (b) $N_\lambda = 64$ with maps from the Gudmundson data set, $\alpha = 10$.

outperforms the LeakyReLU for shallow architectures.

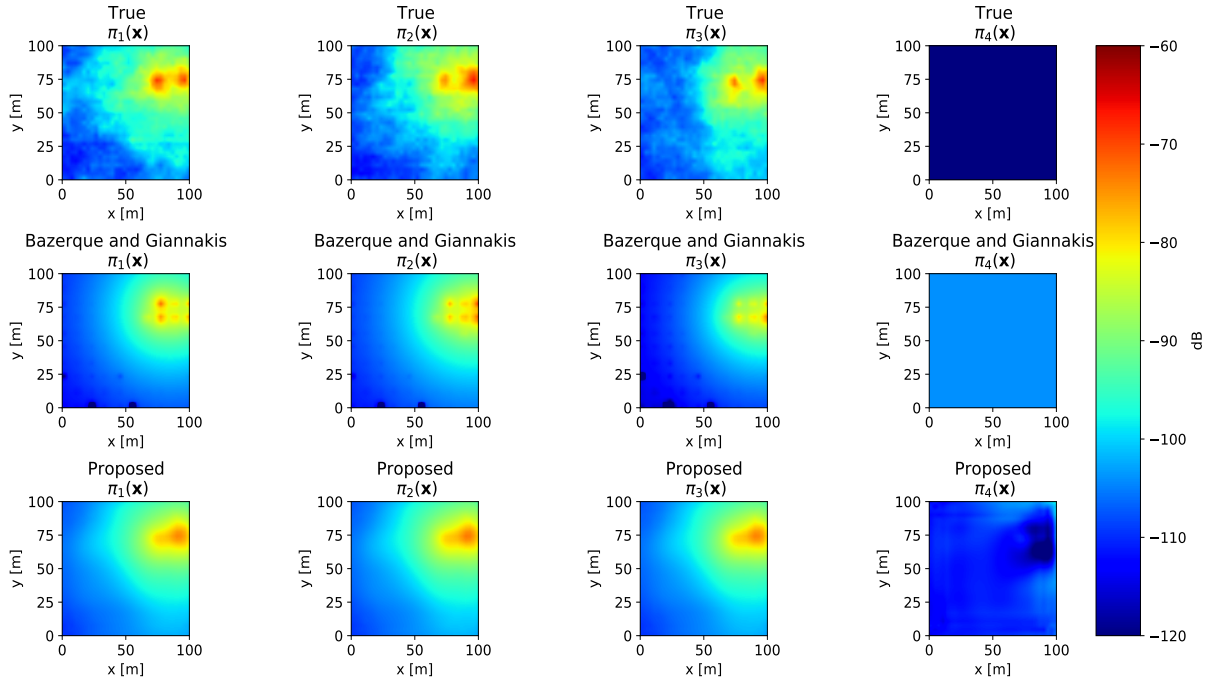


Figure D.14: Maps of the true and estimated coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^B$ over \mathcal{X} , $B = 4$.

D.4.3 Feature Visualization

Although neural networks are mainly treated as black boxes, some visualization techniques offer interpretability of the features that they extract and, therefore, shed light on the nature of the information that is learned. To this end, the next experiment depicts the decoder output when different latent vectors $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda}$ are fed at its input.

First, an instance of the proposed autoencoder with $N_\lambda = 4$ is trained with a dataset of $T_{N_\lambda} = 3 \cdot 10^3$ maps generated using the free-space propagation model with two sources transmitting with a fixed power. Since these maps only differ in the x and y coordinates of the sources, they form a 4-dimensional manifold. Applying the encoder to those maps yields $\{\boldsymbol{\lambda}_t\}_{t=1}^{T_{N_\lambda}}$. The top panel of Fig. D.13a depicts the output of the trained decoder when $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{\text{avg}}$, where $\boldsymbol{\lambda}_{\text{avg}} := (1/T_{N_\lambda}) \sum_t \boldsymbol{\lambda}_t$. As expected, the decoder reconstructs a map with two sources.

