Millimeter Wave Channel Recommendation System Aided by Tensor Completion

Tzu-Hsuan Chou, Nicolo Michelusi, David J. Love, and James V. Krogmeier

Abstract—Accurate and fast beam-alignment is important to cope with the fast-varying environment in millimeter-wave communications. A data-driven approach is a promising solution to reduce the training overhead by leveraging side information and on the field measurements. In this work, a two-stage tensor completion algorithm is proposed to predict the received power on a set of possible users’ positions, given received power measurements on a small subset of positions; based on these predictions, a small subset of beams is recommended to reduce the training overhead of beam-alignment, based on positional side information. The proposed method is evaluated with the DeepMIMO dataset, generated from Wireless Insite, which provides parameterized channels for the experiment. The numerical results demonstrate that, with high probability, the proposed algorithm recommends a small set of beams which contain the best beam, thus achieving correct alignment with small training overhead. Given power measurements on only 8 random positions out of 25, our algorithm attains a probability of incorrect alignment of only 2%, with only 2.7% of trained beams. To the best of our knowledge, this is the first work to consider the beam recommendation problem with only the knowledge at neighboring positions but none at the user’s position.

Index Terms—Millimeter wave, beam-alignment, position-aided, tensor completion, sparse learning.

I. INTRODUCTION

Millimeter wave (mmWave) and massive MIMO are the key technologies to enable high throughput communication in future wireless systems, and applications such as video-streaming, automated driving, cloud computing, etc [1]–[4]. However, massive MIMO techniques with narrow beams are required to compensate the path loss and severe signal propagation at mmWave frequencies. Narrow beam communication is especially challenging in mobile environments, since the beam direction needs to be continuously trained. Typically, this is achieved by sweeping over a finite set of candidate beamforming vectors to find the strongest beam direction [5]–[7]. This process incurs huge overhead [8] due to the potentially large set of candidate beamforming solutions that should be searched for in large antenna systems, calling for efficient beam-alignment protocols [9].

Beam-alignment has been a subject of intense research in recent years, with techniques ranging from beam-sweeping [9], AOA/AOD estimation [10], [11], to data-assisted schemes [5], [12]. In particular, beam-sweeping schemes require to collect a set of beam measurements over the possible beam direction regions. The simplest form of beam-sweeping is exhaustive search, which scans through all possible beams between transmitter and receiver. AOA/AOD estimation reduces the number of measurements by leveraging the sparsity of mmWave channels via compressive sensing [10]. In data-assisted schemes, mmWave channels are related to the environment of the user, like the user’s position, the geometry of the surrounding environment (e.g., buildings, vegetation, etc.) or temporal information (e.g., traffic). Due to the difficulty to comprehensively model and accurately represent all propagation features in the environment, and how these affect mmWave propagation, a data-driven approach based on machine learning may be envisioned for this task. In [5], the authors proposed an inverse multi-path fingerprinting approach for beam-alignment utilizing prior measurements at a given position to provide a set of candidate beam directions at the same position. In [6], the authors proposed an online learning approach for beam-alignment with multi-armed bandits.

All prior work focus on the case where beam directions recommendations are provided only on those positions where prior measurements are available (see, e.g., [5]), and thus require to collect a huge amount of channel propagation measurements to cover the entire operational region; therefore, these algorithms fail to predict the channel in those positions where measurements are not yet available. In many practical settings, the spatial correlation in the channel may be exploited to provide beam directions recommendations also in new positions, where prior measurements are unavailable. To address this more general problem, in this work we leverage the tensor completion technique. This problem has been recently investigated in many areas, such as computer vision, image inpainting, recommendation system, etc [13]–[15].

In this work, we propose a recommendation algorithm to mitigate the training overhead of the position-aided beam-alignment protocol. Thanks to the low-rank of mmWave MIMO channels [1], [2], [10], we construct the data model on a subset of positions as a tensor and formulate the tensor completion problem to estimate the channel on those positions where measurements are missing. To capture the channel spatial correlation, we introduce a smooth constraint that induces similarity among adjacent positions and beams. We propose a two-stage tensor completion algorithm composed of two smooth matrix completions [14] and a greedy selection algorithm to recommend a subset of candidate beams. We show numerically that our proposed beam recommendation algorithm can provide accurate beam candidates with small training overhead.

