

# Source Coding Based Millimeter-Wave Channel Estimation With Deep Learning Based Decoding

Yahia Shabara<sup>ID</sup>, *Student Member, IEEE*, Eylem Ekici<sup>ID</sup>, *Fellow, IEEE*,  
and Can Emre Koksal, *Senior Member, IEEE*

**Abstract**—The speed at which millimeter-wave (mmWave) channel estimation can be carried out is critical for the adoption of mmWave technologies. This is particularly crucial because mmWave transceivers are equipped with large antenna arrays to combat severe path losses, which consequently creates large channel matrices, whose estimation may incur significant overhead. This paper focuses on the mmWave channel estimation problem. Our objective is to reduce the number of measurements required to reliably estimate the channel. Specifically, channel estimation is posed as a “source compression” problem in which measurements mimic an encoded (compressed) version of the channel. Decoding the observed measurements, a task which is traditionally computationally intensive, is performed using a deep-learning-based approach, facilitating a high-performance channel discovery. Our solution not only outperforms state-of-the-art compressed sensing methods, but it also determines the lower bound on the number of measurements required for reliable channel discovery.

**Index Terms**—Millimeter-wave, channel estimation, path discovery, sparse recovery, source coding, machine learning.

## I. INTRODUCTION

THE RAPID increase in mobile data traffic has motivated the exploration of mmWave spectrum bands [1]–[4]. While mmWave communication promises orders of magnitude increase in data rates, it both i) suffers from severe path losses [5] and ii) necessitates the use of power-hungry circuits to operate. To overcome these problems, large-gain, highly-directional antenna arrays are proposed as a counter measure to path losses, along with less flexible, yet energy-efficient transceivers that no longer use fully-digital beamforming. Large antenna arrays, however, create channel matrices with large dimensions, which are complex to estimate. When combined with limited transceiver capabilities, large scale channel estimation may take prohibitively long periods. Reducing the number of measurements is thus

a critical step towards facilitating mmWave networks. Fortunately, this does not necessarily degrade the quality of channel estimation due to the sparse nature of mmWave channels; a feature that has been revealed by empirical measurement studies and further adopted by statistical channel models [3], [6], [7].

This work focuses on *the problem of mmWave channel estimation* with the objective of decreasing the number of required measurements. We treat this problem as that of path/beam discovery which is crucial for *initial link establishment* between a transmitter (TX) and a receiver (RX) (also known as *Initial Access*). We solve this problem using a technique inspired by *binary source coding (data compression)*. Although binary codes are natively designed to compress binary data, we provide a foundation for the same codes to be used for compressing complex-valued data, as well. We devise a method to obtain channel measurements such that they resemble a compressed version of the channel matrix. To estimate the channel from the acquired measurements, we train a Deep Neural Network (DNN) that enables very high speed processing. However, training DNNs that jointly process all measurements poses an overwhelming complexity. Thus, we propose a novel computationally-tractable solution that sequentially processes the acquired measurements. This method is unique in the sense that it reduces the problem of estimating the *channel matrix* as a whole into several smaller sub-problems of estimating the individual rows and columns of that matrix. The key contributions of this work are as follows:

- We show that lossless, fixed-rate, linear source codes can be used to design efficient channel measurements that can be *uniquely* mapped to the underlying channels.
- We accurately evaluate the number of measurements needed for reliable channel discovery (as opposed to a mere scaling law). This number is dependent on the compression ratio of the chosen code.
- We present a tight lower bound on the number of measurements needed to reliably discover the channel and provide a solution that achieves this bound.
- We propose a high-performance DNN based measurement-to-channel mapping.
- We show that our solution outperforms the state-of-the-art compressed sensing based solutions and the IEEE 802.11ad beam alignment method.

Manuscript received May 30, 2020; revised December 1, 2020 and March 23, 2021; accepted April 3, 2021. Date of publication April 13, 2021; date of current version July 15, 2021. This work was supported in part by the Computer and Network Systems (CNF) under Grant 1514260, Grant 1618566, Grant 1731698, and Grant 1814923. The associate editor coordinating the review of this article and approving it for publication was A. Alkhateeb. (Corresponding author: Yahia Shabara.)

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: shabara.1@osu.edu; ekici.2@osu.edu; koksal.2@osu.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCOMM.2021.3072999>.

Digital Object Identifier 10.1109/TCOMM.2021.3072999

0090-6778 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

The mmWave channel estimation problem can generally be divided into two intertwined parts. The first is: *how to obtain “good” measurements that can be used to reliably discover the channel?* and the second is: *how to map these measurements to corresponding channel estimates?* Motivated by our proposed solution, we name these two parts “*Channel Encoding*” and “*Measurement Decoding*”, respectively. Encoding and decoding are intertwined because a selection of a specific decoding method often dictates (i) how the measurements are obtained, and (ii) the number of measurements for which this specific decoding method would yield “good” performance. The dissociation of encoding and decoding as two sub-problems can be seen across almost all mmWave channel estimation research, albeit not always explicitly mentioned. This distinction, however, facilitates the identification of key aspects upon which we could improve the quality of channel estimation.

A well-known classification of encoding paradigms is encoding *with* vs. *without* feedback. Non-feedback encoding is better suited for simultaneous multi-user channel estimation, hence is scalable, while feedback-based encoding operates better at low SNR [8]. Different decoding algorithms are also needed for these two types. This paper focuses on encoding without feedback.

The state-of-the-art mmWave channel estimation algorithms rely on compressed Sensing (CS) to reduce the number of channel measurements [9]–[12]. Other approaches include: i) measurements with hierarchical beam patterns that sequentially narrow down the angular direction(s) which contain strong propagation paths, ii) measurements with overlapped beam patterns where each measurement combines signals received from a randomly selected set of angular directions [13], and iii) machine learning based algorithms for sparse recovery of mmWave channels [14]–[17]. Further, in [18] we first introduced the idea of exploiting *binary codes* for tackling mmWave channel estimation. In particular, we exploit the capability of error discovery of channel codes and construct an analogy to path discovery in mmWave channels.

*Notations:*  $x$  is a scalar quantity, while  $\mathbf{x}$  is a vector and  $\mathbf{X}$  is a matrix. The transpose of a matrix is denoted by  $\mathbf{X}^T$ , while  $\mathbf{X}^*$  denotes its conjugate and  $\mathbf{X}^H$  denotes the conjugate transpose.

## II. RELATED WORK

**Initial Access:** The “Initial Access” problem is concerned with finding the angular bearings of one or more propagation paths between a pair of TX and RX nodes, without prior knowledge about previous channel values. In mobile environments, these angular directions are expected to change after Initial Access. “Beam Tracking” methods are commonly used to correct for smaller angular changes and maintain the viability of active link(s) [19], [20]. Nonetheless, due to the narrow beams at both TX and RX, established communication links are prone to blockage (by objects in the communication environment, and even the users themselves). Hence, the initial link establishment stage might need to be repeated multiple times during every communication session. This results in high

overhead to establish coherent beams during the course of the session, if the initial access process is inefficient. This paper focuses on the Initial Access problem.

**Compressed Sensing (CS):** In CS theory, the main objective is to recover an unknown *sparse vector*  $\mathbf{q}^a$  using a small number (compared to the sparse vector dimensions) of linear measurements. Measurements in CS, denoted by  $\mathbf{y}$ , are modeled as  $\mathbf{y} = \mathbf{B}\mathbf{q}^a$ , where  $\mathbf{B}$  is the sensing matrix. Hence,  $\mathbf{B}$  is a linear transformation that amounts to encoding  $\mathbf{q}^a$ . Sparse recovery algorithms, on the other hand, amount to decoding  $\mathbf{y}$ . To obtain “good” measurements (which best preserve the information contained in the channel matrix), the sensing matrix need to be stochastically optimized based on criteria like the  $\text{spark}(\mathbf{B})$  (i.e., minimum number of linearly dependent columns), the mutual coherence and the Restricted Isometry Property.

Since mmWave channel matrices are sparse, and since channel measurements are linear operations, CS became a dominant approach for tackling mmWave channel estimation problems. The main caveat here is that the standard CS problem is that of a sparse vector recovery, while mmWave channel estimation is a sparse matrix recovery. This distinction poses some challenges in tackling mmWave channel estimation under the umbrella of CS. To formulate MIMO channel estimation as a CS problem, a vectorization step is carried out (i.e., columns of matrices are stacked on top of each other to form one long vector). Nonetheless, unlike standard CS problems in which elements of the sensing matrices are directly chosen and optimized, the mmWave sensing matrix is a function of the transmit precoding and receive combining vectors. This adds an extra layer of complexity which is often ignored under the premise that since CS often requires random sensing matrices, then random beamforming is an obvious necessity. However, it is not immediately clear how a specific choice of precoders and combiners would affect the structure of  $\mathbf{B}$ , and therefore, the performance of sparse recovery. Extending the design principles of sensing matrices from core CS theory to mmWave channel estimation is thus not straightforward and remains an open area of research.

Existing research on CS-based mmWave channel estimation relies on *random* arbitrary choices of precoding and combining vectors, e.g. uniformly distributed phase shifts [21], [22]. When this solution is incorporated in mmWave channel estimation, it translates into designing antenna beam patterns of highly irregular shapes (see Fig. 6). Such beam patterns are sensitive to variations of received signal power, thermal noise and resolution of ADCs and phase-shifters. Our proposed source-coding-based solution overcomes these limitations by imposing better, well-structured antenna patterns, where, in each measurement, a specific angular direction is either included (with constant beamforming gain) or is excluded. This provides better resilience to i) the presence of sidelobes, ii) variations in received signal power along any available path(s), iii) channel noise and iv) quantization error of ADCs and phase shifters. Furthermore, the deterministic nature of our source-coding-based measurements allows us to provide theoretical guarantees for channel recovery at a precise number of measurements. The source coding analogy

also allows us to draw theoretical tight lower bounds on the number of measurements.

On the contrary, the number of required measurements in CS is commonly characterized as an order of magnitude. For instance, several state-of-the-art sparse recovery algorithms require  $O(L \log(\frac{n}{L}))$  measurements, where  $n$  is the number of dimensions of the sparse vector and  $L \ll n$  is its sparsity level [9], [11]. This, however, is just a scaling law, which by definition, works in the asymptotic regime and is missing the constant scaling coefficient. Compare this to our solution, which accurately specifies the required number of measurements (based on  $n$  and  $L$ ).

Developing efficient sparse recovery algorithms for CS-based mmWave channel estimation is a rich area of research. Various algorithms have different computational complexities, recovery performance, favorable range of signal to noise ratio (SNR), etc. A comparison between several classes of sparse recovery algorithms is provided in [22]. These include convex relaxation (e.g.  $\ell_1$  norm minimization), greedy iteration (e.g. Orthogonal Matching Pursuit (OMP)) and Bayesian Inference. Other algorithms also include Approximate Message Passing (AMP) [23] and its variants [24], as well as machine learning based sparse recovery [25].

**Machine Learning:** Deep learning is very powerful in extracting patterns from large amounts of data. It has been widely used in problems of computer vision, speech recognition and natural language processing. Recently, it has also been applied to problems in communications [26], including, but not limited to channel estimation [15]–[17]. For instance, in [17] the beamforming vectors at the TX and RX are “learned” based on uplink pilot signals simultaneously received at multiple base stations. The base stations share their received information on a cloud, on which data processing is performed. This idea, however, is critically dependent on a dense deployment of base stations. In [15], [16], deep learning is leveraged to ease the burden of heavy computations that would otherwise be required for measurement processing.

**Coding:** Exploiting source codes for mmWave channel estimation has not been studied before. Our earlier work in [18] drew an analogy between *path discovery* in mmWave channels and *error discovery* in *Linear Block Channel Codes* (LBC). There exists a *duality* between the error discovery problem of channel codes and linear source compression. That is, we can use LBCs as *linear* source compression codes, as well. Nonetheless, the channel coding analogy did not naturally lend itself to characterizing the lower bound on the required number of measurements. This paper differs from [18] in the following:

- Channel measurements are envisaged as compressed versions of the channel, which are obtained based on lossless, fixed-rate, linear source codes.
- The lower bound on the achievable number of measurements is accurately characterized using the entropy of the direction of the strong reflectors (in a stochastic spatial model). This not only provides a precise metric to quantify the efficiency of a used code, but it also provides a benchmark for evaluating other measurement schemes.

