MULTIANTENNA CHANNEL MAP ESTIMATION USING DEEP SPATIAL INTERPOLATION

Gyungmin Kim[†$] *Rui Jin*[‡$] *Wilson Funk*[‡] *Seung-Jun Kim*[‡⋆] *Hyuk Lim*[#]

[†] School of EECS, Gwangju Institute of Science & Technology, Gwangju, South Korea
gyungminkim@gist.ac.kr
[‡] Dept. of CSEE, University of Maryland, Baltimore County, Baltimore, MD, USA
{rjin1,wfunk1,sjkim}@umbc.edu
[#] School of Energy Engineering, KENTECH, Naju, Jeonnam, South Korea
hlim@kentech.ac.kr

ABSTRACT

The radio maps of multiantenna channel state information (CSI) are constructed using deep learning. The desired CSI is predicted for arbitrary locations in a geographical area based on the measurements collected at sampling locations. Such maps can be used to significantly reduce the overhead associated with CSI acquisition. A novel deep architecture is proposed, consisting of an encoder/decoder pair for transforming high-dimensional CSI features to lower-dimensional embeddings, together with a deep embedding interpolator for exploiting the spatial dependency of the CSI. Two important problem classes are tackled in a unified fashion, namely, CSI interpolation and prediction. Practical scenarios involving missing information are also considered. The efficacy of the proposed methods is verified by numerical tests.

***Index Terms*—** Antenna arrays, channel state information, deep neural networks, spatial interpolation, radio map estimation.

1. INTRODUCTION

Radio environment maps characterize the distribution of quantities of interest regarding the radio frequency (RF) environment over a geographical area [1]. Signal strength maps, such as interference power maps and power spectral density (PSD) maps, reveal regions of high spectrum usage [2, 3]. Propagation maps, such as channel gain maps, can predict propagation characteristics between arbitrary two points in the area [4]. They are instrumental for various decision making and resource allocation tasks in the PHY/MAC/network layers, including interference mitigation, opportunistic spectrum access, and unmanned aerial vehicle (UAV) deployment.

Constructing radio maps based on the physical principles requires detailed modeling of obstacles with their electromagnetic properties and demands high computational power. A viable alternative is a data-driven approach, where map estimation entails interpolation of the RF descriptors for *arbitrary* locations where measurements may not be available, based on the samples taken in a set of locations. Thus, effective signal processing and machine learning exploiting underlying spatial correlation are critical.

An early work used a linear parametric model with spatial sparsity of transmissions to construct a power map [5]. To aid device-to-device communication in TV white space, low-rank matrix completion was employed to interpolate crowdsourced measurements [6]. A Kriging linear spatial interpolator was adopted to predict the channel gain between arbitrary locations [7]. A deep completion autoencoder (DCAE) was employed to construct PSD maps in [8]. A signal strength map with uncertainty quantification was constructed incorporating 3D maps of the area [9]. However, these works did not address the problem of estimating multiantenna channel maps.

To maximally exploit multiantenna transmission, channel state information (CSI) must be acquired, using pilot transmissions or blind estimation techniques. As the number of antennas and the mobility of terminals grow, the overhead associated with CSI acquisition becomes a significant factor. Our goal is to construct multiantenna CSI maps, allowing prediction of multiantenna CSI at arbitrary locations, significantly reducing the CSI estimation burden.

Research shows that multiantenna CSI prediction is feasible via machine learning. Coordinated downlink beamforming vectors were predicted based on uplink pilots using a deep neural network (DNN) [10]. The feasibility of inferring remote site CSI from local site CSI was studied in [11]. Lower-dimensional embeddings of multiantenna CSI were shown to be highly correlated with geographic locations [12]. However, the spatial correlation structure of multiantenna CSI has not been exploited for map estimation.

In this work, spatial dependency of multiantenna CSI is captured by a DCAE, a deep spatial interpolator. However, instead of performing interpolation directly in the high-dimensional CSI space, an encoder/decoder pair shared for all locations is trained to produce a low-dimensional embedding space, where the interpolation is done. Thus, the burden of the DCAE is reduced and the embeddings contain only the information relevant for map estimation. Using this useful architecture, two important problem classes can be tackled in a unified fashion. First, the CSI of a given base station (BS) is interpolated in space. Secondly, the CSI of a remote BS is predicted based on the CSI of local BSs. Furthermore, the proposed architecture provides natural remedial strategies in the inference stage even when the location or the local BS CSI feature are *missing*.