This paper is organized as follows. In Sec. II, we present the system model and architecture; in Sec. III, we propose the recommendation algorithm with two-stage tensor completion. The numerical result are presented in Sec. IV; followed by concluding remarks in Sec. V.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; emails: {chou59, michelus, djlove, jvk}@purdue.edu.

This research has been funded by NSF under grant CNS-1642982.
Notation: $a$ is a scalar, $a$ is a vector, $A$ is a matrix, and $A$ is a tensor. $a^*$ is the complex conjugate. $A^T$, $A^*$ are transpose and conjugate transpose. $\|A\|_F$ is the Frobenius norm. $A \otimes B$ denotes the Kronecker product. $\text{vec}(A)$ is the vectorization and $\text{tr}(A)$ is the trace. $A_{ij}$ is the $(i, j)$-th element of $A$.

II. SYSTEM MODEL

In this section, we first describe the channel model and the beamforming codebook of our communication system. Then, we introduce the construction of the data model for the learning system. Later, we depict the steps of the position-aided beam-alignment and explain the goal of this work.

A. Channel Model

We consider a scenario with one base station (BS), which services an area labeled by Euclidean coordinates $(p_x, p_y)$, where $p_x \in \{\Delta_s, i, i = 1, \cdots, L_x\}$, $p_y \in \{\Delta_s, i, i = 1, \cdots, L_y\}$, and $\Delta_s$ is the separation of the points in the grid. The geometric channel model [2] is assumed for the downlink and is the separation of the points in the grid. The geometric channel model [2] is assumed for the downlink and the conjugate transpose.

$$\mathbf{h}^{p} = \sqrt{N_t} \sum_{\ell=1}^{L} \alpha^{p}_{\ell} \mathbf{a}^{p}_{\ell}(\theta^{p}_{\ell}, \phi^{p}_{\ell}),$$

where $\mathbf{a}_{\ell}(\theta^{p}_{\ell}, \phi^{p}_{\ell})$ (see [1]) is the normalized transmit steering vector of the $\ell$-th path; $\theta^{p}_{\ell}$ and $\phi^{p}_{\ell}$ are its elevation and azimuth angles; $\alpha^{p}_{\ell}$ is the complex channel gain; $L$ is the number of paths; and $N_t$ is the number of transmit antennas.

In order to develop our data-driven approach, we aim to leverage the channel correlation with respect to some features of the environment which the user is operating in. Here, we consider the correlation between the channel and the user's position. To the best of our knowledge, accurate prediction of the propagation channel can only be achieved by real channel measurements or by simulation in ray-tracing software, which requires accurate modeling of the propagation environment, such as position of buildings, scatterers, etc [16]. In this work, our algorithm is based on channel measurements, but these are generated based on a ray-tracing simulation.

B. Beam Codebook and Received Signal Model

We consider a uniform planar array (UPA) with $N_x$ and $N_y$ antennas and $\lambda/2$ antenna spacing along the $x$ and $y$ directions (a total of $N_t=N_x N_y$ antennas), and a beamforming codebook

$$\mathcal{F}=\{\mathbf{f}_{i,j} = \mathbf{a}(\theta^y, \phi^y), i=1, \cdots, C_\theta, j=1, \cdots, C_\phi\},$$

of size $|\mathcal{F}|=C_\theta C_\phi$, where $\mathbf{a}(\theta, \phi)$ represents a beam pointing in the elevation and azimuth angles $(\theta, \phi) \in [-\pi/2, \pi/2]^2$.

$$\mathbf{a}(\theta, \phi) = \frac{1}{\sqrt{N_t}} \left[ 1 \ e^{j \Omega_y} \ldots e^{j(N_y-1)\Omega_y} \right]^T \otimes \left[ 1 \ e^{j \Omega_z} \ldots e^{j(N_z-1)\Omega_z} \right]^T,$$