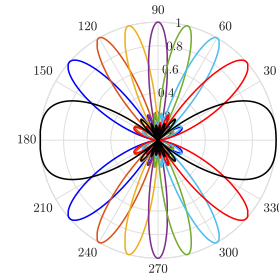


Fig. 1. Antenna sectors corresponding to angular directions  $\{d_i\}_{i=1}^8$ .

- A DNN-based measurement decoding is proposed and evaluated against the more computationally complex “search” method of [18].
- A comparison to compressed sensing based mmWave channel estimation is provided, demonstrating the superiority of our proposed approach.

**Hashed Beams:** An idea of direction inclusion/exclusion used to generate antenna patterns was adopted in [13]. Specifically, every measurement combines signals coming from a *randomly* chosen set of angular directions. This describes the encoding part and it resembles a *random* binary code. For decoding, a threshold-based decision determines whether a strong propagation path exists (if a path exists, it lies at one of the directions included in this measurement). The direction which was most frequently included in the measurements that revealed a strong path is declared as the angular direction of the strongest channel path. This method discovers one path, and requires  $O(L \log(n))$  measurements. In our proposed approach, however, the angular directions whose respective beams are overlapped are *precisely determined* using a carefully chosen code. We also use an elaborate decoding method that is capable of discovering multiple paths. It also guarantees a lower number of measurements since a randomly chosen code is not expected to outperform a carefully designed one.

### III. MOTIVATING EXAMPLE

Consider an RX equipped with an antenna array which can form 8 distinct beams. These 8 beams divide the angular space into resolvable directions, i.e.,  $d_1, d_2, \dots, d_8$ , as shown in Fig. 1. The RX needs to establish a Line of Sight (LoS) communication link with TX. This requires some sort of “searching” over the angular space at both TX and RX. For ease of illustration, let us reduce the link establishment problem to that of “Angle of Arrival (AoA) discovery” at RX by assuming that TX transmits its signals omnidirectionally. Let the path gain of LoS be denoted by  $\alpha$ , which can take arbitrary values. For simplicity of notation, let us assume  $\alpha = 1$ .

Our *objective* is: Find the specific direction  $d_{i^*}$  which contains the LoS path to TX using the *least possible number of measurements*. To do so, we envisage the measurement process as *lossless, fixed-rate channel compression*. This enables *harnessing the power of source compression codes to minimize the number of measurements*. It also enables deriving lower bounds on the number of measurements, using which

TABLE I  
CHANNEL MEASUREMENTS  $\mathbf{y}^s$  CORRESPONDING TO ALL  $\mathbf{q}^\alpha \in \mathcal{Q}^\alpha$

Angular Channel $\mathbf{q}^{\alpha T}$	Channel measurement $\mathbf{y}^{sT}$
$d_0$	[0 0 0 0 0 0 0 0]
$d_1$	[1 0 0 0 0 0 0 0]
$d_2$	[0 1 0 0 0 0 0 0]
$d_3$	[0 0 1 0 0 0 0 0]
$d_4$	[0 0 0 1 0 0 0 0]
$d_5$	[0 0 0 0 1 0 0 0]
$d_6$	[0 0 0 0 0 1 0 0]
$d_7$	[0 0 0 0 0 0 1 0]
$d_8$	[0 0 0 0 0 0 0 1]

we can accurately find the LoS (or conclude it is blocked). We propose a measurement approach which has a *predetermined* measurement sequence that (1) does not require feedback and (2) is capable of finding the LoS path, no matter in which  $d_i$  it exists. Therefore, a constant number of measurements,  $m$ , is needed for all  $d_i$ .

The key idea of LoS discovery using non-feedback linear source coding is to: 1) Construct a *binary* codebook that represents the angular channel, 2) Find a proper fixed-rate linear source code that losslessly compresses all codewords in that codebook, and 3) Use this code to design the measurements. These steps can be elucidated as follows: **i)** Constructing the codebook is as easy as finding all possible *binary* vectors that represent the LoS position. Since  $\alpha = 1$ , this codebook is exactly the set of all possible channel vectors. Let the channel between TX and RX be denoted by  $\mathbf{q}^\alpha$ , and let  $\mathcal{Q}^\alpha$  be the set of all possible channels. The channel  $\mathbf{q}^\alpha \in \mathcal{Q}^\alpha$  has 8 components; each one represents the path gain corresponding to a unique angular sector as shown in Fig. 1. Table I shows all possible  $\mathbf{q}^\alpha$  in our setup (for arbitrary gain values, simply replace the '1's in Table I with  $\alpha$ ). **ii)** Choose the linear source code, denoted by its generator matrix  $\mathbf{G}$  as:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (1)$$

To compress  $\mathbf{q}^\alpha$ , we simply need to find the matrix multiplication  $\mathbf{y}^s = \mathbf{G}\mathbf{q}^\alpha$  (see Table I). **iii)** Design the measurements such that  $\mathbf{y}^s$  is imitated by the measurement results. **This is done by beamforming at RX.** Notice that the  $i^{\text{th}}$  measurement, i.e.,  $y_i^s$  (the  $i^{\text{th}}$  component of  $\mathbf{y}^s$ ) is the multiplication of the  $i^{\text{th}}$  row of  $\mathbf{G}$  by  $\mathbf{q}^\alpha$ . Mathematically, this is just adding all elements  $q_j^\alpha$  of  $\mathbf{q}^\alpha$  which corresponds to  $g_{i,j} = 1$ , ( $g_{i,j}$  is the element at row  $i$ , and column  $j$  in  $\mathbf{G}$ ). That is

$$y_i^s = \sum_{j=1}^8 q_j^\alpha \times g_{i,j} = \sum_{j: g_{i,j}=1} q_j^\alpha. \quad (2)$$

Hence, measurement  $i$  should only contain the directions  $d_j$  whose corresponding  $g_{i,j}$  equals 1, and exclude the rest (Notice that we can map the  $i^{\text{th}}$  row of  $\mathbf{G}$  to the  $i^{\text{th}}$  measurement and the  $j^{\text{th}}$  column to the  $j^{\text{th}}$  sector (direction  $d_j$ )). Essentially, this means that in each of the measurements  $y_i^s$ ,

we combine the signals received at a specific set of AoA directions. This can be realized by carefully shaping the antenna pattern using beamforming. Fig. 2 highlights this process for the 1<sup>st</sup> measurement in which only the direction  $d_1, d_5$  and  $d_8$  are included. *The measurement results  $\forall d_i$  are shown in Table I.* Note that the number of required measurements is 4 for all  $d_i$ .

**Lower Bound:** A fundamental question that arises here is: *Can we find a better fixed-rate, lossless source code (other than the one given in Eq. (1)) that would produce fewer measurements, and hence increase the efficiency of the measurement process?* To answer this question, we need to find **the minimum expected number of measurements** required to discover the channel using our proposed source coding solution. This number is identical to the minimum average code length (over all fixed-rate, lossless codes). The minimum average code length is well-known to be lower bounded by the **Shannon Entropy**; denoted by  $H_2$  and defined as

$$H_2(\mathbf{q}^\alpha) = \sum_{\mathbf{q}^\alpha \in \mathcal{Q}^\alpha} \mathbb{P}(\mathbf{q}^\alpha) \log_2 \left( \frac{1}{\mathbb{P}(\mathbf{q}^\alpha)} \right) \quad (3)$$

Calculating  $H_2(\mathbf{q}^\alpha)$  requires knowledge of the probability distribution  $\mathbb{P}(\mathbf{q}^\alpha)$ . Fixed-rate codes, however, do not account for the frequency of  $\mathbf{q}^\alpha$  (hence, the mapping to equal-length codes). By limiting the space of codes to be over fixed-rate codes, we can improve the bound to be

$$\lceil \log_2(|\mathcal{Q}^\alpha|) \rceil \geq H_2(\mathbf{q}^\alpha) \quad (4)$$

where  $|\mathcal{Q}^\alpha| = 9$  (recall that there exists 9 possible scenarios for  $\mathbf{q}^\alpha$  as shown in Table I). This tighter bound is obtained by assuming a *Uniform* distribution, which is the entropy maximizing distribution, over the channel space  $\mathcal{Q}^\alpha$ . Eq. (4) reveals that our chosen code achieves the lower bound of 4 measurements. We provide a formal discussion on the lower bound in Section V-D.

*Remark 1: This Motivating Example only dealt with a simplified channel model, with only one channel path and a fixed path gain of  $\alpha = 1$ . However, in the rest of the paper, we will consider generalized channel models with possibly several paths of arbitrary path gain values, i.e.,  $\alpha \in \mathbb{C}$ .*

#### IV. SYSTEM MODEL

We consider point-to-point mmWave channels with  $n_t$  and  $n_r$  antennas at TX and RX, respectively. Antennas at TX and RX form Uniform Linear Arrays (ULA). Generalization to Uniform Planar Arrays is straightforward but not considered in this paper for simplicity. Every antenna element is connected to a phase-shifter and a low-power variable-gain amplifier (VGA).<sup>1</sup> On the TX side, a single RF chain feeds its ULA through an  $n_t$ -way power splitter, while on the RX side, the outputs of the ULA, after being processed by amplifiers and phase-shifters, are then linearly combined using an adder

<sup>1</sup>The use of VGAs in analog transceivers is common in practice. For instance, in IEEE 802.11ad [27] both phase and amplitude components are used to specify antenna weights, and commercial devices like Wilocity Wil6200 offer this capability. VGAs are also used along with phase-shifters in practice to help compensate for their phase-dependent insertion loss [28].

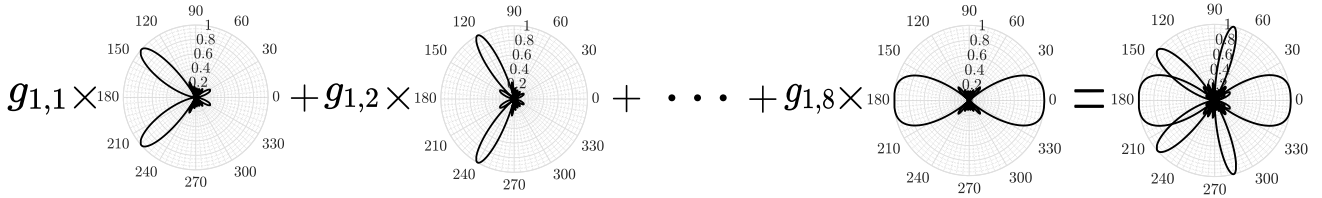


Fig. 2. Generating the beam pattern required for the  $1^{st}$  measurement ( $y_1^s$ ).

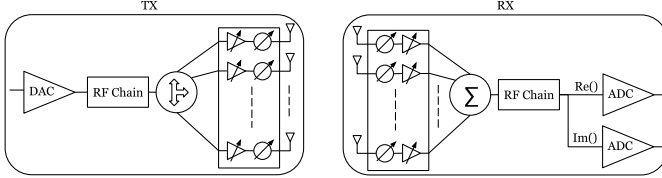


Fig. 3. Transceiver architecture: At TX, an  $n_t$ -way power splitter divides the transmit signal which is then passed through variable-gain amplifiers and phase-shifters. A single DAC is required since TX sends real valued signals. At RX, the acquired signal is passed through a similar network of power amplifiers and phase-shifters before being combined and fed to a single RF chain. Two ADCs are required to obtain I/Q components of received signals.

and fed through to a single RF chain with in-phase (I) and quadrature (Q) channels. Two mid-tread ADCs with  $2^b+1$  levels are used to quantize the I and Q components of the received signal. The term  $b$  loosely denotes the number of bits that describe the ADC resolution. Fig. 3 depicts the transceiver architecture.