The rest of the paper is organized as follows. The problem statements are given in Sec. 2. The proposed map estimation architecture and machine learning formulations are given in Sec. 3. Numerical test results are presented in Sec. 4, and conclusions in Sec. 5.

2. PROBLEM STATEMENT

Our radio map estimation problem is to predict the CSI between a *target* BS equipped with an antenna array and a user equipment (UE) at arbitrary locations in the deployment area $\mathcal{X}$, based on the

This work was supported in part by the US National Science Foundation grant 2242412. [$]Equal contribution. [⋆]Corresponding author.

CSI of one or more *anchor* BSs acquired at sampling locations $\mathcal{S} := \{\mathbf{x}_n\}_{n\in\mathcal{N}} \subset \mathcal{X}$, where $\mathcal{N} := \{1, 2, \ldots, N\}$. To facilitate the implementation, $\mathcal{X}$ is assumed to consist of regular grid points covering the deployment area. Two classes of map estimation problems **(P1)** and **(P2)** are considered.

2.1. Problem (P1): Multiantenna CSI Interpolation

In **(P1)**, the target BS is the same as the anchor BS, and the objective is to *interpolate* the CSI for a BS at arbitrary UE locations in $\mathcal{X}$ based on the CSI measurements of the same BS at sample UE locations $\mathcal{S}$. Suppose that the BS has an M-element antenna array and the UE a single-element antenna. Then, the instantaneous channel vector between the BS and the UE at location $\mathbf{x}$ can be denoted as $\mathbf{h}_\mathbf{x} \in \mathbb{C}^M$. Since the instantaneous channel may vary too fast for a mapping application, here we adopt the channel covariance as the CSI, which is given as $\mathbf{R}_\mathbf{x} := \mathbb{E}\{\mathbf{h}_\mathbf{x}\mathbf{h}_\mathbf{x}^\mathcal{H}\} \in \mathbb{C}^{M\times M}$, where $\cdot^\mathcal{H}$ denotes Hermitian transpose. Thus, **(P1)** can be stated formally as follows. Given a training set $\{(\mathbf{x}_n, \mathbf{R}_{\mathbf{x}_n})\}_{n\in\mathcal{N}}$, construct a map $\mathcal{M}_1 : \mathcal{X} \to \mathbb{C}^{M\times M}$ such that $\mathcal{M}_1(\mathbf{x}) \approx \mathbf{R}_\mathbf{x}$ for arbitrary $\mathbf{x} \in \mathcal{X}$.

2.2. Problem (P2): Multiantenna CSI Prediction

In **(P2)**, the goal is to *predict* the CSI of a target (remote) BS at an arbitrary UE location in $\mathcal{X}$ based on the CSI of anchor (local) BSs at the same UE location. Denote the set of B anchor BSs as $\mathcal{B}$. Consider a target BS that does not belong to $\mathcal{B}$. For simplicity, assume that all BSs are equipped with M-element antenna arrays. At each location $\mathbf{x}_n$, $n \in \mathcal{N}$, the anchor CSI matrices $\mathcal{R}_{\mathbf{x}_n} := \{\mathbf{R}_{\mathbf{x}_n}^{(b)} : b \in \mathcal{B}\}$ and the target-BS CSI $\mathbf{R}_{\mathbf{x}_n}$ are acquired. Then, **(P2)** can be stated as follows. Given a training set $\{(\mathbf{x}_n, \mathcal{R}_{\mathbf{x}_n}, \mathbf{R}_{\mathbf{x}_n})\}$ for $n \in \mathcal{N}$, construct a map $\mathcal{M}_2 : \mathcal{X} \times (\mathbb{C}^{M\times M})^B \to \mathbb{C}^{M\times M}$ such that $\mathcal{M}_2(\mathbf{x}, \mathcal{R}_\mathbf{x}) \approx \mathbf{R}_\mathbf{x}$ for all $\mathbf{x} \in \mathcal{X}$.

Note that $\mathcal{M}_2$ requires both the location $\mathbf{x}$ and the anchor CSI $\mathcal{R}_\mathbf{x}$ as the input features for prediction. In practice, there may be cases where only either of them is available. When only the location is available, a location-based map $\mathcal{M}_2^{\text{loc}} : \mathcal{X} \to \mathbb{C}^{M\times M}$ is desired, with $\mathcal{M}_2^{\text{loc}}(\mathbf{x}) \approx \mathbf{R}_\mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}$. On the other hand, if only the anchor CSI features are available, a CSI-based map $\mathcal{M}_2^{\text{CSI}} : (\mathbb{C}^{M\times M})^B \to \mathbb{C}^{M\times M}$ will be useful, with $\mathcal{M}_2^{\text{CSI}}(\mathcal{R}_\mathbf{x}) \approx \mathbf{R}_\mathbf{x}$.