(1)

with $\Omega_y=\pi \sin \theta \sin \phi$, $\Omega_z=\pi \sin \theta \cos \phi$. To construct $\mathcal{F}$, $\theta^y$ and $\phi^y$ are uniformly quantized in $[-\pi/2, \pi/2]$ as

$$\theta_i = \frac{\pi}{2} + (i-1) \times \frac{\pi}{C_\theta}, \ i=1, \cdots, C_\theta$$

(2)

$$\phi_j = \frac{\pi}{2} + (j-1) \times \frac{\pi}{C_\phi}, \ j=1, \cdots, C_\phi,$$

(3)

where $C_\theta$ and $C_\phi$ define the resolution of elevation and azimuth angles, respectively. We define the set of indices corresponding to the beamforming vectors in $\mathcal{F}$ as

$$\mathcal{I} \equiv \{(i, j) : i = 1, \cdots, C_\theta, j = 1, \cdots, C_\phi\}.$$

With the beamforming vector $\mathbf{f}_{i,j}$, $(i, j) \in \mathcal{I}$, the signal received at position $p$ can be written as

$$y_{i,j}^p = \sqrt{P_t} \mathbf{h}^{p} \mathbf{f}_{i,j} s + v,$$

where $s$ is the known training signal with $|s|=1$ and $v$ is the received zero-mean complex Gaussian noise with variance $\sigma^2_v$; $P_t$ is the transmit power. The received power with the $(i, j)$-th beamformer can be estimated as

$$r_{i,j}^p = |s^* y_{i,j}^p|^2 = |\sqrt{P_t} \mathbf{h}^{p} \mathbf{f}_{i,j} + \tilde{v}|^2,$$

where $\tilde{v} = s^* v$ is still a zero-mean complex Gaussian noise with variance $\sigma^2_v$. These received powers are stored in a database, and then used in our completion framework to estimate the received power in other positions and beam pairs.

C. Data Model

Data is the essential element for the machine learning approach. Here, we describe how we store the information in the database. The database contains the received power $r$, computed with a fixed transmit power $P_t$, along with the side information, including the user’s position $p = (p_x, p_y)$ and the indices of the beamforming codeword $\mathbf{f}_{i,j}$:

| TABLE I: Database form |
|-------------------------|
| $p_x$ | $p_y$ | $i$ | $j$ | rx power |
| 1     | 1     | 1   | 4   | 5.2    |
| 1     | 2     | 4   | 5   | 6.1    |
| :     | :     | :   | :   | :      |

We represent the database as a 4-th order tensor

$$\mathcal{T}(p_x, p_y, i, j) = \begin{cases} r_{i,j}^{(p_x, p_y)}, & (p_x, p_y, i, j) \in \Psi, \\ 0, & \text{otherwise} \end{cases},$$

(4)

where $\Psi$ is the set of observed combinations of positions and beams stored in the database; the unobserved entries $(p_x, p_y, i, j) \notin \Psi$ are set to zero; However, it is impractical to collect the information with all combinations of positions and beam-directions into the database due to the limited sampling resources, so some positions possibly have no representation in the database. Even in the observed position, there might be only limited number of beams’ information recorded. Therefore, the tensor $\mathcal{T}$ is highly incomplete.

D. Position-Aided Beam-Alignment

The idea of this approach is to provide a set of candidate beam-directions according to the user’s position. Since the overhead of the conventional beam-sweeping approach is unacceptable (it scales with $|\mathcal{F}|$ and is typically very large), our objective is to design a learning algorithm that provides a small subset $\mathcal{S} \subset \mathcal{F}$ of candidate beams for training, which
is likely to contain the best beam (the one associated with highest received power). In Fig. 1 we introduce the flow of the position-aided beam alignment. In step 1, after the BS notifies the user to initiate downlink mmWave transmission, the user sends its position \( u = (u_x, u_y) \) to the BS using sub-6GHz control channels. The position is available via a suite of sensors such as GPS, camera, LIDAR \([5], [17]\). In step 2, the BS transfers the user’s position \( u \) to the cloud. The cloud processes the learning algorithm and provides the recommended beam set \( S \). In step 3, the BS trains the recommended set \( S \) by transmitting a sequence of \(|S|\) signals, each associated with its own codeword in \( F \), to the user. In step 4, the user selects the beam with the largest received power and feeds its index back to the BS. Finally, the BS uses the selected beam for the subsequent mmWave downlink transmission.