We assume single-tap channels where all channel paths have just one significant tap. We also adopt a channel clustering model where paths between TX and RX form clusters in the angular domain [1], [6]. Let  $L$  denote the number of available channel clusters. Due to the sparse nature of mmWave channels, only a limited number of clusters exist,<sup>2</sup> where  $L \ll n_r, n_t$ . Since distinct paths within each cluster cannot typically be resolved, we assume that each cluster contains only one path. Each channel path (e.g.,  $l^{th}$  path) is attributed with an AoD  $\theta_l$ , an AoA  $\phi_l$  and a path gain  $\alpha_l$ . Let  $\alpha_l^b \in \mathbb{C}$  denote the baseband path gain such that

$$\alpha_l^b = \alpha_l \sqrt{n_r n_t} e^{-j \frac{2\pi \rho_l}{\lambda_c}}, \quad (5)$$

where  $\rho_l$  is the path length and  $\lambda_c$  is the carrier wavelength. We define the *directional cosines* of the AoD and AoA of the  $l^{th}$  path as  $\Omega_{tl} \triangleq \cos(\theta_l)$  and  $\Omega_{rl} \triangleq \cos(\phi_l)$ , respectively. The transmit and receive spatial signatures at an arbitrary directional cosine  $\Omega$  is denoted by  $\mathbf{e}_t(\Omega)$  and  $\mathbf{e}_r(\Omega)$ , respectively. We define  $\mathbf{e}_t(\Omega)$  (and similarly  $\mathbf{e}_r(\Omega)$ ) as:

$$\mathbf{e}_t(\Omega) = \frac{1}{\sqrt{n_t}} \times \left( 1, e^{-j2\pi\Delta_t\Omega}, e^{-j2\pi2\Delta_t\Omega}, \dots, e^{-j2\pi(n_t-1)\Delta_t\Omega} \right)^T \quad (6)$$

<sup>2</sup>Prior knowledge about the number of clusters can be obtained from statistical channel information, which in turn are obtained from channel measurement campaigns. For instance, measurements carried out in New York City revealed that an average number of 2 or 3 clusters exists in mmWave channels at 28 and 73 GHz [3].

where  $\Delta_t$  and  $\Delta_r$  are the antenna separations at TX and RX ULAs, normalized by  $\lambda_c$ .

Let  $\mathbf{Q} \in \mathbb{C}^{n_r \times n_t}$  denote the channel matrix such that

$$\mathbf{Q} = \sum_{l=1}^L \alpha_l^b \mathbf{e}_r(\Omega_{rl}) \mathbf{e}_t^H(\Omega_{tl}). \quad (7)$$

The corresponding angular channel of  $\mathbf{Q}$ , whose rows and columns divide the channel into resolvable RX and TX angular bins, respectively, is denoted by  $\mathbf{Q}^a$  and can be obtained as:

$$\mathbf{Q}^a = \mathbf{U}_r^H \mathbf{Q} \mathbf{U}_t. \quad (8)$$

If  $n_t$  or  $n_r$  equals 1,  $\mathbf{Q}$  and  $\mathbf{Q}^a$  are reduced to vectors which we denote by  $\mathbf{q}$  and  $\mathbf{q}^a$ , respectively. The matrices  $\mathbf{U}_t$  and  $\mathbf{U}_r$  are the transmit and receive *unitary Discrete Fourier Transform (DFT)* matrices whose columns form an orthonormal basis for the transmit and receive signal spaces  $\mathbb{C}^{n_t}$  and  $\mathbb{C}^{n_r}$ , respectively. The definition of  $\mathbf{U}_t$  (likewise for  $\mathbf{U}_r$ ) is given as [29, Chapter 7.3.4]

$$\mathbf{U}_t \triangleq \left( \mathbf{e}_t(0) \ \mathbf{e}_t\left(\frac{1}{L_t}\right) \ \dots \ \mathbf{e}_t\left(\frac{n_t-1}{L_t}\right) \right), \quad (9)$$

where  $L_t = n_t \Delta_t$  ( $L_r = n_r \Delta_r$ ) denote the length of the TX (RX) antenna array, normalized by  $\lambda_c$ .

Similar to [11], [30], [31], we assume perfect sparsity where channel paths lie along AoD and AoA directions defined in  $\mathbf{U}_t$  and  $\mathbf{U}_r$ . Hence, each path only contributes to a single component of  $\mathbf{Q}^a$ . Thus, only  $L$  (possibly less) non-zero components in  $\mathbf{Q}^a$  exists. The baseband model is

$$\mathbf{u}_b = \mathbf{Q} \mathbf{x} + \mathbf{n} \quad (10)$$

where  $\mathbf{u}_b$  is the received vector at RX front-end while  $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I}_{n_r})$  is an i.i.d. complex Gaussian noise vector. TX sends pilot symbols  $s$ , with power  $P$ , which are processed using precoders  $\mathbf{f}_j \in \mathbb{C}^{n_t}$  to obtain the transmit vectors  $\mathbf{x} = \mathbf{f}_j s$ . Hence, the transmit SNR is

$$\text{SNR} \triangleq \frac{P}{N_0} \times \mu, \quad (11)$$

where  $\mu$  is the average path loss (which depends on the carrier frequency, atmospheric conditions, average distance between TX and RX). Note that SNR and  $\mu$  are not path dependent. The rx-combining vectors  $\mathbf{w}_i \in \mathbb{C}^{n_r}$  are used to obtain the received symbols  $u_{i,j}$  such that

$$u_{i,j} = \mathbf{w}_i^H \mathbf{Q} \mathbf{f}_j s + \mathbf{w}_i^H \mathbf{n} \quad (12)$$

where  $i \in \{1, \dots, m_r\}$ ,  $j \in \{1, \dots, m_t\}$ . Finally, a quantized version  $u_{i,j}^s$  of  $u_{i,j}$  is obtained such that

$$u_{i,j}^s = \left[ \mathbf{w}_i^H \mathbf{Q} \mathbf{f}_j s + \mathbf{w}_i^H \mathbf{n} \right]_+, \quad (13)$$

where  $[\cdot]_+$  represents the quantization function. The noise component, normalized by  $\|\mathbf{w}_i\|$  has a complex Gaussian distribution, i.e.,  $\frac{\mathbf{w}_i^H \mathbf{n}}{\|\mathbf{w}_i\|} \sim \mathcal{CN}(0, N_0)$ .

Let  $y_{i,j}^s = \mathbf{w}_i^H \mathbf{Q} \mathbf{f}_j s$  denote the error-free measured symbols and let  $z_{i,j} = u_{i,j}^s - y_{i,j}^s$  denote the *measurement error* which includes both **channel noise** and **quantization error**.

### A. Problem Formulation

The problem we need to solve is to minimize the number of measurements  $m = m_t \times m_r$  such that  $\mathbf{Q}$  can be reliably reconstructed. This problem can be mathematically stated as:

$$P1: \underset{\mathbf{w}_i, \mathbf{f}_j, \mathcal{D}}{\text{minimize}} \quad m_t \times m_r \quad (14a)$$

$$\text{subject to} \quad y_{i,j}^s = \mathbf{w}_i^H \mathbf{Q} \mathbf{f}_j, \quad (14b)$$

$$\mathcal{D}(\{y_{i,j}^s\}) = \mathbf{Q}^a \quad (14c)$$

Note that  $y_{i,j}^s$  exists  $\forall i, j \in \{1, \dots, m_r\} \times \{1, \dots, m_t\}$ . That is, measurements are taken using all combinations of  $\mathbf{f}_j$  and  $\mathbf{w}_i$ . We also use  $s = 1$ . The design variables are the tx-precoders  $\mathbf{f}_j$ , the rx-combiners  $\mathbf{w}_i$  and the decoding function  $\mathcal{D}$ . We do not explicitly consider the impact of errors in this formulation but its effect will be studied in Section V-E. Note also that due to the use of VGAs at each antenna element, the constant modulus constraint on  $\mathbf{f}_j$  and  $\mathbf{w}_i$ , that is often incorporated in analog beamforming designs, is not needed.

## V. SOURCE-CODING-BASED MEASUREMENTS

In this section, we formally introduce *mm-wave beam discovery* as a source coding problem. We initially focus on channels with single-transmit, multiple-receive antennas. Specifically, we provide the conditions under which a chosen fixed-rate source code can be used to uniquely “encode” channel vectors in  $\mathbb{C}^{n_r}$  into measurement vectors of fewer components. This setting is **identical to that of multiple-transmit, single-receive antennas**. In Section VI, we show how to use DNNs to “decode” the measurements and obtain an estimate for the observed channel. Then, in Section VII, we consider general channels with multiple TX and RX antennas. Now, let us start with the following discussion on source codes.

### A. Source Codes

Let  $C$  be a binary *linear* source code with encoding and decoding functions denoted by  $\mathcal{E}_{\mathbb{F}_2}$  and  $\mathcal{D}_{\mathbb{F}_2}$ , respectively. We refer to  $C$  as the encoding-decoding function pair  $(\mathcal{E}_{\mathbb{F}_2}, \mathcal{D}_{\mathbb{F}_2})$ . The subscript  $\mathbb{F}_2$  denotes the finite field of two elements  $0_{\mathbb{F}_2}$  and  $1_{\mathbb{F}_2}$  (also referred to as  $GF(2)$ ) over which the code  $C$  is defined. Later on, we will drop the subscripts to simplify notation as long as they can be inferred from the context.

*Definition 1 (Linear Source Code):* A source code  $C$  whose encoding function  $\mathcal{E}_{\mathbb{F}_2}$  is a linear function of the source sequences is called a **linear source code**.

Let  $\mathbf{s}$  be a source sequence of length  $n$  where  $\mathbf{s} \in \mathcal{S} \subseteq \{0_{\mathbb{F}_2}, 1_{\mathbb{F}_2}\}^n$ , and let  $\mathbf{c}_s \in \mathcal{I}_S \subseteq \{0_{\mathbb{F}_2}, 1_{\mathbb{F}_2}\}^m$  be its associated binary representation under  $C$  where  $\mathcal{I}_S$  is the image of  $\mathcal{S}$

under  $\mathcal{E}_{\mathbb{F}_2}$ . Thus, using a linear source code  $C$ , we can find the representation of  $\mathbf{s}$  under  $C$  using

$$\mathbf{c}_s = \mathbf{G} \mathbf{s}, \quad (15)$$

where  $\mathbf{G} \in \{0_{\mathbb{F}_2}, 1_{\mathbb{F}_2}\}^{m \times n}$  is called the **generator matrix**. Note that linearity guarantees fixed-rate since the code length is a constant value (equals the number of rows of  $\mathbf{G}$ ).

The decoding function  $\mathcal{D}_{\mathbb{F}_2}$ , maps sequences  $\mathbf{c}_s$  to a corresponding source sequence  $\hat{\mathbf{s}} \in \hat{\mathcal{S}} \subseteq \{0_{\mathbb{F}_2}, 1_{\mathbb{F}_2}\}^n$ . Suppose that  $\mathcal{S}$  is the set of all sequences such that if  $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}$ , we have that  $\mathbf{s}_1 \neq \mathbf{s}_2 \iff \mathbf{c}_{s_1} \neq \mathbf{c}_{s_2}$ . In other words,  $\mathcal{E}_{\mathbb{F}_2}: \mathcal{S} \rightarrow \mathcal{I}_S$  is injective (one-to-one). Consequently, if we define the function  $\mathcal{D}_{\mathbb{F}_2}$  over  $\mathcal{I}_S$  as the inverse function of  $\mathcal{E}_{\mathbb{F}_2}$ , i.e.,  $\mathcal{E}_{\mathbb{F}_2}^{-1} \triangleq \mathcal{D}_{\mathbb{F}_2}: \mathcal{I}_S \rightarrow \mathcal{S}$ , then we have that  $\hat{\mathbf{s}} = \mathcal{D}_{\mathbb{F}_2}(\mathcal{E}_{\mathbb{F}_2}(\mathbf{s})) = \mathbf{s}, \forall \mathbf{s} \in \mathcal{S}$ .

### B. MmWave Beam Discovery

Let  $\mathbf{q}^a \in \mathbb{C}^{n_r}$  denote the angular **channel vector** between TX and RX. Define  $\mathbf{q}_s^a \in \{0, 1\}^{n_r}$  to be the **support vector** associated with  $\mathbf{q}^a$  such that  $\mathbf{q}_s^a = (q_{s_1}^a \ q_{s_2}^a \ \dots \ q_{s_{n_r}}^a)^T$  where  $q_{s_i}^a = 1$  if  $q_i^a \neq 0$  and  $q_{s_i}^a = 0$  otherwise. More generally, a support vector can be defined as:

*Definition 2 (Support vector):* The support vector  $\mathbf{v}_s$  associated with an arbitrary  $n$ -dimensional vector  $\mathbf{v} \in \mathbb{C}^n$  is a binary vector of the same size that identifies the non-zero components of  $\mathbf{v}$  and whose components,  $v_{s_i}$ , are defined as  $v_{s_i} = 1$  if  $v_i \neq 0$  and  $v_{s_i} = 0$  if  $v_i = 0$ .

We further define the set of non-zero indexes  $\mathcal{X}_v$  of an arbitrary vector  $\mathbf{v}$  as follows:

*Definition 3 (Set of Non-Zero Indexes  $\mathcal{X}_v$ ):* For any arbitrary  $n$ -dimensional vector  $\mathbf{v}$ , we define  $\mathcal{X}_v$  as the set of indexes of its non-zero components, i.e.,  $\mathcal{X}_v = \{i | v_i \neq 0, 0 \leq i \leq n-1\}$ .