The code $\boldsymbol{\lambda}$ acts as the coordinates of a map in the learned manifold. To study this manifold, the output of the decoder is depicted for different values of these coordinates. Specifically, each of the remaining panels in Fig. D.13a corresponds to a value of $\boldsymbol{\lambda} = \check{\boldsymbol{\lambda}}$ with $[\check{\boldsymbol{\lambda}}]_k = [\boldsymbol{\lambda}_{\text{avg}}]_k - [\boldsymbol{\lambda}_{\text{std}}]_k$ if $k \in \mathcal{S}$ and $[\check{\boldsymbol{\lambda}}]_k = [\boldsymbol{\lambda}_{\text{avg}}]_k$ otherwise, where $[\boldsymbol{\lambda}_{\text{std}}]_k := \sqrt{\sum_t ([\boldsymbol{\lambda}_t]_k - [\boldsymbol{\lambda}_{\text{avg}}]_k)^2 / T_{N_\lambda}}$ and the set \mathcal{S} is indicated in the panel titles. It can be observed that moving along the manifold coordinates produces maps of the kind in the training set.

These panels focus on path loss. To understand how shadowing is learned, an instance of the proposed autoencoder with code $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda}$, $N_\lambda = 64$, is trained with the Gudmundson data set. The top panel of Fig. D.13b depicts the output of the trained decoder for $\boldsymbol{\lambda} = \dot{\boldsymbol{\lambda}}$, where $\dot{\boldsymbol{\lambda}}$ was chosen uniformly at random among $\{\boldsymbol{\lambda}_t\}_{t=1}^{T_{N_\lambda}}$. As expected, the decoder reconstructs a map with two sources and the effects of shadowing are noticeable.

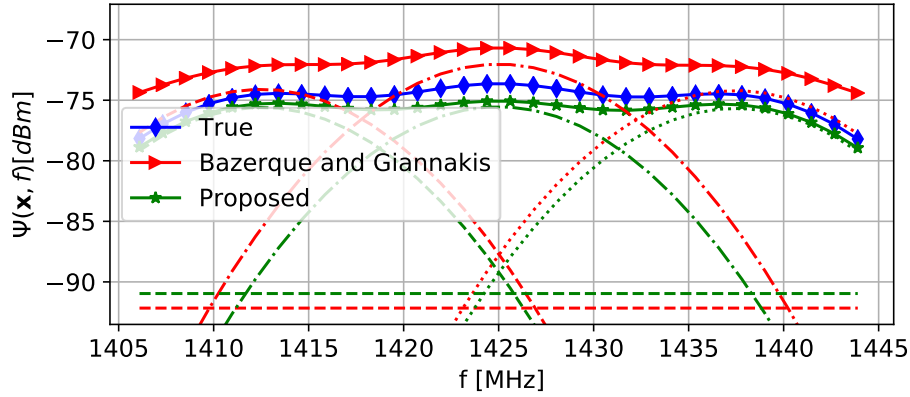


Figure D.15: PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ where the basis expansion model uses Gaussian functions. The non-continuous red (green) curves represent the products $\hat{\pi}_b(\mathbf{x})\beta_b(f)$ estimated by the competing (proposed) algorithm.

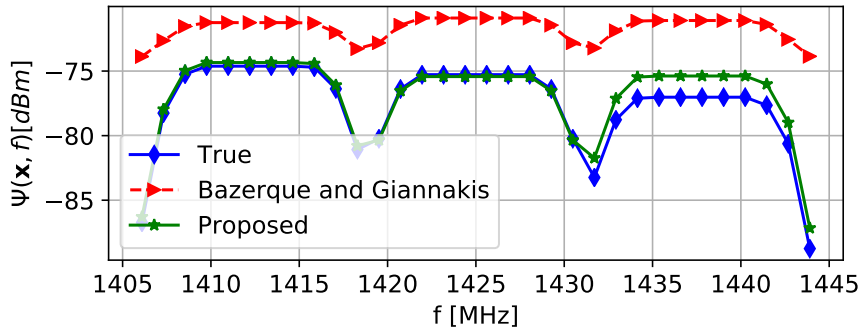


Figure D.16: PSD reconstruction at a random location $\mathbf{x} \in \mathcal{X}$ with a signal basis formed by using raised-cosine functions.