### III. Proposed Beam Recommendation Algorithm

Our goal is to recommend a set \( S \) of \( N_{tr} \) candidate beams for the user to train, based on its position. If the user is in a position \( u \) represented in the database, and with all beam measurements available, \( \gamma_{i,j}^{u}, \forall(i,j) \), then this task can be easily accomplished by recommending the \( N_{tr} \) beams with highest received power in the given position.

Otherwise, we design an algorithm based on tensor completion that employs the knowledge at neighboring positions to support the beam recommendation for the user. Note that successful completion is highly dependent on the sampling set: if measurements are missing on a certain row or column of an incomplete matrix then no reconstruction is possible on that row or column, if we only rely on a low-rank approximation \([13]\). A similar issue also exists in the tensor case: if measurements are missing on a certain row or column of an incomplete tensor then no reconstruction is possible on that row or column.

Therefore, the low-rank tensor completion algorithm often fails to provide predictions on unobserved positions.

To address this challenge, in addition to the low-rank approximation, we enforce a smoothness constraints across adjacent entries on a given dimension. This constraint captures realistic spatial correlations across adjacent positions arising in mmWave channels: similarity between neighboring beams at a given position; and similarity between neighboring positions on a given beam direction. In Sec. III-A we propose a two-stage tensor completion implemented by dividing the tensor completion into two smooth matrix completions (SMCs), considered in Sec. III-B. Finally, in Sec. III-C we propose a greedy algorithm to provide the recommended beams based on the predicted received power.

#### A. Two-stage Tensor Completion

Given the data tensor \( T \) in \([4]\) and the set of observed combinations of positions and beams \( \Psi \), we aim to recover the incomplete tensor \( T \). Since the data tensor \( T \) in \([4]\) is highly incomplete, we might only have limited number of beams’ information on few observed positions, causing the low-rank completion to fail. To address this challenge, we propose a two-stage tensor completion, each based on SMC.

The two-stage tensor completion is done by dividing the tensor completion into two SMCs for the matrix in different dimensions. In the first stage, for each observed position \((x_o, y_o)\) such that \((x_o, y_o, i, j) \in \Psi\) for some \((i, j)\), we do the SMC on the beam matrix to predict the received power on the unobserved beam directions, by exploiting the fact that the received power should vary smoothly between neighboring beams in the same position. To this end, let \( B^{(x_o, y_o)} = \mathcal{T}(x_o, y_o, :, :) \) be the (possibly, incomplete) matrix of received powers along beam directions, for the given position \((x_o, y_o)\); let \( \Omega \equiv \{(i, j) : (x_o, y_o, i, j) \in \Psi\} \) be the set of observed beam directions in position \((x_o, y_o)\); then the SMC can be expressed as

\[
\text{SMC} \left( B^{(x_o, y_o)}, \Omega \right) = \arg \min_{X} \|X\|_* + \gamma_{+} \left( \|D_{C_{y}}X\|_{F}^{2} + \|X D_{C_{x}}^{T}\|_{F}^{2} \right) \tag{5}
\]

s.t. \( X_{\Omega} = B_{\Omega}^{(x_o, y_o)} \)

and is considered in Sec. III-B. Above, the first term \( \|X\|_* \) of the objective function denotes the nuclear norm of \( X \), \( \|X\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i \), where \( \sigma_i \) is the \( i \)-th largest singular value of \( X \). The second term of the objective function in (5) is a penalty term which induces smoothness across entries in each row and column of the matrix. The matrix \( D_{m} \in \mathbb{R}^{(m-1)\times m} \) is the smoothness constraint matrix, which captures the differences between neighboring entries of a matrix:

\[
D_{m} = \begin{bmatrix}
1 & -1 & \cdots & 0 & 0 \\
\vdots & 1 & -1 & \vdots & 0 \\
0 & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & -1
\end{bmatrix}
\] \tag{6}

Thus, \( \|X D_{C_{x}}^{T}\|_{F}^{2} \) and \( \|D_{m}X\|_{F}^{2} \) quantify the row and column smoothness of matrix \( X \), respectively. In our optimization, we...
consider smoothness on rows and columns simultaneously, as opposed to LTVNN \cite{18}, which considers them separately.