3. MAP ESTIMATION VIA DEEP INTERPOLATION

To address both **(P1)** and **(P2)** in a unified fashion, our architectural approach is to employ three key components in the DNN, which will be jointly trained. First, an encoder maps the input feature per location to an embedding in a low-dimensional space. Then, the embeddings are interpolated over the geographical area via a spatial interpolator. Finally, a decoder brings the embeddings to the target space, yielding the predictions. This process is illustrated in Fig. 1.

3.1. Method for (P1)

The channel covariance $\mathbf{R}_\mathbf{x}$ can be viewed as the product of the channel gain $G_\mathbf{x}$ and the normalized covariance $\tilde{\mathbf{R}}_\mathbf{x}$ given by

$$G_\mathbf{x} := \text{tr}\{\mathbf{R}_\mathbf{x}\}/M \quad \text{and} \quad \tilde{\mathbf{R}}_\mathbf{x} := \mathbf{R}_\mathbf{x}/G_\mathbf{x} \tag{1}$$

respectively. $G_\mathbf{x}$ depends on large- and medium-scale fading, while $\tilde{\mathbf{R}}_\mathbf{x}$ is akin to small-scale fading. Thus, it is prudent to treat them separately. For each sample location $\mathbf{x} \in \mathcal{S}$, collect in a feature vector $\mathbf{v}_\mathbf{x} \in \mathbb{R}^L$ the gain $G_\mathbf{x}^{\text{dB}} := 10\log_{10} G_\mathbf{x}$ and the upper-triangular part of Hermitian matrix $\tilde{\mathbf{R}}_\mathbf{x}$. The complex entries are separated to

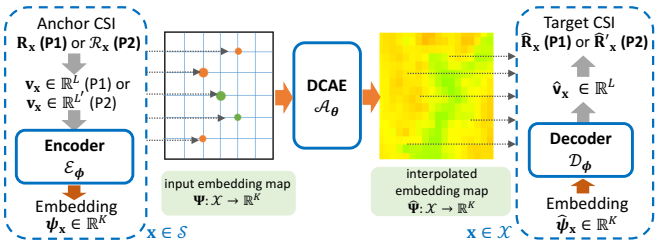


Fig. 1. Proposed map estimation architecture.

real and imaginary parts. Therefore, the dimension L of $\mathbf{v}_\mathbf{x}$ becomes $L := 1 - M + M^2$. Let us denote this feature extraction operation as $\mathbf{v}_\mathbf{x} = \mathcal{F}(\mathbf{R}_\mathbf{x})$, and the inverse operation as $\mathcal{F}^{-1}$. A DNN-based encoder $\mathcal{E}_{\boldsymbol{\phi}}$ with parameters $\boldsymbol{\phi}$ transforms $\mathbf{v}_\mathbf{x}$ to a low-dimensional embedding

$$\boldsymbol{\psi}_\mathbf{x} = \mathcal{E}_{\boldsymbol{\phi}}(\mathbf{v}_\mathbf{x}) \in \mathbb{R}^K, \ \mathbf{x} \in \mathcal{S} \tag{2}$$

where $K < L$. The encoding reduces the dimension of the input features, capturing only the information useful for the interpolation task and suppressing noise and uninformative signals.

Based on the tuples $\{(\mathbf{x}, \boldsymbol{\psi}_\mathbf{x})\}_{\mathbf{x}\in\mathcal{S}}$, spatial interpolation is performed using a DCAE $\mathcal{A}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ [8]. To do this, first construct a tensor $\boldsymbol{\Psi} : \mathcal{X} \to \mathbb{R}^K$ such that

$$\boldsymbol{\Psi}(\mathbf{x}) = \begin{cases} \boldsymbol{\psi}_\mathbf{x} & \text{if } \mathbf{x} \in \mathcal{S} \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{3}$$

which is an embedding map with non-zero values only at the sample locations. The DCAE yields an interpolated embedding map through

$$\hat{\boldsymbol{\Psi}} = \mathcal{A}_{\boldsymbol{\theta}}(\boldsymbol{\Psi}) \tag{4}$$

which satisfies $\hat{\boldsymbol{\Psi}}(\mathbf{x}) \approx \boldsymbol{\psi}_\mathbf{x}$ for $\mathbf{x} \in \mathcal{S}$, but also contains estimates $\hat{\boldsymbol{\psi}}_\mathbf{x} := \hat{\boldsymbol{\Psi}}(\mathbf{x})$ for locations $\mathbf{x} \in \mathcal{S}^c$, where ^{c} denotes set complement.