Hence, if  $\mathbf{v}_s$  is the support vector corresponding to  $\mathbf{v}$ , then we have that  $\mathcal{X}_v = \mathcal{X}_{\mathbf{v}_s}$ , since  $v_i = 0 \iff v_{s_i} = 0$ . Now, let  $\mathcal{Q}^a$  be the set containing all possible channel vectors  $\mathbf{q}^a$ . Also let  $\mathcal{Q}_s^a$  be the set of all support vectors  $\mathbf{q}_s^a$  such that their corresponding channels  $\mathbf{q}^a \in \mathcal{Q}^a$ . An interesting behavior we have for these sets is as follows: If we have a channel  $\mathbf{q}_1^a$  whose support vector  $\mathbf{q}_{s_1}^a \in \mathcal{Q}_s^a$ , then removing any non-zero component(s) from  $\mathbf{q}_1^a$  (due to blockage for example) would still yield a valid channel  $\mathbf{q}_2^a \in \mathcal{Q}^a$ , whose support vectors  $\mathbf{q}_{s_2}^a$  also belongs to  $\mathcal{Q}_s^a$ . We call this the **inclusion property**.

*Definition 4: [Inclusion Properties of  $\mathcal{Q}_s^a$ ]*

- (i) Let  $\mathbf{q}_{s_1}^a, \mathbf{q}_{s_2}^a \in \{0, 1\}^{n_r}$  such that  $\mathcal{X}_{\mathbf{q}_{s_2}^a} \subseteq \mathcal{X}_{\mathbf{q}_{s_1}^a}$ . If  $\mathbf{q}_{s_1}^a \in \mathcal{Q}_s^a$ , then  $\mathbf{q}_{s_2}^a \in \mathcal{Q}_s^a$ .
- (ii)  $\mathbf{0} \in \mathcal{Q}_s^a$ . In fact, this is a consequence of property 4 above since for any  $\mathbf{q}_s^a \in \mathcal{Q}_s^a$ , we have that  $\mathcal{X}_0 = \emptyset \subseteq \mathcal{X}_{\mathbf{q}_s^a}$ .

Now, we are ready to present the theorem that establishes the conditions that need to be satisfied by a linear source code so that each possible channel  $\mathbf{q}^a$  would result in a **unique** measurement vector  $\mathbf{y}^s$ . Impairments under noise are not addressed in this theorem.

*Theorem 1: Consider a binary linear source code  $C$  whose encoding function  $\mathcal{E}$  (defined by the binary generator matrix  $\mathbf{G}$ ) is an injective function defined over  $\mathcal{Q}_s^a \in \{0, 1\}^{n_r}$ . If we consider  $\mathbf{G}$  to be defined over the complex field, then*

for all channel vectors  $\mathbf{q}_1^a, \mathbf{q}_2^a \in \mathcal{Q}^a \subseteq \mathbb{C}^{n_r}$  we have  $\mathbf{q}_1^a \neq \mathbf{q}_2^a$  if and only if  $\mathbf{G}\mathbf{q}_1^a = \mathbf{y}_1^s \neq \mathbf{y}_2^s = \mathbf{G}\mathbf{q}_2^a$ .

*Proof:* Let  $\mathbf{q}_1^a, \mathbf{q}_2^a \in \mathcal{Q}^a$ , and let  $\mathbf{y}_i^s = \mathbf{G}\mathbf{q}_i^a$ . Now, assume that  $\mathbf{q}_1^a \neq \mathbf{q}_2^a$ . Then, we have that

$$\mathbf{y}_1^s - \mathbf{y}_2^s = \mathbf{G}\mathbf{q}_1^a - \mathbf{G}\mathbf{q}_2^a = \mathbf{G}(\underbrace{\mathbf{q}_1^a - \mathbf{q}_2^a}_{=\mathbf{v}}) = \mathbf{G}\mathbf{v} \quad (16)$$

$$= \sum_{i=1}^{n_r} v_i \times \mathbf{g}_i = \sum_{i \in \mathcal{X}_v} v_i \times \mathbf{g}_i \quad (17)$$

where  $\mathbf{g}_i$  is the  $i^{\text{th}}$  column of  $\mathbf{G}$ . **To show that  $\mathbf{y}_1^s - \mathbf{y}_2^s \neq \mathbf{0}$ , we need to show that all vectors  $\mathbf{g}_i \forall i \in \mathcal{X}_v$ , are linearly independent.** Otherwise, if such vectors  $\mathbf{g}_i$  are linearly **dependent**, then  $\exists v_i \in \mathbb{R}$  for  $i \in \mathcal{X}_v$  such that  $\mathbf{y}_1^s - \mathbf{y}_2^s = \mathbf{G}\mathbf{v} = \mathbf{0}$ .

In fact, we can show a stronger statement: “all vectors  $\mathbf{g}_i \forall i \in \mathcal{X}_{\mathbf{q}_1^a} \cup \mathcal{X}_{\mathbf{q}_2^a} \supseteq \mathcal{X}_v$ , are linearly independent”. Note that  $\mathcal{X}_{\mathbf{q}_1^a}$  and  $\mathcal{X}_{\mathbf{q}_2^a}$  are the sets of indexes of the non-zero components of  $\mathbf{q}_1^a$  and  $\mathbf{q}_2^a$ , respectively (recall Definition 3) and that  $\mathcal{X}_{\mathbf{q}_1^a} = \mathcal{X}_{\mathbf{q}_{s_1}^a}$  and  $\mathcal{X}_{\mathbf{q}_2^a} = \mathcal{X}_{\mathbf{q}_{s_2}^a}$ .

- First, let us show that  $\mathcal{X}_v$  is a subset of  $\mathcal{X}_{\mathbf{q}_1^a} \cup \mathcal{X}_{\mathbf{q}_2^a}$ . Since  $v_i = q_{1,i}^a - q_{2,i}^a \forall 1 \leq i \leq n_r$ , then  $q_{i,1}^a = q_{i,2}^a = 0 \implies v_i = 0$ . Therefore, we have

$$\mathcal{X}_{\mathbf{q}_1^a}^c \cap \mathcal{X}_{\mathbf{q}_2^a}^c = \{\mathcal{X}_{\mathbf{q}_1^a} \cup \mathcal{X}_{\mathbf{q}_2^a}\}^c \subseteq \mathcal{X}_v^c \quad (18)$$

Then, by taking the complements of both sides we obtain the required result (note that  $\{\cdot\}^c$  denotes a set complement).

- Second, we show that vectors in the set  $\mathcal{G} \triangleq \{\mathbf{g}_i | i \in \mathcal{X}_{\mathbf{q}_1^a} \cup \mathcal{X}_{\mathbf{q}_2^a}\}$  are linearly independent over  $\mathbb{F}_2$ : Assume towards contradiction that  $\mathcal{G}$  is linearly dependent over  $\mathbb{F}_2$ . Hence, there exists a set  $\mathcal{G}_D \subseteq \mathcal{G}$  such that any  $\mathbf{g}_{i_0} \in \mathcal{G}_D$  can be written as a linear combination of all other vectors in  $\mathcal{G}_D$ , i.e.,

$$\mathbf{g}_{i_0} = \sum_{\substack{j:j \neq i_0 \\ \mathbf{g}_j \in \mathcal{G}_D}} \mathbf{g}_j \pmod{2}. \quad (19)$$

Note that over  $\mathbb{F}_2$ , we can assume, without loss of generality (W.L.O.G.), that the coefficients of the linear combination above are 1's. Hence, we have that

$$\sum_{j:\mathbf{g}_j \in \mathcal{G}_D} \mathbf{g}_j \pmod{2} = \mathbf{0} \quad (20)$$

Next, assume that  $\exists \mathbf{q}_{s_3}^a, \mathbf{q}_{s_4}^a \in \mathcal{Q}_s^a$  such that  $\mathcal{X}_{\mathbf{v}_s} = \{j | \mathbf{g}_j \in \mathcal{G}_D\}$  where  $\mathbf{v}_s = \mathbf{q}_{s_3}^a - \mathbf{q}_{s_4}^a \pmod{2}$ . Then, since  $\mathbf{G}$  is injective over  $\mathcal{Q}_s^a$ , then we have that

$$\mathbf{G}\mathbf{v}_s \pmod{2} = \sum_{j \in \mathcal{X}_{\mathbf{v}_s}} \mathbf{g}_j \pmod{2} \quad (21)$$

$$= \sum_{j:\mathbf{g}_j \in \mathcal{G}_D} \mathbf{g}_j \pmod{2} \neq \mathbf{0} \quad (22)$$

$$\iff \mathbf{v}_s \pmod{2} \neq \mathbf{0} \quad (23)$$

But, if  $\mathcal{G}_D$  is non-empty, then  $\mathbf{v}_s \neq \mathbf{0}$ . Hence, we arrive at a contradiction to Eq. (19). Therefore, the set  $\mathcal{G}$  is linearly independent over  $GF(2)$ . It remains to show that such  $\mathbf{q}_{s_3}^a$  and  $\mathbf{q}_{s_4}^a$  indeed exist. Let us construct  $\mathbf{q}_{s_3}^a$  as follows:

First, let  $\mathbf{q}_{s_3}^a = \mathbf{q}_{s_1}^a$ , then, reset its  $i^{\text{th}}$  component to 0 ( $q_{s_3,i}^a = 0$ ) if  $\mathbf{g}_i \notin \mathcal{G}_D$ . Similarly, set  $\mathbf{q}_{s_4}^a = \mathbf{q}_{s_2}^a$ , then, reset the  $i^{\text{th}}$  component to 0 ( $q_{s_4,i}^a = 0$ ) if  $\mathbf{g}_i \notin \mathcal{G}_D$  OR if  $q_{s_1,i}^a = 1$ . Then, by construction, we have  $\mathcal{X}_{\mathbf{q}_{s_3}^a} \subseteq \mathcal{X}_{\mathbf{q}_{s_1}^a}$  and  $\mathcal{X}_{\mathbf{q}_{s_4}^a} \subseteq \mathcal{X}_{\mathbf{q}_{s_2}^a}$ . Hence, by the inclusion property (recall Definition 4) we have  $\mathbf{q}_{s_3}^a, \mathbf{q}_{s_4}^a \in \mathcal{Q}_s^a$  since both  $\mathbf{q}_{s_1}^a, \mathbf{q}_{s_2}^a \in \mathcal{Q}_s^a$ . Also, it is easy to see that  $q_{s_3,j}^a - q_{s_4,j}^a \pmod{2} = 1 \forall j : \mathbf{g}_j \in \mathcal{G}_D$ .

- Third, by Lemma 1 below, the set  $\mathcal{G}$ , now taken over  $\mathbb{R}$ , is linearly independent.

Therefore, in Eq. (17), it follows that  $\mathbf{y}_1^s - \mathbf{y}_2^s \neq \mathbf{0}$  if and only if  $\mathbf{q}_1^a - \mathbf{q}_2^a \neq \mathbf{0}$ , which concludes the proof.  $\square$

*Lemma 1:* Any set of  $n$ -dimensional linearly independent vectors over  $\mathbb{F}_2$  are also linearly independent over  $\mathbb{C}$  if we interpret their  $0_{\mathbb{F}_2}$  and  $1_{\mathbb{F}_2}$  components to be real scalars.

The proof is provided in Appendix A.

### C. Beamforming Design

Now, we focus our attention on the design of beamforming vectors  $\mathbf{w}_i$ , such that the measurement vector  $\mathbf{y}^s$  is such that  $\mathbf{y}^s = \mathbf{G}\mathbf{q}^a$ . Obviously,  $\mathbf{w}_i$  depends on the chosen source code. Specifically, we want  $\mathbf{w}_i$  to satisfy

$$\mathbf{w}_i^H \mathbf{q} = \mathbf{y}_i^s = \mathbf{g}_i \mathbf{q}^a = \sum_{j=1}^{n_r} g_{i,j} q_j^a \quad (24)$$

where  $\mathbf{g}_i$  is the  $i^{\text{th}}$  row of  $\mathbf{G}$ , and  $g_{i,j}$  is the  $j^{\text{th}}$  element of  $\mathbf{g}_i$ . Recall that if  $n_r$  antennas exist at RX, then there exists  $n_r$  resolvable angular directions. Let us call these directions  $d_1, d_2, \dots, d_{n_r}$ . We want  $\mathbf{w}_i$  to combine the received signal components at specific angular directions. Those angular directions are determined by  $\mathbf{g}_i = (g_{i,1}, g_{i,2}, \dots, g_{i,n_r})$ . Specifically, we want  $\mathbf{w}_i$  to include the signal at directions  $d_j$  for all  $j$  such that  $g_{i,j} = 1$ .