After the first-stage completion by SMC, we get the data tensor $\hat{T}'$ completed on the observed positions and update $\hat{\Psi}'$ by setting all predicted terms as observed. In the second-stage, the SMC is implemented on the position matrix for each beam direction, by leveraging the fact that the received power should also vary smoothly between neighboring positions on a given beam. Specifically, in each beam $(i_o, j_o)$ such that $(p_x, p_y, i_o, j_o) \in \hat{\Psi}'$ for some $(p_x, p_y)$, let $M(i_o,j_o) = \hat{T}'(::,::,i_o,j_o)$ be the (possibly, incomplete) matrix of received powers on different positions, along the given beam direction indexed by $(i_o,j_o)$; let $\Omega = \{(p_x, p_y) : (p_x, p_y, i_o, j_o) \in \hat{\Psi}'\}$ be the set of observed positions along the beam direction $(i_o, j_o)$; then the SMC can be expressed as

\[
\text{SMC}(M^{(i_o,j_o)}, \Omega) = \arg \min_X \|X\|_* + \gamma(\|D_L X\|_F^2 + \|X D^T_L\|_F^2) \quad \text{s.t. } X_{\Omega} = M^{(i_o,j_o)}
\]

and is considered in Sec. III-B. After the second-stage completion by SMC, we get the completed tensor $\hat{T}$, which predicts the received power of all unknown positions and beams, and is used in Sec. III-C to recommend the set of training beams at a given user position $u$. The two-stage tensor completion algorithm is described in Algorithm 1. In the next section, we discuss the solutions of the SMC problems (5) and (7), solved with Algorithm 2.

### Algorithm 1 Two-stage Tensor Completion

**Input:** incomplete data tensor $\mathcal{T}$ and the observed set $\hat{\Psi}$

**Output:** $\hat{T}$

1. for $(x_o, y_o, -,-) \in \hat{\Psi}$ do
2. $B^{(x_o,y_o)} = \mathcal{T}(x_o, y_o, ::)$
3. $\Omega = \{(i,j) | (x_o, y_o, i,j) \in \hat{\Psi}\}$
4. $\hat{T}'(x_o, y_o, ::) = \text{SMC}(B^{(x_o,y_o)}, \Omega)$
5. Update the observed set $\hat{\Psi}'$.
6. end for
7. for $(-, -, i_o, j_o) \in \hat{\Psi}'$ do
8. $M^{(i_o,j_o)} = \hat{T}'(::,::,i_o,j_o)$
9. $\Omega = \{(p_x, p_y) | (p_x, p_y, i_o, j_o) \in \hat{\Psi}'\}$
10. $\hat{T}(::,::,i_o,j_o) = \text{SMC}(M^{(i_o,j_o)}, \Omega)$
11. end for

### Algorithm 2 Smooth Matrix Completion (SMC)

**Input:** incomplete data matrix $M$ and observed set $\Omega$

**Output:** $M$

1. Initialization $X_t = Y_t = M$, $Z_t = 0$, $\epsilon_t = \infty$
2. while $\epsilon_t > \epsilon$ do
3. $X_{t+1} = D_{1/\lambda}(X_t + \frac{1}{\lambda} Z_t)$
4. Derive $Y_{t+1}$ by \cite{13}
5. $Y_{t+1} = Y_{t} + \lambda M_{t}$
6. $Z_{t+1} = Z_t + \beta(Y_{t+1} - X_{t+1})$
7. $\epsilon_{t+1} = \|X_{t+1} - Y_{t+1}\|_F^2$; $t := t + 1$
8. end while
9. $M = X_{t+1} + M_{t}$

### B. Smooth Matrix Completion (SMC)

Here, we introduce the smooth matrix completion which would be implemented as the sub-function in the two-stage tensor completion. The smooth matrix completion exploits both the low rank and the smoothness of the data. We define $\Omega$ as the observed set of the matrix. Given the matrix $M \in \mathbb{R}^{m \times n}$ and the observed set $\Omega$, the matrix $M_{\Omega}$ means that $M_{ij}$ for $(i,j) \in \Omega$ are given; the unobserved elements $(i,j) \notin \Omega$ are set to zero. The optimization problem is expressed as

\[
\min_X \|X\|_* + \gamma(\|D_m X\|_F^2 + \|X D^T_m\|_F^2) \quad \text{s.t. } X_{\Omega} = M_{\Omega}
\]

where $\gamma$ is a regularization parameter.