To obtain the CSI for an arbitrary point $\mathbf{x} \in \mathcal{X}$, pass the interpolated embedding $\hat{\boldsymbol{\psi}}_\mathbf{x}$ through a decoder $\mathcal{D}_{\boldsymbol{\phi}}$ to get

$$\hat{\mathbf{v}}_\mathbf{x} := \mathcal{D}_{\boldsymbol{\phi}}(\hat{\boldsymbol{\psi}}_\mathbf{x}) \in \mathbb{R}^L \tag{5}$$

from which the gain estimate $\hat{G}_\mathbf{x}^{\text{dB}}$ can be read off and the normalized covariance estimate $\hat{\tilde{\mathbf{R}}}_\mathbf{x}$ is constructed. Their product yields the covariance estimate $\hat{\mathbf{R}}_\mathbf{x}$. Thus, $\mathcal{M}_1$ can be compactly expressed as

$$\mathcal{M}_1(\mathbf{x}) := \mathcal{F}^{-1}\left(\mathcal{D}_{\boldsymbol{\phi}}\left(\hat{\boldsymbol{\Psi}}(\mathbf{x})\right)\right), \ \mathbf{x} \in \mathcal{X}. \tag{6}$$

To train the DNNs, the locations $\{\mathbf{x}_n\}_{n\in\mathcal{N}}$ in the training set is randomly partitioned to $\mathcal{S}$ and $\mathcal{V}$ with $\mathcal{S} \cap \mathcal{V} = \emptyset$ and $\mathcal{S} \cup \mathcal{V} = \{\mathbf{x}_n\}_{n\in\mathcal{N}}$. The training problem is then formulated as

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \ \mathbb{E}\left\{\sum_{n=1}^{N}\left[\lambda_1\left(\hat{G}_{\mathbf{x}_n}^{\text{dB}} - G_{\mathbf{x}_n}^{\text{dB}}\right)^2 + \lambda_2\left\|\hat{\tilde{\mathbf{R}}}_{\mathbf{x}_n} - \tilde{\mathbf{R}}_{\mathbf{x}_n}\right\|_F^2 + \lambda_3\left\|\hat{\boldsymbol{\Psi}}(\mathbf{x}_n) - \boldsymbol{\Psi}(\mathbf{x}_n)\right\|_2^2\right]\right\} \tag{7}$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ are the ℓ_2-norm of a vector and the Frobenius norm of a matrix, respectively, and the expectation is w.r.t. the random partition $\mathcal{S}$ and $\mathcal{V}$. Inside the sum in (7), the first and the second terms ensure that the gain and the normalized covariance matrix are reconstructed well, respectively, and the third term encourages $\mathcal{A}_{\boldsymbol{\theta}}$ to be trained as an autoencoder, keeping the input and the output close to each other. Positive parameters λ_1, λ_2, λ_3 balance these three training objectives.

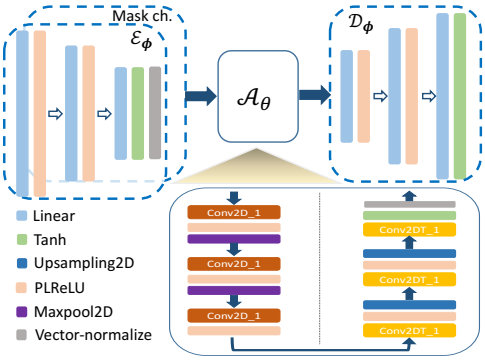


Fig. 2. DNN architecture.