Recall that  $\mathbf{q} = \mathbf{U}_r \mathbf{q}^a$ . Hence, we can rewrite Eq. (24) as  $\mathbf{w}_i^H \mathbf{q} = \mathbf{w}_i^H \mathbf{U}_r \mathbf{q}^a = \mathbf{g}_i \mathbf{q}^a$ . Thus, we need to design  $\mathbf{w}_i$  such that  $\mathbf{w}_i^H \mathbf{U}_r = \mathbf{g}_i$ , which can be rewritten as:

$$\mathbf{w}_i^H \left( \mathbf{e}_r(0) \mathbf{e}_r \left( \frac{1}{L_r} \right) \dots \mathbf{e}_r \left( \frac{n_r-1}{L_r} \right) \right) = \mathbf{g}_i. \quad (25)$$

Since the columns of  $\mathbf{U}_r$  constitute an orthonormal basis, a very simple design of  $\mathbf{w}_i$  is as a summation of the columns  $\mathbf{e}_r \left( \frac{j-1}{L_r} \right)$  such that  $g_{i,j} = 1$ . In other words, we can design  $\mathbf{w}_i$  as:

$$\mathbf{w}_i = \sum_{j:g_{i,j}=1} \mathbf{e}_r \left( \frac{j-1}{L_r} \right) \quad (26)$$

*Remark 2:* The adopted beamforming design is ideal under the perfect sparsity assumption (which we adopt). That is when channel paths lie at the angular directions defined in  $\mathbf{U}_r$ . In practice, however, channel paths arrive at arbitrary angles in  $[0, 2\pi]$ . This makes each path contribute to multiple components in  $\mathbf{q}^a$ , hence,  $\mathbf{q}^a$  is not perfectly sparse. This happens due to (i) antenna side-lobes, and (ii) beam-overlap. To resolve this problem, we can use side-lobe suppression techniques, e.g. Taylor Window, as well as large antenna arrays, which can form pencil-beam antenna patterns that avoid the beam-overlap problem. These come at the expense of a slight

reduction in beam resolution. We leave this investigation for a future study and only focus on the main idea of using source-coding-based measurements.

#### D. On the Lower Bound on the Number of Measurements

In Theorem 1, we showed that a linear source code  $C$  which can **uniquely** encode all  $\mathbf{q}_s^a \in \mathcal{Q}_s^a$  can be used to design a framework that uniquely measures all  $\mathbf{q}^a \in \mathcal{Q}^a$ . Let the compression ratio of the code  $C$  be denoted by  $r_c$  such that  $r_c = \frac{m}{n_r}$ , where  $m$  and  $n_r$  are the number of rows and columns of  $C$ 's generator matrix  $\mathbf{G}$ , respectively.

Reducing the number of measurements is a fundamental objective for the mm-wave beam discovery problem. In light of Theorem 1, we can see that finding a source code with a high compression rate (small  $r_c$ ) is crucial for attaining such an objective. In the following discussion, we try to better understand the nature of this lower bound in the context of our proposed solution.

*Corollary 1: Let  $\underline{m}$  denote the lowest possible number of measurements for mm-wave beam discovery using lossless, fixed-rate source coding. Then, we have*

$$\underline{m} \geq \left\lceil \log_2 \left( \sum_{i=0}^L \binom{n_r}{i} \right) \right\rceil \geq H_2(\mathbf{q}_s^a) \quad (27)$$

where  $H_2(\cdot)$  is the binary entropy function.

*Proof:* Suppose that  $C$  is a linear lossless fixed-rate source code which can uniquely compress all  $\mathbf{q}_s^a \in \mathcal{Q}_s^a$ . By Theorem 1, we have that the number of measurements needed for estimating the mm-wave channel is equal to  $m$  (the length of encoded channel support vectors). Since the length of compressed sequences for any such code is lower bounded by  $H_2(\mathbf{q}_s^a)$ , then we have  $\underline{m} \geq H_2(\mathbf{q}_s^a)$ . Moreover, since fixed-rate source codes do not take the probability distribution (i.e., frequency) of  $\mathbf{q}_s^a$  into account, then we have  $\underline{m} \geq \sup_{\mathbb{P}(\mathbf{q}_s^a)} H_2(\mathbf{q}_s^a) \geq H_2(\mathbf{q}_s^a)$ , where

$$\sup_{\mathbb{P}(\mathbf{q}_s^a)} \sum_{\mathbf{q}_s^a \in \mathcal{Q}_s^a} \mathbb{P}(\mathbf{q}_s^a) \log_2 \left( \frac{1}{\mathbb{P}(\mathbf{q}_s^a)} \right) = \log_2(|\mathcal{Q}_s^a|) \quad (28)$$

The result of solving the sup problem in Eq. (28) is  $\mathbb{P}(\mathbf{q}_s^a) = \frac{1}{|\mathcal{Q}_s^a|} \forall \mathbf{q}_s^a$  since the uniform distribution maximizes the entropy. Since the number of measurement has to be an integer, we take the ceil of right hand side of Eq. (28). Finally, by the inclusion property in Definition 4, we have  $|\mathcal{Q}_s^a| = \sum_{i=0}^L \binom{n_r}{i}$ , which concludes the proof.  $\square$

#### E. Channel Estimation Error

In Theorem 1, we have shown how to obtain unique measurements  $\mathbf{y}^s \forall \mathbf{q}^a \in \mathcal{Q}^a$ . Recall that  $\mathbf{y}^s = \mathbf{G}\mathbf{q}^a$  is an error-free measurement vector. In practice, however, measurements are never error-free. Measurements errors are bound to happen due to the effects of **thermal and quantization noise**, among others factors. The quality of channel estimates obtained using error-corrupted measurements is essentially degraded, which calls for a deeper understanding of the effects of such errors. A crucial question we try to answer here is: **Do small perturbations/imperfections in channel measurements make channel**

**estimates considerably deviate from their true values?** In this section, we shed some light on this problem by deriving an upper bound on channel estimation error as a function of measurement error. We also show that for a special class of generator matrices, the channel estimation error, measured using the  $\ell_2$ -norm, is smaller than or equal to the  $\ell_2$  norm of the measurement error.

We denote error-corrupted measurements using  $\mathbf{u}^s$  such that

$$\mathbf{u}^s = \mathbf{y}^s + \mathbf{z}, \quad (29)$$

where  $\mathbf{z}$  is the measurement error (recall Eq. (13) and the discussion that follows). Assume that we can **perfectly** decode any measurement vector into its corresponding channel. That is, for any measurement vector  $\mathbf{y}^s$ , we can find a corresponding  $\mathbf{q}^a$  such that  $\mathbf{y}^s = \mathbf{G}\mathbf{q}^a$  (measurement decoding will be further discussed in Section VI). Let us also denote the *channel estimate* obtained using error-corrupted measurements  $\mathbf{u}^s$  by  $\hat{\mathbf{q}}^a$ , i.e.,  $\mathbf{u}^s = \mathbf{G}\hat{\mathbf{q}}^a$ . The following proposition provides an upper bound on the channel estimation error in terms of measurements errors.

*Proposition 1: Assume perfect measurement decoding, and let  $\sigma_{\min}(\cdot)$  denote the minimum singular value of a given matrix. Then, the channel estimation error is upper bounded as:*

$$\|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2 \leq \frac{1}{\sigma_{\min}(\mathbf{G})} \|\mathbf{z}\|_2 \quad (30)$$

*Proof:* Let us start by writing  $\mathbf{z}$  as:  $\mathbf{z} = \mathbf{u}^s - \mathbf{y}^s = \mathbf{G}(\hat{\mathbf{q}}^a - \mathbf{q}^a)$ . Therefore, we have

$$\implies \|\mathbf{z}\|_2 = \|\mathbf{G}(\hat{\mathbf{q}}^a - \mathbf{q}^a)\|_2 \quad (31)$$

$$= \|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2 \frac{\|\mathbf{G}(\hat{\mathbf{q}}^a - \mathbf{q}^a)\|_2}{\|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2} \quad (32)$$

$$\geq \|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2 \sigma_{\min}(\mathbf{G}) \quad (33)$$

Finally, by rearranging (33), we obtain the required statement

$$\|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2 \leq \frac{1}{\sigma_{\min}(\mathbf{G})} \|\mathbf{z}\|_2 \quad \square$$

Now, we see that if  $\sigma_{\min}(\mathbf{G}) \geq 1$ , then the channel estimation error (measured using the  $\ell_2$ -norm) is smaller than or equal to the  $\ell_2$ -norm of the measurement error, i.e.,  $\|\hat{\mathbf{q}}^a - \mathbf{q}^a\|_2 \leq \|\mathbf{z}\|_2$ . This, in fact, is the case for the class of generator matrices introduced in the following proposition

*Proposition 2: Let  $\mathbf{I}_m$  be the  $m \times m$  identity matrix. Then,  $\sigma_{\min}(\mathbf{G}) \geq 1$  for  $\mathbf{G}$  of the form:*

$$\mathbf{G} = (\mathbf{I}_m \mid \mathbf{P}_{m \times n-m}) \quad (34)$$

See Appendix C for proof.

*Remark 3: It is not difficult to obtain generator matrices of the form in Eq. (34). For instance, syndrome source codes can be manipulated using row and column operations over the binary field to produce equivalent codes with  $\mathbf{G}$  as in Eq. (34).*

## VI. MEASUREMENT DECODING

Designing channel measurements that have one-to-one correspondence with  $\mathbf{q}^a$  is only part of the solution. Equally important, however, is the ability to “decode”  $\mathbf{y}^s$  back to



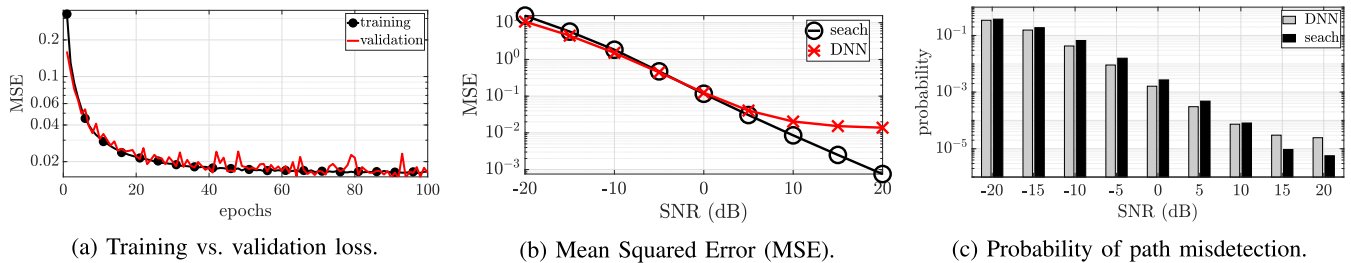


Fig. 4. Evaluation of DNN-based measurement to channel mapping.

$q^a$ , i.e., figuring out what the function  $\mathcal{D}(\cdot)$ , in Eq. (14c), is. The one-to-one correspondence between  $q^a$  and  $y^s$  guarantees that there exists an inverse function that maps  $y^s$  back to  $q^a$ . Nevertheless, since we can only obtain  $u^s$ ; an error-corrupted version of  $y^s$ , we cannot exactly regenerate  $q^a$ , but rather, an estimate  $\hat{q}^a$ . Given that measurement errors occur, our objective is to obtain  $\hat{q}^a$  such that its *distance* to  $q^a$  is as small as possible (i.e., minimize the estimation error). We use the  $\ell_2$  norm as a distance measure between  $q^a$  and  $\hat{q}^a$ , defined as  $\delta(q^a, \hat{q}^a) \triangleq \|q^a - \hat{q}^a\|_2$ .

Optimal measurement decoding requires solving an  $\ell_0$ -norm minimization problem [9]. This problem is not convex and its solution requires heavy computations, which is intractable for channels with large dimensions and/or relatively high sparsity level. An example for optimal measurement decoding is the “search” decoding, proposed in [18], which requires a combinatorial search over the column subspaces of  $G$ , and whose complexity is of order  $O(n_r^L)$ . Again, this is prohibitive for large antenna arrays ( $n_r$ ) and large  $L$ . Another solution is the “look-up” table method in [18], where quantized measurement-channel pairs are stored in memory. Here, the channel vector whose corresponding stored measurement is closest to the collected measurement is selected. The main disadvantage with this method is that the table size increases dramatically with the number of measurements and ADC quantization resolution. **Motivated by the drawbacks of the look-up table and search methods, we propose an alternative Machine Learning (ML) based approach that uses Deep Neural Networks (DNN).** DNNs in particular are commonly used as function approximation algorithms, hence they provide an appealing light-weight, data-driven, alternative solution for the measurement decoding problem. Our main goal here is to reduce the computational complexity while still maintaining reliable measurement decoding.