The alternating direction method of multipliers (ADMM) is a Lagrangian based approach to efficiently solve (8). With ADMM, we reformulate the problem as

\[
\min_{X,Y,Z} \|X\|_* + \gamma(\|D_m Y\|_F^2 + \|Y D^T_n\|_F^2) + \frac{\lambda}{2} \|Y - X\|_F^2
\]

\[
\text{s.t. } Y_{\Omega} = M_{\Omega}, \quad X = Y,
\]

where $\lambda > 0$ is a small fixed parameter. We introduce the Lagrangian multiplier $Z$ associated with the constraint $X = Y$. The augmented Lagrange function of (9) is

\[
L(X,Y,Z) = \|X\|_* + \gamma(\|D_m Y\|_F^2 + \|Y D^T_n\|_F^2) + \frac{\lambda}{2} \|Y - X\|_F^2 + \text{tr}(Z^T(Y - X)) + \frac{\lambda}{2} \|Y - X\|_F^2.
\]

The ADMM algorithm is implemented by minimizing iteratively $L(X,Y,Z)$ over $X$ and $Y$, and then update the Lagrangian multiplier $Z$:

\[
X_{t+1} = \arg \min_X L(X,Y_{t},Z_t)
\]

\[
Y_{t+1} = \arg \min_Y L(X_{t+1}, Y, Z_t)
\]

\[
Z_{t+1} = Z_t + \beta(Y_{t+1} - X_{t+1}),
\]

where $\beta$ is a step-size. To optimize $X$, we minimize $L(X,Y,Z)$ with fixed $Y$ and $Z$, yielding

\[
\arg \min_X \frac{1}{\lambda} \|X\|_* + \frac{1}{2} \|X - (Y + \frac{1}{\lambda} Z)\|_F^2.
\]

In \cite{14}, it is shown that this problem is strictly convex and its solution is given by the singular value thresholding, $X_{t+1} = D_{1/\lambda}(Y_t + \frac{1}{\lambda} Z_t)$, where $D_\tau$ is the soft-thresholding operator. For a matrix $X$ with singular value decomposition $X = U \Sigma V^T$, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r)$, this is defined as

\[
D_\tau(X) = UD_\tau(\Sigma)V^T, \quad D_\tau(\Sigma) = \text{diag}((\max\{|\sigma_i - \tau|, 0\})).
\]

To optimize $Y$, we minimize $L(X,Y,Z)$ over $Y$ with fixed $X$ and $Z$. We can reformulate the problem as

\[
\arg \min_Y \gamma(\|D_m Y\|_F^2 + \|Y D^T_n\|_F^2) + \text{tr}(Z^T(Y - X)) + \frac{\lambda}{2} \|Y - X\|_F^2
\]

\[
\text{s.t. } Y_{\Omega} = M_{\Omega}.
\]

To solve this problem, we restrict its optimization to the unobserved set $\Omega = \{(i,j) : (i,j) \notin \Omega\}$, and force $Y_{ij} = M_{ij}$ for $(i,j) \in \Omega$. By computing the derivative of (12) with respect
We express the linear equation in matrix form as

$$Y = C.$$  

### Recommendation Algorithm

**Input:**
Beam Subset Selection

**Algorithm 3**

1. **Initialization** $S_0 \leftarrow \emptyset$
2. for $n = 1 : N_{tr}$ do
3. $(i^*, j^*) = \arg \max_{(i,j) \in \mathcal{I} \backslash S_{n-1}} \hat{T}(u_x, u_y, i, j)$
4. $S_n \leftarrow S_{n-1} \cup (i^*, j^*)$
5. end for

### IV. NUMERICAL RESULT

In this simulation, we evaluate our proposed beam recommendation algorithm with the DeepMIMO dataset [20].