3.2. Method for (P2)

For Problem **(P2)**, the input features at each location $\mathbf{x}$ consist of the CSI set $\mathcal{R}_{\mathbf{x}}$ of the anchor BSs. The feature vector can be constructed from $\mathcal{R}_{\mathbf{x}}$ by collecting the dB-scale gains and the upper-triangular parts of the normalized covariance matrices of $\mathbf{R}_{\mathbf{x}}^{(b)}$ for $b \in \mathcal{B}$. With some abuse of notation, let us denote this feature vector by $\mathbf{v}_{\mathbf{x}} \in \mathbb{R}^{L'}$, where $L' = BL$. Furthermore, the feature extraction operation is again denoted by $\mathcal{F}$, i.e., $\mathbf{v}_{\mathbf{x}} = \mathcal{F}(\mathcal{R}_{\mathbf{x}})$. Also supplied for training is the CSI of the target BS $\mathbf{R}'_{\mathbf{x}}$, from which the dB-scale gain $G'^{\mathrm{dB}}_{\mathbf{x}}$ and the normalized covariance $\tilde{\mathbf{R}}'_{\mathbf{x}}$ are obtained.

The feature vectors $\{\mathbf{v}_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{S}}$ are mapped by an encoder into embeddings $\{\boldsymbol{\psi}_{\mathbf{x}}\}$ via (2). Then, the embeddings are interpolated over $\mathcal{X}$ using the DCAE through (3)–(4). To estimate the CSI of the target BS at arbitrary $\mathbf{x} \in \mathcal{X}$, the interpolated embedding $\hat{\boldsymbol{\psi}}_{\mathbf{x}} := \hat{\boldsymbol{\Psi}}(\mathbf{x})$ is passed through the decoder as (5). Note that the decoded feature $\hat{\mathbf{v}}_{\mathbf{x}}$ has dimension L as it is for the CSI of a single BS. From the decoder output $\hat{\mathbf{v}}_{\mathbf{x}}$, the gain $\hat{G}'^{\mathrm{dB}}_{\mathbf{x}}$ and the normalized covariance matrix $\hat{\tilde{\mathbf{R}}}'_{\mathbf{x}}$ for the target BS are constructed. The training for **(P2)** is done via (7) with $\hat{G}^{\mathrm{dB}}_{\mathbf{x}_n}$, $G^{\mathrm{dB}}_{\mathbf{x}_n}$, $\hat{\tilde{\mathbf{R}}}_{\mathbf{x}_n}$, and $\tilde{\mathbf{R}}_{\mathbf{x}_n}$ replaced by $\hat{G}'^{\mathrm{dB}}_{\mathbf{x}_n}$, $G'^{\mathrm{dB}}_{\mathbf{x}_n}$, $\hat{\tilde{\mathbf{R}}}'_{\mathbf{x}_n}$, and $\tilde{\mathbf{R}}'_{\mathbf{x}_n}$, respectively.

Once the training is done, in the operational phase, $\mathcal{M}_2(\bar{\mathbf{x}}, \mathcal{R}_{\bar{\mathbf{x}}})$ for arbitrary $\bar{\mathbf{x}} \in \mathcal{X}$ can be evaluated as follows. Define the input tensor $\bar{\boldsymbol{\Psi}}$ as

$$\bar{\boldsymbol{\Psi}}(\mathbf{x}) := \begin{cases} \boldsymbol{\psi}_{\mathbf{x}}, & \text{if } \mathbf{x} \in \mathcal{S} \\ \mathcal{E}_{\boldsymbol{\phi}}(\mathcal{F}(\mathcal{R}_{\bar{\mathbf{x}}})), & \text{if } \mathbf{x} = \bar{\mathbf{x}} \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{8}$$

Then, the interpolated tensor is obtained as $\hat{\bar{\boldsymbol{\Psi}}} = \mathcal{A}_{\boldsymbol{\theta}}(\bar{\boldsymbol{\Psi}})$, from which the target-BS CSI can be estimated as

$$\mathcal{M}_2(\bar{\mathbf{x}}, \mathcal{R}_{\bar{\mathbf{x}}}) := \mathcal{F}^{-1}\left(\mathcal{D}_{\boldsymbol{\phi}}\left(\hat{\bar{\boldsymbol{\Psi}}}(\bar{\mathbf{x}})\right)\right). \tag{9}$$

When the UE does not have the anchor CSI $\mathcal{R}_{\bar{\mathbf{x}}}$ available, the interpolated embedding can be read off from $\hat{\boldsymbol{\Psi}}$ (cf. (4)) using the location information only. Thus, $\mathcal{M}_2^{\mathrm{loc}}$ can be defined as

$$\mathcal{M}_2^{\mathrm{loc}}(\bar{\mathbf{x}}) := \mathcal{F}^{-1}\left(\mathcal{D}_{\boldsymbol{\phi}}\left(\hat{\boldsymbol{\Psi}}(\bar{\mathbf{x}})\right)\right). \tag{10}$$