#### A. DNN-Based Mapping

ML is widely used to solve very complex problems through *learning*. We focus on *supervised learning* to solve the decoding problem, which is a multi-dimensional non-linear regression problem for which neural networks is a powerful tool. Specifically, we use a fully connected classical DNN with an input layer of  $m$  nodes (equal to the measurement dimensions) and an output layer of  $n_r$  nodes (equal to the channel dimensions). The DNN model is designed to handle real-valued input-output data. But, on the contrary, both the channel  $q^a$  and measurements  $y^s$  are complex-valued. To

overcome this problem, observe that  $y^s$  can be written as  $y_R^s + jy_I^s$  and  $q^a$  as  $q_R^a + jq_I^a$  (i.e., in terms of their real and imaginary components). And notice that  $y_R^s = Gq_R^a$  and  $y_I^s = Gq_I^a$ . Therefore, we can construct an estimate  $\hat{q}^a$  using its real and imaginary components, i.e.,  $\hat{q}_R^a + j\hat{q}_I^a$ , where  $\hat{q}_R^a$  and  $\hat{q}_I^a$  are estimated using  $y_R^s$  and  $y_I^s$  as inputs to the DNN model, respectively. Therefore, our DNN takes the measurement vectors  $y^s$  as inputs and produces the corresponding channel estimates  $\hat{q}^a$  at its output, but it does so in two different steps, handling the real and imaginary parts separately. For ease of notation, we will not use real and imaginary components to refer to inputs and outputs of the DNN models but it should be understood that this is how we handle it. The number of hidden layers and their corresponding number of nodes are design parameters that depend on the sizes of the input and output, and the relationship governing them. For all hidden layers, we use the rectified linear (ReLU) activation function while for the output layer we use the linear activation function. We also use the *ADAM* optimizer [32] for training and the Mean Squared Error (MSE) loss function to quantify the model error.<sup>3</sup>

**Model training:** Although we do not have a closed form expression for mapping  $y^s$  to  $q^a$  (hence the need for an algorithmic solution), generating training data is actually straightforward. This is because the reverse direction (i.e., mapping  $q^a$  to  $y^s$ ) is just a simple linear transformation. Training data is generated as follows: For every  $q_s^a \in \mathcal{Q}_s^a$  (recall that  $\mathcal{Q}_s^a$  is the set of all possible channel support vectors defined in Section V-B), we generate  $n_s = 300$  random channels,  $q^a$ , by choosing the non-zero components of  $q^a$  to be uniformly distributed in  $[-\alpha_{\max}^b, \alpha_{\max}^b]$  where  $\alpha_{\max}^b$  (recall Eq. (5)) is the maximum magnitude of baseband path gains, which can be obtained using channel statistics. Note that we can set  $\alpha_{\max}^b$  to be the maximum ADC quantized value of  $|\alpha_l^b|$ . Thus, the total number of input-output samples we have is  $n_s \times |\mathcal{Q}_s^a|$ . We use 70% of these samples for training and the remaining 30% for validation. Training is done using 200 epochs with batches of size 32. We monitor the validation error to make sure that the model does not over-fit the training data. If over-fitting is observed (which is indicated by a persistent increase in validation error at the end of every epoch), we stop the training process and only keep the model which produced the least

<sup>3</sup>We use Keras API [33] to build, train, test and use the DNN model we propose. Our pre-trained models and DNN-related codes are available in [34].

validation error. DNN training is done offline, and a trained DNN model is stored in memory to be used when needed.

### B. DNN Model Assessment

To argue the reliability of DNN-based mapping, we test it using a channel with  $n_t = 1$ ,  $n_r = 23$  and a maximum of 3 paths i.e.,  $L \leq 3$ . We compare its performance against the “search” method of [18]. Based on the described channel parameters, only  $m = 11$  measurements are needed to discover its paths (more details about this particular example are discussed in Section VIII). We design a DNN model with an input layer of  $m = 11$  nodes and an output layer of  $n_r = 23$  nodes. The model also has 5 hidden layers with 1024, 512, 512, 128 and 128 nodes, respectively. We train the DNN model using data generated as described in Section VI-A. Fig. 4a shows the average MSE loss of both training and validation data sets for 100 epochs. Training achieves validation error of  $\approx 0.0143$  (averaged across all samples of validation data). The figure also shows close MSE values for training and validation. This indicates that the model generalizes well to measurements it had not seen before, which guarantees reliability for arbitrary measurements.

These initial results are promising. However, they are obtained using error-free measurements. This prompts us to test the resilience of DNN-based mapping against noisy measurements. We also compare its performance against the search method proposed in [18]. To do so, we generate a testing data set in the same way we created the training data. We also generate sets of uniformly distributed noise vectors where each noise set is drawn at a different value of transmit SNR from  $-20$  to  $20$  dB. The noise vectors are then added to the inputs (channel measurements) of the testing data set then passed through the trained model. The decoded  $\hat{q}^a$  is recorded at the output. Similarly, we use the “search” method to decode the same noise-corrupted measurements.

For evaluation, we use i) average MSE, as well as ii) the probability of path misdetection (i.e., no path discovery). We say a path is correctly discovered if the path gain of its corresponding component in  $\hat{q}^a$  is among the  $L = 3$  strongest components in  $\hat{q}^a$ . Fig. 4b shows the average MSE obtained using the search and DNN-based mapping methods on a log scale. We see that at low SNR, DNN-based decoding outperforms the search method. This indicates that the DNN model is more resilient against measurement errors. At high SNR, however, the DNN’s MSE saturates at  $\approx 0.014$  which is the same value we obtain for validation during model training using noise-free inputs (note that the MSE value at which DNN-based mapping saturates can be made lower by further improvement of the DNN model). The search method’s MSE, on the other hand, keeps improving as SNR increases, nevertheless, for values below  $10^{-2}$  the improvement is marginal. The probability of path misdetection, shown in Fig. 4c, confirms the performance trend of the MSE. Specifically, we see that at low SNR, the DNN-based model outperforms the search method (i.e., has lower probability of misdetection) while at high SNR we see that the search method is better.

**Computational complexity:** As we have previously discussed, the search method requires high computational power.

Precisely,  $\binom{n_r}{L}$  iterations with one matrix inversion and two matrix multiplication operations are performed per iteration, which then produces a vector of length  $m$ . Finally, an additional step of finding the minimum  $\ell_2$ -norm of all  $\binom{n_r}{L}$  vectors is performed. On the other hand, the DNN-based mapping just requires  $N_k$  linear computations for the hidden and output layers, where  $N_k$  is the number of nodes at the  $k^{\text{th}}$  layer. These computations are of the form  $\sum_{i=1}^{N_{k-1}} w_i a_i$  where  $a_i$  is the value passed from the  $i^{\text{th}}$  node of the previous layer and  $w_i$  is the weight on its link. For this particular example, the search method and the DNN-based method were implemented on the same machine and on average the search method’s execution time was 11.2 ms compared to 47  $\mu$ s for the DNN model.

## VII. MULTIPLE TRANSMIT AND RECEIVE ANTENNAS

So far, we only dealt with channels of single-transmit, multiple-receive antennas. Recall that this setting is almost identical to multiple-transmit, single-receive antenna channels, except that in the former setting we seek to design  $w_i$ ’s to estimate the angular channel at RX, while in the latter, we design  $f_j$ ’s to estimate the angular channel at TX. In this section, we extend the channel setting to be of **multiple-transmit, multiple-receive** antennas. We build on the design principles and decoding methods of single transmit antenna channels and show how measurements are obtained and decoded to estimate the entire  $n_r \times n_t$  channel.

### A. Measurements

Unlike the single transmit antenna scenario where TX sends signals omnidirectionally, it can now focus its transmission on narrow angular directions. However, from RX’s point of view, no matter which set of directions the TX is transmitting into, it can only see a number of  $n_r$  resolvable bins; only  $L$  of which may have paths to TX. The same is true from TX’s perspective, where the TX can only see  $n_t$  resolvable bins, only  $L$  of which may have paths to the receiver.<sup>4</sup> Thus, for an arbitrary tx-precoder, the receiver would need to measure the channel using the same set of  $w_i$ ’s it needs for the  $n_t = 1$  setting. Upon decoding, the result would be  $n_r$  angular rx bins (corresponding to the particular  $f_j$  used at TX). Similarly, for an arbitrary rx-combiner, the transmitter would need the same set of  $f_j$ ’s it needs for the  $n_r = 1$  setting to find its respective tx bins. To find such  $f_j$ ’s and  $w_i$ ’s, we invoke Theorem 1.

Let  $f_j \forall j \in \{1, \dots, m_t\}$  be the tx-precoding vectors and  $w_i \forall i \in \{1, \dots, m_r\}$  be the rx-combining vectors. Then, the channel measurements are obtained as follows: The transmitter sends a number of  $m_r$  pilot symbols using **each** of its  $m_t$  precoders. On the receiver side, for every tx-precoder,  $m_r$  channel measurements are obtained using the distinct  $m_r$  rx-combiners. Recall that  $u_{i,j} = y_{i,j}^s + w_i^H \mathbf{n}$  where  $y_{i,j}^s = w_i^H \mathbf{Q} f_j$  (see Eq. (12)). Let us arrange the  $m_r$

<sup>4</sup>Recall that the directions at which the TX is transmitting and the RX is receiving are determined by their antenna beam patterns which are in turn determined by  $f_j$  and  $w_i$ , respectively (see Fig. 2).

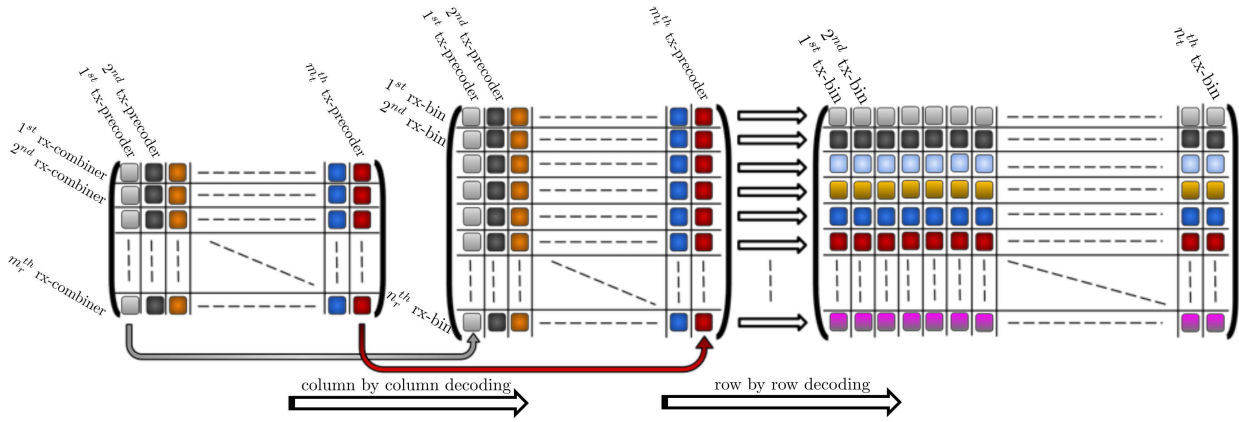


Fig. 5. Measurement decoding for channels with multiple TX/RX antennas is done in two steps. Given the matrix  $\mathbf{Y}^s$  whose  $y_{i,j}^s$  component is  $= \mathbf{w}_i^H \mathbf{Q} \mathbf{f}_j$  (shown on the left), we first do a column by column decoding where the  $j^{\text{th}}$  decoded column of  $\mathbf{Y}^s$  is  $\mathbf{q}_{rx,j}^a$ . Then, in the second step we decode the intermediary matrix row by row to produce  $\hat{\mathbf{Q}}^a$  (shown on the right). The two decoding functions of first and second steps are dependent on the source codes used to design  $\mathbf{w}_i$ 's and  $\mathbf{f}_j$ 's, respectively.

measurements corresponding to the  $j^{\text{th}}$  tx-precoder in  $\mathbf{y}_j^s$  and define  $\mathbf{Y}^s$  as

$$\mathbf{Y}^s \triangleq (\mathbf{y}_1^s \mathbf{y}_2^s \cdots \mathbf{y}_{m_t}^s) \quad (35)$$

Thus,  $\mathbf{Y}^s$  contains all  $m_t \times m_r$  channel measurements necessary to discover all available paths.