To introduce perturbations in this code along directions that are informative to different extents, let $\mathbf{C}_\lambda := (1/T_{N_\lambda})(\mathbf{\Lambda} - \lambda_{\text{avg}}\mathbf{1}^\top)(\mathbf{\Lambda} - \lambda_{\text{avg}}\mathbf{1}^\top)^\top \in \mathbb{R}^{N_\lambda \times N_\lambda}$ denote the sample covariance matrix of the T_{N_λ} training codes, where $\mathbf{\Lambda} := [\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_{T_{N_\lambda}}]$. The latent vectors are set to $\boldsymbol{\lambda} = \dot{\boldsymbol{\lambda}} + \alpha \mathbf{v}_i$, where α is a fixed constant and \mathbf{v}_i is the i -th principal eigenvector of \mathbf{C}_λ . The remaining panels of Fig. D.13b show the map estimates for $i = 1, 2, 3, 44, 45, 46$ and $\alpha = 10$. As anticipated, changes along the directions of high variability yield maps with markedly different shadowing patterns. The opposite is observed by moving along directions of lower variability, where the reconstructed maps are roughly similar to the one in the top panel.

D.4.4 PSD Cartography

This section provides empirical support for the approach proposed in Sec. D.3.3.2 for PSD cartography. To this end, each sensor samples the received PSD at $N_f = 32$ uniformly spaced frequency values in the band of interest. The performance of the proposed method is compared with that of the non-negative Lasso algorithm in [2] with regularization parameter 10^{-11} , which yields approximately the best performance. To improve its performance, this algorithm was extended to assume that the noise power is the same at all sensors.

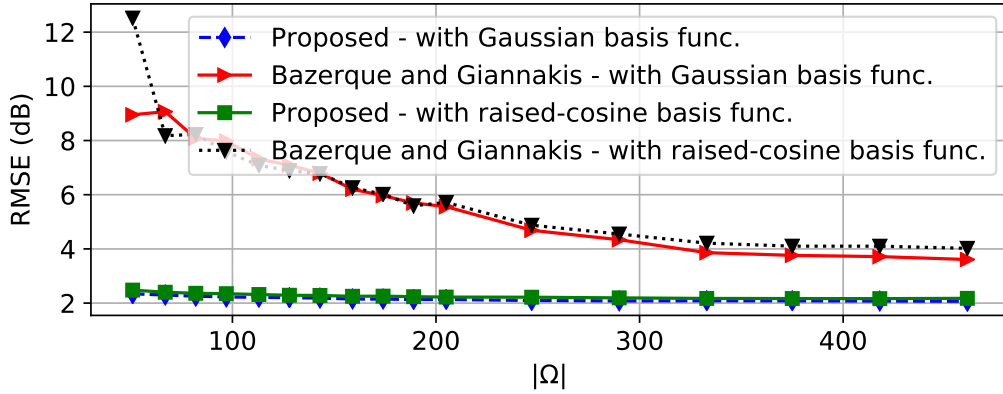


Figure D.17: Performance comparison of the proposed scheme with that of the algorithm in [2].

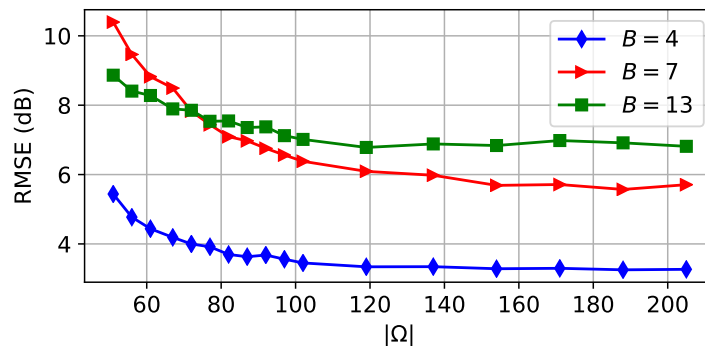


Figure D.18: Map estimate RMSE of the proposed scheme for PSD cartography. The training and testing maps were obtained from the Wireless InSite data set, $Q_t = 5$.

D.4.4.1 Gudmundson Data Set

The first part of this section assesses the performance of the proposed scheme using the training approach in Sec. D.3.5.1 when the training and testing maps were obtained from the Gudmundson data set. The $B - 1 = 3$ *signal* basis functions are uniformly spaced across the band, whereas a fourth constant basis function is introduced to model *noise*; see Sec. D.3.3.2. Two types of signal basis functions are investigated: Gaussian radial basis functions with standard deviation 5 MHz and raised-cosine functions with roll-off factor 0.4 and bandwidth 10 MHz. The noise basis function is scaled to yield $v(\mathbf{x}, f) = v$, where v is a uniform random variable between -100 and -90 dBm/MHz.