We consider the outdoor scenario O1 represented in Fig. 2. There is the main street with buildings which are solid with rectangular shapes on both sides. Only one base station (BS1) is activated. For the position coordinates, we consider $L_x = 5$, $L_y = 5$, $\Delta_x = 5m$ and the position $(p_x, p_y) = (1, 1)$ corresponds to location $=(242.42, 402.77)m$ in Fig 2. For each position in the grid, we extract the corresponding channel from DeepMIMO. Downlink MISO channels are considered and assumed $16 \times 16$ UPA at BS. The UPA codebook size is $|\mathcal{F}|=256$ with $C_d=16$ and $C_0=16$.

In our evaluation, the number of observed positions $N_{ob}$ is varied, and these positions are chosen randomly, but we assume that the four corners of the position coordinate are chosen to avoid the sampling set of data being over-concentrated. In each observed position, we assume that only the measurement of top-$N_b$ beams (ranked by the received power) are stored in the database. We consider $N_b = 5$ in the experiment.

#### A. Performance of Proposed Beam-alignment Algorithm

We evaluate the power loss probability $P_{pl}(\mathcal{S}_p)$ versus the percentage of trained beams $(|\mathcal{S}_p|/|\mathcal{F}|)$ with different number of observed positions. The noise impact is ignored. To measure the beam alignment accuracy for the recommended set $\mathcal{S}_p$, we define the metric as the power loss probability $P_{pl}(\mathcal{S}_p) = 1 - P_{s}(\mathcal{S}_p)$, where $P_{s}(\mathcal{S}_p)$ is the probability that the best beam is included in the set $\mathcal{S}_p$ at position $p$.

$$P_{s}(\mathcal{S}_p) = \mathbb{P} \left( \max_{(i,j) \in \mathcal{S}_p} r_{p,k}^b = \max_{(p,q) \in \mathcal{Z}} r_{p,q}^b \right).$$

In each iteration, we average the power loss probability over all unobserved positions $p$. In Fig. 3, the average $P_{pl}(\mathcal{S}_p)$ decreases when the database contains more observed positions with fixed percentage of trained beams. Our proposed method performs significantly better than the method with randomly-selected beams. To the best of our knowledge, the previous machine learning work [5], [6] for mmWave beam alignment utilize the prior knowledge already available at a given position, but are unable to make predictions if measurements are not available. In contrast, our formulation allows to make predictions by exploiting spatial correlation via tensor completion.

#### B. Spectral Efficiency

We evaluate the average spectral efficiency versus the average transmit power. We first define the transmission rate as

$$R = W \log_2(1 + \eta P_t ||h||^2)$$
where $W$ is the bandwidth, $P_t$ the transmit power, $h$ is the MISO channel, and $f$ is the selected beamforming vector. The SNR scaling factor $\eta \triangleq \frac{f^2}{\varsigma N_0}$, where $A = 1/f_c$ is the wavelength, $N_0$ the noise power spectral density, $d$ the distance, and $\varsigma$ is the antenna efficiency. In the experiment, we set $f_c = 60$ GHz, $W = 1.76$ GHz, $N_0 = -174$ dBm/Hz, and $\varsigma = 1$. The microslot duration $\delta_S = 10$ $\mu$s is the time for one transmission. The frame time $T_{frame} = 5$ $\mu$s is fixed. The training time is $T_{train} = (N_{tr} + 1) \times \delta_S$, where the additional one is for the feedback. The fraction of time used for data transmission is $f_{comm} = \frac{T_{frame} - T_{train}}{T_{frame}}$. Then, the average throughput is $\bar{R} = R \times f_{comm}$.

In Fig. 4, the trend of spectral efficiency ($\bar{R}/W$) is monotone increasing. We take the exhaustive search for comparison, which trains all $|F|$ beams in the training phase. The spectral efficiency of our proposed scheme with $(N_{op}, N_{tr}) = (5, 5)$ is around twice better than the spectral efficiency of exhaustive search. It is due to $f_{comm} \approx 0.5$ for exhaustive search, but $f_{comm}$ of our proposed scheme is close to 1 because of the accurate beam-alignment with small $T_{train}$. If we increase $N_{op}$ or $N_{tr}$, the improvement of the spectral efficiency is minor because $I_{pd}(S)$ is lower than 10% in this region.