### B. Decoding $\mathbf{Y}^s$

To obtain  $\hat{\mathbf{Q}}^a$  from  $\mathbf{Y}^s$ , we perform multiple SIMO decoding operations,<sup>5</sup> as described in Section VI. This procedure is highlighted in the diagram in Fig. 5 and is detailed as follows:

- 1) Decode every  $\mathbf{y}_j^s \forall j \{1, 2, \dots, m_t\}$  to obtain  $\mathbf{q}_{rx,j}^a$ . Recall that  $\mathbf{y}_j^s$  is the measurement vector corresponding to the  $j^{\text{th}}$  tx-precoder. Thus,  $\mathbf{q}_{rx,j}^a$  is the  $n_r \times 1$  mm-wave channel observed at RX due to the TX signal transmission through the angular directions featured by  $\mathbf{f}_j$ .
- 2) After Step (i), we obtain a sequence of  $m_t$  “measurement” components corresponding to each rx-bin. Each of these components is produced using a distinct tx-precoder. Let us denote these sequences by  $\mathbf{y}_{tx,k}^s$  ( $1 \times m_t$  row vectors), where  $k \in \{1, 2, \dots, n_r\}$ .
- 3) Decode each  $\mathbf{y}_{tx,k}^s$  to obtain  $\mathbf{q}_{tx,k}^a$  ( $1 \times n_t$  row vectors) whose components constitute all the tx-bins corresponding to the  $k^{\text{th}}$  rx-bin.
- 4) Stack all  $\mathbf{q}_{tx,k}^a$  to obtain  $\hat{\mathbf{Q}}^a$  (each representing the  $k^{\text{th}}$  row of  $\hat{\mathbf{Q}}^a$ ).

<sup>5</sup>Alternatively, we could have trained a large DNN model which accepts all measurements  $\mathbf{Y}^s$  and outputs an estimate  $\hat{\mathbf{Q}}^a$ . Adopting this strategy, however, results in overwhelming training complexity since this model would need to be trained with a massive training data set of size  $n_s \times |\mathcal{Q}_s^a| = n_s \sum_{i=0}^L \binom{n_r \times n_t}{i}$ , where  $\mathcal{Q}_s^a$  now is the set of all support vectors of size  $n_r \times n_t \times 1$  that represent the  $n_r \times n_t$  vectorized channel matrices. Compare this to our solution of using 2 DNN models trained with data sets of sizes  $n_s \sum_{i=0}^L \binom{n_r}{i}$  and  $n_s \sum_{i=0}^L \binom{n_t}{i}$ , respectively.

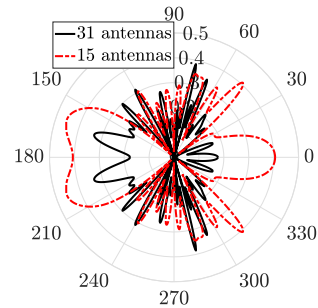


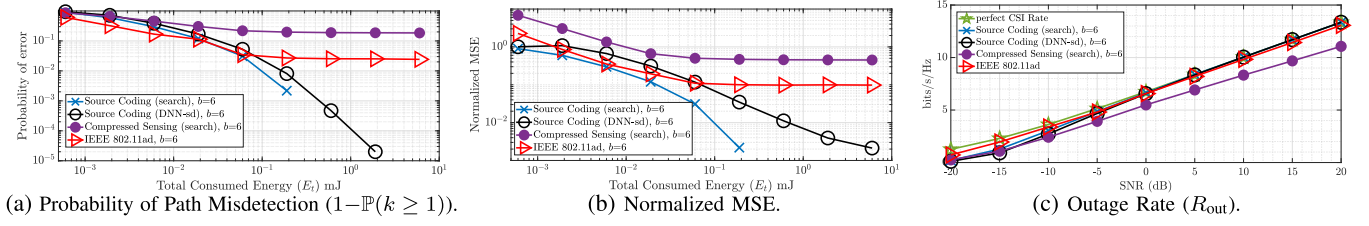
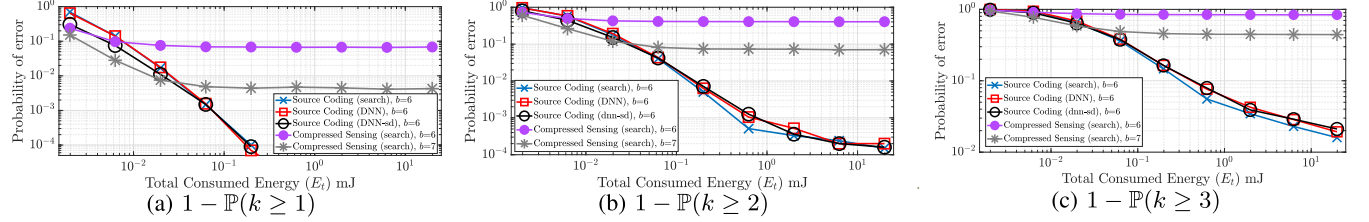
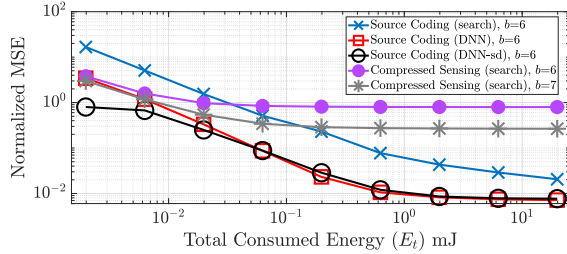
Fig. 6. Antenna pattern example using CS.

## VIII. PERFORMANCE EVALUATION

We evaluate the performance of our proposed coding-based solution under various simulation scenarios. Specifically, we consider  $23 \times 23$  and  $15 \times 32$  multi-path channels. For both channel settings, we assume the existence of a maximum of 3 paths.<sup>6</sup> We also consider  $15 \times 31$  single-path channels. The single-path assumption is appropriate for LoS scenarios where the path gain of the LoS is significantly higher than the gains of Non-LoS (NLoS) paths ( $\approx 20$  dB higher [3]).

To understand how our solution compares to the state-of-the-art, we implement a compressed-sensing-based channel estimation solution, as well as the IEEE 802.11ad's (WiGig) beam discovery method. Note that, while compressed sensing is a generic solution that can be applied to multi-path channels (similar to our solution), the WiGig method is designed to discover one channel path, hence, we only use it for the  $15 \times 31$  single-path channel. Our results demonstrate that our proposed solution is more resilient to errors compared to both CS and 802.11ad, and produces higher quality estimates. Furthermore, we study the effect of ADC resolution on channel estimation performance. This is important because ADC's power consumption is directly proportional to their resolution. Hence, it is necessary to understand the resolution limit beyond

<sup>6</sup>Note that information on  $L$  can be obtained from statistical channel models [3], [35].

Fig. 7. Performance of single-path  $15 \times 31$  channels.Fig. 8. Beam detection probability of error for  $23 \times 23$  channels with  $L \leq 3$ .Fig. 9. Normalized MSE ( $23 \times 23$  channels with  $L \leq 3$ ).

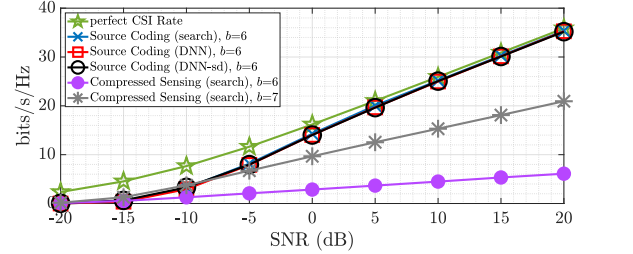
which only minimal gains, in terms of channel estimation performance, can be achieved.

### A. Performance Metrics

We adopt performance metrics that highlight: (i) the measurement overhead, (ii) accuracy of path discovery (iii) quality of the estimated path gains, and (iv) the impact of the channel estimates on achievable data rate. These metrics are evaluated numerically, using Monte Carlo simulations (averaged over  $10^5$  simulation runs) and evaluated against different values of SNR. We define the performance metrics as follows: *i) Number of measurements*: Given by  $m_t \times m_r$ . *ii) Probability of path discovery*: Paths are said to exist at the strongest  $L$  components in the estimated channel  $\hat{Q}^a$ . *iii) MSE* (Normalized): Defined as the squared value of the Frobenius-norm of the estimation error,  $Q^a - \hat{Q}^a$ , normalized by the Frobenius-norm of  $Q^a$ , i.e.,  $\frac{\|Q^a - \hat{Q}^a\|_F^2}{\|Q^a\|_F^2}$ . *iv) Outage Rate*: Denoted by  $R_{out}$  and defined as  $R_{out} \triangleq \mathbb{E}[(1 - \mathbb{1}_{\{out\}}) \times C_Q]$  where  $C_Q$  is the MIMO channel capacity of the channel  $Q$  [29], and  $\mathbb{1}_{\{out\}}$  is the indicator function that takes a value of 1 in case of outage and 0 otherwise. We assume that an outage occurs if any of the strong channel paths were misdected.

### B. Implemented Solutions

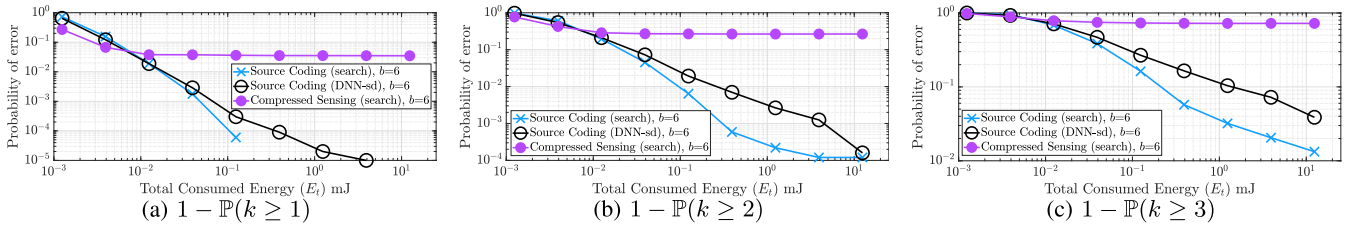
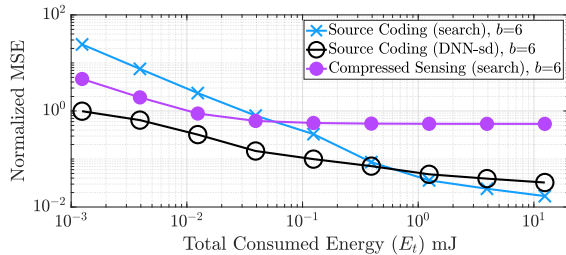
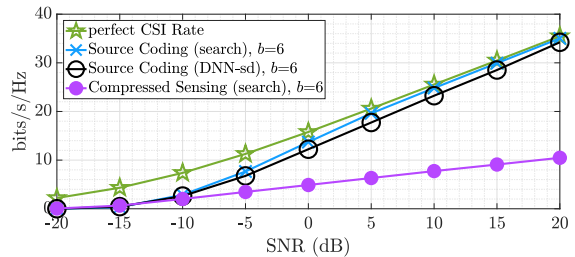
**1- Source Coding**: We test three different measurement decoding methods, which we integrate with our coding-based

Fig. 10. Outage Rate ( $R_{out}$ ,  $23 \times 23$  channels with  $L \leq 3$ ).

solution. All three methods are used to solve each of the sub-problems depicted in Fig. (5). The first is the “search” method of [18]. The second and third decoding methods are based on DNNs, but they differ in the way they are trained, i.e., with or without measurement errors. We explain them as follows:

- **DNN**: Here, DNN models are trained using pure measurements, with no added noise components. Since models are not trained with errors, only one model can be used at all SNR and ADC resolution levels.
- **DNN-sd**: Since measurement errors tend to degrade the performance of path discovery, we try to overcome this impediment by training DNN models with error-corrupted measurements. Since errors are dependent on ADC resolution and SNR (see Eq. (11)), we train multiple DNNs for different values of each. We call such model “DNN with selective defense” or “DNN-sd”. Note that the DNN-sd model is not dependent on specific path gain values since our SNR definition does not include the effect of individual path gains  $\alpha_l$  and because training is done using a wide range of uniformly distributed gain values.