The top row of Fig. D.14 portrays the maps of the true coefficients $\{\pi_b(\mathbf{x})\}_{b=1}^4$ over \mathcal{X} ; the second and last rows show their estimates with both schemes when $|\Omega| = 512$. Visually, the proposed scheme produces better estimates despite the fact that it does not exploit the fact that the noise power is the same at all sensors. To demonstrate the reconstruction quality of the proposed scheme, Figs. D.15 and D.16 show the true and estimated PSDs at a random location $\mathbf{x} \in \mathcal{X}$. As observed, the PSD estimate produced by the proposed scheme follows the true PSD more closely compared to the one produced by the competing algorithm. A quantitative comparison is provided in Fig. D.17, which shows the RMSE as a function of the number of measurements $|\Omega|$. As observed, the proposed method outperforms the competing approach with significant margin for small

$|\Omega|$.

D.4.4.2 Wireless Insite Data Set

The second part of this section evaluates the performance of the proposed scheme using the training approach in Sec. D.3.5.2, where the sets $\Omega_{t,q}^{(I)}$ and $\Omega_{t,q}^{(O)}$ are drawn from Ω_t uniformly at random without replacement with $|\Omega_{t,q}^{(I)}| = |\Omega_{t,q}^{(O)}| = 1/2|\Omega_t|$, $q = 1, \dots, Q_t$, and $Q_t = 5 \forall t$. The training and testing maps were obtained from the Wireless InSite data set. The transmit PSD is generated with the raised-cosine functions described in Sec. D.4.4.1. The noise PSD is set to $v(\mathbf{x}, f) = v$, where v is a uniform random variable between -180 and -170 dBm/MHz. Fig. D.18 shows the RMSE of the proposed method as a function of the number of measurements $|\Omega|$. Because of the high RMSE of the competing approach [2] (possibly in part due to the reasons in Remark 8), its performance is not shown on the figure. As observed, the proposed scheme yields a low RMSE in this realistic scenario which emulates training with real measurements.

D.5 Conclusions

Data-driven radio map estimation has been proposed to learn the spatial structure of propagation phenomena such as shadowing, reflection, and diffraction. Learning such structure from past measurements yields estimators that require fewer measurements to attain a target performance. Motivated by the observation that radio maps lie close to a low-dimensional manifold embedded in a high-dimensional space, a deep completion network with an encoder-decoder architecture was proposed to estimate PSD maps. The resulting schemes significantly outperform state-of-the-art alternatives. Future work will include mapping other channel metrics such as channel-gain with alternative network architectures.

References

- [1] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.

- [4] H. B. Yilmaz, T. Tugcu, F. Alagöz, and S. Bayhan, “Radio environment map as enabler for practical cognitive radio networks,” *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 162–169, Dec. 2013.
- [5] P. Huang, O. Castañeda, E. Gönültaş, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, “Improving channel charting with representation-constrained autoencoders,” in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun.*, Cannes, France, Jul. 2019, pp. 1–5.
- [6] N. Patwari and P. Agrawal, “Effects of correlated shadowing: Connectivity, localization, and RF tomography,” in *Int. Conf. Info. Process. Sensor Networks*, St. Louis, MO, Apr. 2008, pp. 82–93.
- [7] D. Romero, Donghoon Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, 2018.
- [8] S. Grimoud, S. B. Jemaa, B. Sayrac, and E. Moulines, “A REM enabled soft frequency reuse scheme,” in *Proc. IEEE Global Commun. Conf.*, Miami, FL, Dec. 2010, pp. 819–823.
- [9] E. Dall’Anese, S.-J. Kim, G. B. Giannakis, and S. Pupolin, “Power control for cognitive radio networks under channel uncertainty,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Aug. 2011.
- [10] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.
- [11] S. Zhang and R. Zhang, “Radio map based 3d path planning for cellular-connected UAV,” in *Proc. IEEE Global Commun. Conf.*, Waikoloa, HI, Dec. 2019.
- [12] D. Romero and G. Leus, “Non-cooperative aerial base station placement via stochastic optimization,” in *Proc. IEEE Mobile Ad-hoc Sensor Netw.*, Shenzhen, China, Dec. 2019, pp. 131–136.
- [13] E. Bulut and I. Guevenc, “Trajectory optimization for cellular-connected uavs with disconnectivity constraint,” in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, May 2018, pp. 1–6.
- [14] J. Chen and D. Gesbert, “Optimal positioning of flying relays for wireless networks: A LOS map approach,” in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017, pp. 1–6.
- [15] G. Boccolini, G. Hernandez-Penalzoza, and B. Bekerull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.