V. CONCLUSION

In this paper, we propose a learning-based beam recommendation algorithm to reduce the training overhead for the position-aided beam alignment protocol. We consider the scenario that the user is located in the position with no measurement in the database. We propose the two-stage tensor completion to predict the received power of the user, and then provide the set of recommended beams by ranking the predicted power of beams. The two-stage tensor completion exploits both the low-rank and smoothness of the data. In the numerical result, it shows that the proposed channel recommendation algorithm does improve the performance of beam-alignment by reducing large amounts of training set.

REFERENCES

[1] T. S. Rappaport et al., “Millimeter wave mobile communications for 5G cellular: It will work!” IEEE Access, vol. 1, pp. 335–349, 2013.
[2] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave mimo systems,” IEEE J. Sel. Topics Signal Process., vol. 10, no. 3, pp. 436–453, April 2016.
[3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive mimo for next generation wireless systems,” IEEE Communications Magazine, vol. 52, no. 2, pp. 186–195, February 2014.
[4] B. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” IEEE Communications Magazine, vol. 52, no. 2, pp. 74–80, February 2014.
[5] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, “Inverse multipath fingerprinting for millimeter wave v2i beam alignment,” IEEE Trans. Veh. Technol., vol. 67, no. 5, pp. 4042–4058, May 2018.
[6] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, “Online learning for position-aided millimeter wave beam training,” IEEE Access, vol. 7, pp. 30507–30526, 2019.
[7] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, “Millimeter wave beamforming for wireless backhaul and access in small cell networks,” IEEE Transactions on Communications, vol. 61, no. 10, pp. 4391–4403, October 2013.
[8] N. R. Akdeniz et al., “Millimeter wave channel modeling and cellular capacity evaluation,” IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1164–1179, June 2014.
[9] M. Hussain and N. Michelusi, “Energy-efficient interactive beam alignment for millimeter-wave networks,” IEEE Trans. on Wireless Communications, vol. 18, no. 2, pp. 838–851, Feb 2019.
[10] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, “Channel estimation and hybrid precoding for millimeter wave cellular systems,” IEEE J. Sel. Topics Signal Process., vol. 8, no. 5, pp. 831–846, Oct 2014.
[11] Z. Marzi, D. Ramasamy, and U. Madhow, “Compressive channel estimation and tracking for large arrays in mm-wave picocells,” IEEE J. Sel. Topics Signal Process., vol. 10, no. 3, pp. 514–527, April 2016.
[12] A. Ali and R. W. Heath, “Compressed beam-selection in millimeterwave systems with out-of-band partial support information,” in Proc. of the IEEE International Conf. on Acoustics, Speech and Signal Processing., March 2017, pp. 3499–3503.
[13] E. Candès and B. Recht, “Exact matrix completion via convex optimization,” Commun. ACM, vol. 55, no. 6, pp. 111–119, Jun. 2012.
[14] J. feng Cai, E. J. Candés, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” SIAM J. Optimization, 2010.
[15] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 208–220, Jan 2013.
[16] A. Alkhateeb, “Deepmimo: A generic deep learning data for millimeter wave and massive MIMO applications,” CoRR, abs/1902.06435, 2019. [Online]. Available: http://arxiv.org/abs/1902.06435
[17] E. Ward and J. Folkesson, “Vehicle localization with low cost radar sensors,” in 2016 IEEE Intelligent Vehicles Symposium (IV), June 2016, pp. 864–870.
[18] X. Han, J. Wu, L. Wang, Y. Chen, L. Sehadjii, and H. Shu, “Linear total variation approximate regularized nuclear norm optimization for matrix completion,” Abstract and Applied Analysis, p. 8, 2014.
[19] J. Yang and X. Yuan, “Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization,” Mathematics of Computation, vol. 82, no. 281, pp. 301–329, 2013.
[20] Remcom. Wireless insite. [Online]. Available: http://www.remcom.com/wireless-insite