Finally, when the UE has only the anchor CSI $\mathcal{R}$ and not the location information $\bar{\mathbf{x}}$, the DCAE can simply be omitted. That is, $\mathcal{M}_2^{\mathrm{CSI}}(\mathcal{R})$ can be evaluated as

$$\mathcal{M}_2^{\mathrm{CSI}}(\mathcal{R}) := \mathcal{F}^{-1}\left(\mathcal{D}_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\phi}}(\mathcal{F}(\mathcal{R})))\right). \tag{11}$$

It is emphasized that in (10)–(11), the same network trained on the CSI samples with locations is used without separate training.

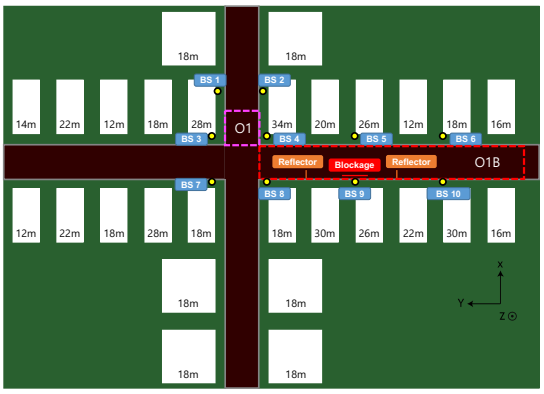


Fig. 3. A top view of the tested propagation environment.

3.3. DNN Architecture

Fig. 2 depicts the DNN architecture for encoder $\mathcal{E}_{\boldsymbol{\phi}}$, decoder $\mathcal{D}_{\boldsymbol{\phi}}$, and DCAE $\mathcal{A}_{\boldsymbol{\theta}}$. The DNN for the encoder consists of three fully-connected layers. In the first two layers, parametric leaky rectified linear units (PLReLUs) are used as nonlinearities, whose leaky parameters can be trained [13]. The input dimension, L for **(P1)** or L' for **(P2)**, is gradually reduced over the layers to the output dimension K of the embedding. The DNN for the decoder similarly contains three fully-connected layers, with the dimension increased to L in the output. In the output layer, $\tanh$ is used to maintain the entries of the embedding vector normalized to the interval $(-1, 1)$, as we preprocess (scale) the data.

The DCAE itself is based on a convolutional encoder/decoder architecture as shown in the lower part of Fig. 2. First, the input tensor $\boldsymbol{\Psi}$ is augmented with a binary mask channel that indicates whether CSI measurements are available at each location $\mathbf{x} \in \mathcal{X}$. Our implementation is based on a 2D grid $\mathcal{X}$. Thus, the augmented tensor is processed by three 2D convolutional layers with PLReLU activations using $\kappa \times \kappa$ kernels ($\kappa = 3$ for **(P1)** and 5 for **(P2)**) with stride 1 to generate an internal code, which is then fed to the decoder, similarly consisting of three 2D transposed convolutional layers. The 2D max pooling operators with pool size 2 and stride 2 are added to the first two layers of the encoder, and factor-2 2D upsampling operators are employed in the first two layers of the decoder.

4. NUMERICAL TESTS

4.1. Test Setup

The proposed method is tested using the DeepMIMO data set, a public data set constructed from a ray-tracing channel simulator in the 3.5 GHz band [14]. A top view of the propagation environment comprising cross streets with buildings is depicted in Fig. 3. The BS locations and the heights of the buildings are also indicated. The UEs are located on the streets. Two scenarios are considered for our tests. In the first scenario, dubbed the "O1" scenario, a square area of size 36 m $\times$ 36 m (shown as a magenta square in Fig. 3) with a 2D grid having a spacing of 10 cm is considered for $\mathcal{X}$. In the second "O1B" scenario, a strip of size 36 m $\times$ 324 m (shown as a red rectangle in the figure) with a grid of a 20 cm-spacing is used. In O1B, two reflectors and one blockage are present. It is assumed that each BS is equipped with an 8-element cubic array antenna, resulting in $L = 57$. Up to 6 multipaths are traced. The embedding dimension K was set to 10. The DNNs were implemented via PyTorch and trained using Adam optimizer [15].