- [16] A. Agarwal and R. Gangopadhyay, "Predictive spectrum occupancy probability-based spatio-temporal dynamic channel allocation map for future cognitive wireless networks," *Trans. Emerging Telecommun. Technol.*, vol. 29, no. 8, pp. e3442, 2018.
- [17] D. Romero, R. Shrestha, Y. Teganya, and S. P. Chepuri, "Aerial spectrum surveying: Radio map estimation with autonomous UAVs," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process., Sep. 2020 (Submitted)*, arXiv preprint arXiv:2005.02432.
- [18] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, "Joint link learning and cognitive radio sensing," in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.
- [19] S.-J. Kim and G. B. Giannakis, "Cognitive radio spectrum prediction using dictionary learning," in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [20] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, "Cellular-base-station-assisted device-to-device communications in TV white space," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [21] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, "Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [22] M. Hamid and B. Beferull-Lozano, "Non-parametric spectrum cartography using adaptive radial basis functions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [23] S. Zha, J. Huang, Y. Qin, and Z. Zhang, "An novel non-parametric algorithm for spectrum map construction," in *Proc. IEEE Int. Symp. Electromagn. Compat.*, Amsterdam, Netherlands, Aug. 2018, pp. 941–944.
- [24] Y. Teganya, D. Romero, L. M. Lopez-Ramos, and B. Beferull-Lozano, "Location-free spectrum cartography," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4013–4026, Aug. 2019.
- [25] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, "Group-lasso on splines for spectrum cartography," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [26] J.-A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via kernel-based learning," *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [27] M. Tang, G. Ding, Q. Wu, Z. Xue, and T. A. Tsiftsis, "A joint tensor completion and prediction scheme for multi-dimensional spectrum map construction," *IEEE Access*, vol. 4, pp. 8044–8052, Nov. 2016.

- [28] G. Zhang, X. Fu, J. Wang, and M. Hong, “Coupled block-term tensor decomposition based blind spectrum cartography,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2019, pp. 1644–1648.
- [29] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.
- [30] D. Lee, S.-J. Kim, and G. B. Giannakis, “Channel gain cartography for cognitive radios leveraging low rank and sparsity,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5953–5966, Jun. 2017.
- [31] D. Lee, D. Berberidis, and G. B. Giannakis, “Adaptive bayesian radio tomography,” *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 1964–1977, Mar. 2019.
- [32] Y. Teganya and D. Romero, “Data-driven spectrum cartography via deep completion autoencoders,” in *Proc. IEEE Int. Conf. Commun., Jun. 2020, arXiv preprint arXiv:1911.12810*.
- [33] X. Han, L. Xue, F. Shao, and Y. Xu, “A power spectrum maps estimation algorithm based on generative adversarial networks for underlay cognitive radio networks,” *Sensors*, vol. 20, no. 1, pp. 311, Jan. 2020.
- [34] P. Stoica and R. L. Moses, “Spectral analysis of signals,” 2005.
- [35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.
- [36] D. Romero, D. Lee, and G. B. Giannakis, “Blind radio tomography,” *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2055–2069, Apr. 2018.
- [37] B. R. Hamilton, X. Ma, R. J. Baxley, and S. M. Matechik, “Propagation modeling for radio frequency tomography in wireless networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 1, pp. 55–65, Feb. 2014.
- [38] J. Fan and T. Chow, “Deep learning based matrix completion,” *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.
- [39] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Trans. Graphics*, vol. 36, no. 4, pp. 107, Jul. 2017.
- [40] G. Vázquez-Vilar and R. López-Valcarce, “Spectrum sensing exploiting guard bands and weak channels,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6045–6057, Sep. 2011.
- [41] D. Romero and G. Leus, “Wideband spectrum sensing from compressed measurements using spectral prior information,” *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6232–6246, Dec. 2013.
- [42] R. Tandra and A. Sahai, “SNR walls for signal detection,” *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 4–17, Feb. 2008.

- [43] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos,” *Pattern Recognition Letters*, vol. 105, pp. 13–22, Apr. 2018.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE Int. Conf. Comput. Vision*, Washington, DC, Dec. 2015, pp. 1026–1034.
- [45] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.
- [46] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of communication systems: modeling, methodology and techniques*, Springer Science & Business Media, 2006.
- [47] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electron. Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.
- [48] G. Ding, J. Wang, Q. Wu, Y. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jan. 2016.