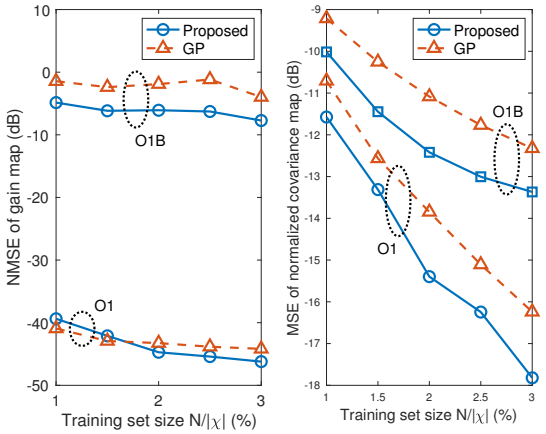


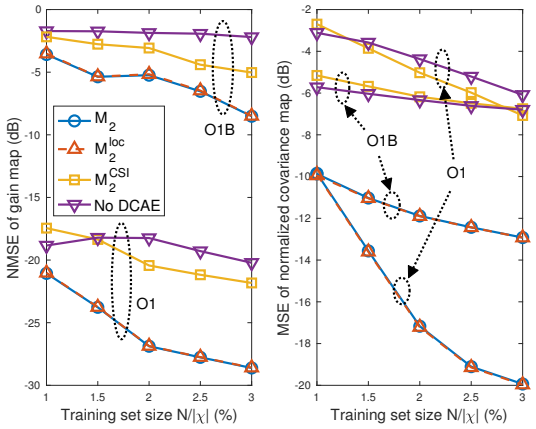
Fig. 4. Map estimation performance for (P1). (Left) NMSE of gain estimation. (Right) MSE of normalized covariance estimation.

4.2. Results for (P1)

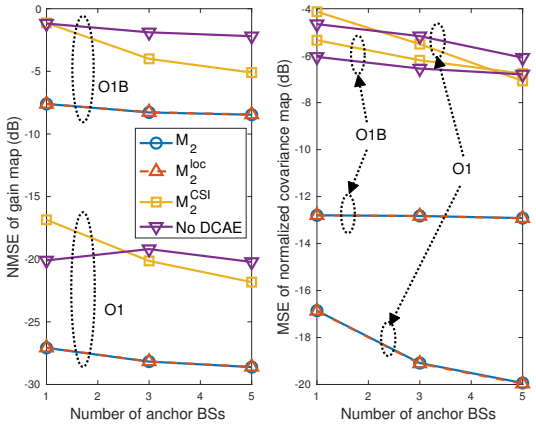
The CSI interpolation method was tested in the O1 and the O1B scenarios. In O1, the CSI for BS 2 in Fig. 3 was targeted, and in O1B, the CSI for BS 9. Fig. 4 shows the map estimation performances in the O1 and O1B scenarios. In the left panel, the gain estimation error is depicted in terms of the normalized mean-square error (NMSE), defined as $10\log_{10}\left(\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}}(G_{\mathbf{x}}-\hat{G}_{\mathbf{x}})^2/G_{\mathbf{x}}^2\right)$, where $\hat{G}_{\mathbf{x}} := 10^{\hat{G}_{\mathbf{x}}^{\mathrm{dB}}/10}$. In the right panel, the MSE of the normalized covariance estimates, computed as $10\log_{10}\left(\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}}\frac{1}{M^2}\|\tilde{\mathbf{R}}_{\mathbf{x}}-\hat{\tilde{\mathbf{R}}}_{\mathbf{x}}\|_F^2\right)$, is shown. The x-axes of the graphs represent the training set size N, normalized by the total number of grid points $|\mathcal{X}|$. For comparison, the performance of Gaussian process (GP) regression is also shown. An isotropic squared exponential function and an isotropic rational quadratic function were adopted for the GP kernels for the gain and the covariance estimation, respectively. Their hyper-parameters were optimized by maximizing the marginal likelihood [16]. It can be seen that the errors for the O1 scenario is smaller than those for the O1B scenario. This is expected as the latter scenario exemplifies a more complex propagation environment than the former. In either scenario, the estimation performance of the proposed method is superior to that of GP regression, except the gain estimation in the O1 scenario, where both methods perform closely and extremely well. It is also seen that the errors for covariance estimation are influenced more by the training set size than the gain estimation errors. In fact, the covariance estimation errors decrease at a slower rate in O1B than in O1 as the training set size increases. These seem to indicate that covariance estimation is more challenging than gain estimation.

4.3. Results for (P2)

For **(P2)**, the target BS in the O1 scenario is BS 3 (cf. Fig. 3), and the anchor BS set $\mathcal{B} = \{\text{BS } 2\}$ when $B = 1$, $\mathcal{B} = \{\text{BSs } 2,4,8\}$ when $B = 3$, and $\mathcal{B} = \{\text{BSs } 1,2,4,7,8\}$ when $B = 5$. For O1B, the target BS is BS 9, and $\mathcal{B} = \{\text{BS } 5\}$ when $B = 1$, $\mathcal{B} = \{\text{BSs } 4,5,6\}$ when $B = 3$, and $\mathcal{B} = \{\text{BSs } 4,5,6,8,10\}$ when $B = 5$. Fig. 5(a) shows the map prediction performance when $B = 5$. In addition to the performances of $\mathcal{M}_2$, $\mathcal{M}_2^{\mathrm{loc}}$, and $\mathcal{M}_2^{\mathrm{CSI}}$, the performance of the case when no spatial interpolation is employed (that is, the DNN is trained without a DCAE) is also depicted. Among the proposed methods, the performances of $\mathcal{M}_2$ and $\mathcal{M}_2^{\mathrm{loc}}$ turn out to be the best



(a) Performance versus training set sizes



(b) Performance versus number of anchor BSs

Fig. 5. Map estimation performance for (P2).

and almost identical. This suggests that incorporating the spatial structure plays a crucial role in multiantenna CSI prediction. The performance of $\mathcal{M}_2^{\mathrm{CSI}}$ is seen to be very close to or slightly better than that of the no DCAE case. This indicates that when the location information is not available, the simple fallback strategy of using just the trained encoder/decoder pair can provide the performance as good as the DNN specifically trained without location information. Fig. 5(b) presents the performance with varying number of anchor BSs and 3% sampling. It is seen that fusing the CSI from multiple anchor BSs can indeed improve the performance.

5. CONCLUSION

Map estimation techniques for the multiantenna CSI have been proposed using deep learning. Two problem classes were tackled. First, the CSI of a BS was interpolated for arbitrary UE locations based on the CSI collected at sample UE locations. Secondly, the CSI of a target BS was predicted over space based on the CSI of anchor BSs collected at sample locations. Both problem classes were tackled using a novel DNN architecture consisting of an encoder/decoder pair, which transforms the high-dimensional CSI features to a lower-dimensional embedding, and a DCAE, which performs spatial interpolation in the embedding space. Numerical tests performed on ray-tracing data sets verified the effectiveness of the methods, and indicated that incorporating spatial correlation structure is critical.

6. REFERENCES

[1] D. Romero and S.-J. Kim, "Radio map estimation: A data-driven approach to spectrum cartography," *IEEE Sig. Process. Mag.*, vol. 39, no. 6, pp. 53–72, Nov. 2022.

[2] 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. Pers. Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.

[3] D. Romero, S.-J. Kim, G. B. Giannakis, and R. Lopez-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Trans. Wireless Commun.*, vol. 65, no. 10, pp. 2547–2560, May 2017.

[4] 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.

[5] J.-A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Sig. Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[6] 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.

[7] E. Dall'Anese, S.-J. Kim, and G. B. Giannakis, "Channel gain map tracking via distributed Kriging," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1205–1211, Mar. 2011.

[8] Y. Teganya and D. Romero, "Deep completion autoencoders for radio map estimation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1710–1724, Aug. 2021.

[9] E. Krijestorac, S. Hanna, and D. Cabric, "Spatial signal strength prediction using 3d maps and deep learning," in *Proc. IEEE Int. Conf. Commun.*, Montreal, Canada (virtual), Jun. 2021, pp. 1–6.

[10] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, Jul. 2018.

[11] S. Chen, Z. Jiang, S. Zhou, Z. Niu, Z. He, A. Marinescu, and L. A. DaSilva, "Learning-based remote channel inference: Feasibility analysis and case study," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3554–3568, Jul. 2019.

[12] C. Studer, S. Medjkouh, E. Gönültas, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47682–47698, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. ICCV*, Washington, DC, Dec. 2015, pp. 1026–1034.

[14] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. Info. Theory Appl. Workshop*, San Diego, CA, Feb. 2019, pp. 1–8.

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

[16] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2005.