The DNN model parameters, including the number of layers, the number of nodes (neurons) per layer and the activation function, are selected using cross-validation. We also select the DNN model’s size such that we have a reasonably good input-output mapping performance while keeping the processing speed fairly fast. We used tensorflow [36] for creating and

Fig. 11. Beam detection probability of error for  $15 \times 32$  channels with  $L \leq 3$ .Fig. 12. Normalized MSE ( $15 \times 32$  channels with  $L \leq 3$ ).Fig. 13. Outage Rate ( $R_{\text{out}}$ ,  $15 \times 32$  channels with  $L \leq 3$ ).

using DNN models. **Both types of DNN models are trained offline and stored in memory.**

**2- Compressed Sensing:** We use a similar formulation for the mmWave channel estimation problem as in [21], [22]. The tx-precoders and rx-combiners are obtained using random, uniformly distributed phase shift values. That is, the components of all  $f_j$ 's and  $w_i$ 's are of the form  $\exp(j\vartheta)$  where  $\vartheta \sim [0, 2\pi)$ . Fig. 6 shows antenna pattern examples for random beamforming with 15 and 31 antennas. For measurement decoding, we use the “search” method, which is the optimal  $\ell_0$ -norm minimization solution [9] for solving each of the sub-problems of channel decoding. While this may still be too computationally complex to be of practical use, it provides an upper bound on the performance of other sparse recovery algorithms like OMP,  $\ell_1$  and  $\ell_2$ -norm minimization, etc.

**3- 802.11ad:** We only consider the Sector Level Sweep stage of the channel estimation scheme of 802.11ad. At this stage, the TX starts by sequentially transmitting packets in all possible transmit AoDs (sectors) while the receiver performs quasi omni-directional reception. Then, the TX and RX switch modes where TX forms a quasi omni-directional pattern while RX sweeps through all possible receive AoAs (sectors). The directions that reveal the highest received signal strength is denoted as the AoA and AoD of the strongest channel path.

### C. Equating Energy Consumption

Various channel estimation solutions may require different number of measurements and may have different beamforming gains. Thus, it would not be fair to compare them at fixed transmission power. Instead, it is more fair to fix the total energy consumption for the whole channel estimation process of each solution. Thus, for comparison purposes, we opt to adjust the transmit power of each scheme such that **the total amount of energy consumption for the entire measurement process remains the same.**

**Energy Calculation:** The energy consumption, denoted by  $E_T$ , is given by  $E_T = m \times P_T \times \tau$ , where  $m$  is the number of measurements,  $P_T$  is the total transmit power per measurement and  $\tau$  is the time duration of one measurement. Since the antenna patterns at TX/RX of our proposed scheme consist of multiple overlapped beams (recall Fig. 2), the total power  $P_T$  is an integer multiple of the transmit power per direction/beam  $P$ , which depends on the number of overlapped beams at TX and RX. Let  $o_t$  and  $o_r$  denote the number of overlapped beams at the TX and RX, respectively. Hence, we have that  $P_T = o_t \times o_r \times P$ . We can further write the transmit power per beam as  $P = \text{SNR} \frac{N_0}{\mu}$  (recall Eq. (11)). This gives us a total energy consumption (in millijoules (mJ)) for the entire measurement process as:  $E_T = m \times o_t \times o_r \times \text{SNR} \frac{N_0}{\mu} \times \tau$ . Let  $\mu = -88\text{dB}$  and  $N_0 = 88\text{dBm}$ .<sup>7</sup> Finally, let  $\tau \approx 23\mu\text{s}$  (from IEEE 802.11ad).

### D. Results

**15x31 single-path channels:** For this scenario, we choose ADCs of resolution  $b = 6$  bits. We provide results for our coding-based solution with both search and DNN-sd decoding. We also provide results for compressed sensing with search-based decoding, and IEEE 802.11ad beam alignment. We plot the results against the consumed energy  $E_T$ . **Source code selection:** We choose codes which satisfy the requirements in Theorem 1 as follows: At the TX side, we choose the (31, 26) Hamming code to design tx-precoders, while at the

<sup>7</sup>To find  $\mu$  and  $N_0$ , we assume a channel operating at a carrier frequency  $f_c = 60$  GHz with bandwidth  $B = 100$  MHz and distance between TX/RX of  $d = 10\text{m}$ . Further, we assume a receiver system with noise figure  $\text{NF} = 6$  dB and temperature  $T_0 = 293^\circ$  kelvin. The path loss constitutes both the free space path loss (FSPL) and atmospheric absorption. FSPL is given as:  $\text{FSPL} = -10 \log_{10} \left( \frac{4\pi}{c} df_c \right)^{n_p}$ , where  $n_p = 2$  is the path loss exponent. Atmospheric absorption, however, can be ignored for small distances ( $\leq 50\text{m}$ ) [37]. Hence,  $\mu = \text{FSPL} = -88\text{dB}$ . The noise power (in dBm) can be given as  $N_0 = 10 \log_{10} (k_B T_0 B \times 1000) + \text{NF}$ , where  $k_B$  is the Boltzmann constant.

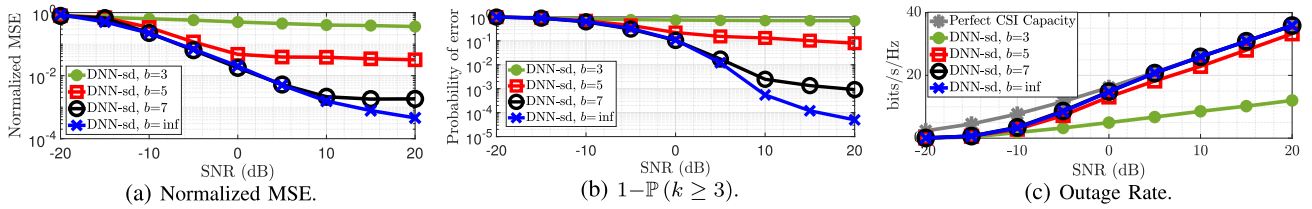


Fig. 14. Performance of various ADC quantizations.

RX side we choose the (15, 11) Hamming code to design rx-combiners. Both of which operate as syndrome source codes with generator matrices  $\mathbf{G}_t$  and  $\mathbf{G}_r$  of sizes  $5 \times 31$  and  $4 \times 15$ , respectively. Hence, we have  $m_t = 5$  and  $m_r = 4$ , which gives us a **total number of 20 measurements**. For compressed sensing, we use the same number of measurements (i.e.,  $m = 20$ ). An exhaustive search, on the other hand, requires 465 measurements (our solution represents a measurement 95% reduction), while the IEEE 802.11ad scheme requires 46 measurements (57% reduction).

For the source coding method, both the normalized MSE and probability of path misdetection results, shown in Fig. 11, depict that DNN-sd decoding has a slightly worse performance compared to the search method. This is a small sacrifice in performance that is traded for a huge advantage in processing speed. The IEEE 802.11ad's method shows superior performance at low  $E_T$  (below 0.1 mJ). As  $E_T$  increases, however, its performance ceases to improve, while our source coding solution keeps approaching perfect channel discovery. When examining the outage rate, in Fig. 11c, we see that the relatively high MSE error and probability of path misdetection of 802.11ad, does not result in a significant degradation in  $R_{\text{out}}$ . In fact, it has very close value to the perfect CSI capacity. Recall that 802.11ad requires almost twice the number of channel measurements. The compressed sensing method, on the other hand, has the lowest  $R_{\text{out}}$  and the highest MSE and probability of path misdetection among all other schemes.

**23 × 23 multi-path channels:** This is a more challenging multi-path scenario where, in addition to the previous solutions, we also investigate the performance of DNN decoding for which training is done with pure uncorrupted measurements. **Source code selection:** For this channel, since  $n_r = n_t$ , the same source code works for designing both tx-precoders and rx-combiners. The *perfect binary Golay code* used as a syndrome source code is a suitable choice for this problem. It has a generator matrix of size  $11 \times 22$ , hence, we have  $m_t = m_r = 11$ , and the **total number of required channel measurements is  $m_t \times m_r = 121$** . Compared to the exhaustive search approach, which requires scanning all 529 combinations of TX/RX angular directions, this represents **75% measurement reduction**. For compressed sensing, we use the same number of precoders and combiners, as well.

First, for the probability of path detection, shown in Fig. 11, we notice very close performance for all three measurement decoding methods (search, DNN and DNN-sd) when integrated with our source coding solution. This suggests that DNNs are very efficient. And, while DNN-sd has a slight edge

over DNN, the improvement is not significant. Hence, it is possible to use fewer DNNs trained over larger ranges of error components. Interestingly, however, in Fig. 10, we observe that the search decoding of our source-coding-based measurements tend to have significantly larger MSE. This indicates that DNNs tend to suppress the error components in the estimated channel values, even though the correct channel paths may not be efficiently discovered. This is an artifact of DNN training which suggests that we may be able to improve the DNN's decoding performance if they were directly trained to discover the channel paths rather than just decrease the MSE of the channel estimates.

Compressed sensing, however has significantly worse performance in terms of path discover, MSE and outage Rate. At very low  $E_T$ , CS's performance improves as  $E_T$  gets higher, but at  $E_T \geq 0.7$  the improvement stops. This suggests that the CS-based solution is more sensitive to quantification error. This is verified by using a higher resolution ADC ( $b = 7$  bits) which shows a significant improvement of performance.

**15 × 32 multi-path channel:** The performance results under this scenario is very similar to the  $23 \times 23$  channel setting shown above. Specifically, Fig. 11 shows the probability of path discovery where our coding method with search decoding shows superior performance compared to DNN-sd decoding. On the contrary, search has worse MSE compared to DNN-sd (as shown in Fig.12). Compressed sensing, has close performance to our proposed solution at low  $E_T$ . At high  $E_T$ , however, its performance only sees marginal improvement, unlike our proposed solution which keeps approaching perfect channel discovery. **Code selection:** We choose codes whose  $m_r = 11$  and  $m_t = 16$ . The **total number of required channel measurements is 176**, which constitutes 63.3% measurement reduction compared to exhaustive search.

### E. Effect of ADC Resolution on Performance

Now, we provide and compare results for  $23 \times 23$  channels with ADCs of  $b = 3, 5$ , and 7-bit resolution, as well as the ideal  $b = \infty$ . We only show results for DNN-sd decoding since the performance of the other two methods compare similarly to the trends of the ideal ADC case shown above. In Fig. 14a, we plot the MSE. As expected, we see that MSE is inversely proportional to resolution. We can also see that as SNR increases, higher resolution is required to keep the MSE close to that of ideal ADCs. For instance,  $b = 5$  is reasonably good up to  $\text{SNR} = -5\text{dB}$ , while  $b = 7$  is very close to  $b = \infty$  up to  $\text{SNR} = 5\text{dB}$ . Even at high SNR, the  $b = 7$

curve has a gap with ideal ADCs that is smaller than  $5 \times 10^{-3}$ . Similar performance trend is exhibited for the probability of path discovery, shown in Fig. 14b. Finally, the outage rate is depicted in Fig. 14c. We see that  $b = 7$  results in  $R_{\text{out}}$  that is almost identical to that of the ideal ADC, and that both of which are very close to the perfect CSI capacity. We also see that at low SNR, there is a considerable gap between the perfect CSI capacity and outage rate even for ideal ADCs.

## IX. CONCLUSION

In this work, we treat the mmWave channel estimation problem as a source compression problem. Our goal is to reconstruct the channel matrix using a small number of measurements. We exploit linear binary source codes for encoding the channel (do measurements) and a deep neural network based algorithm for measurement decoding (channel reconstruction). We are able, using a small number of measurements, to obtain high quality channel estimates. The lower bound on the achievable number of measurements is accurately characterized. Through simulation, the superiority of our proposed solution is demonstrated, in comparison to compressed-sensing-based solutions and IEEE 802.11ad's beam alignment.

## APPENDIX A PROOF OF LEMMA 1

*Proof:* Consider a set of  $n$ -dimensional linearly independent vectors,  $\mathbf{v}_1, \dots, \mathbf{v}_m$  defined over  $\mathbb{F}_2$ . Then, construct a matrix  $\mathbf{M}_{\mathbb{F}_2}$  whose columns are  $\mathbf{v}_1, \dots, \mathbf{v}_m$ . Since  $\mathbf{v}_i$ 's are independent, then  $\mathbf{M}_{\mathbb{F}_2}$  has full column rank, i.e.,  $\mathbf{M}_{\mathbb{F}_2}$  is left-invertible over  $\mathbb{F}_2$  ( $m \leq n$ ). Thus  $\mathbf{M}_{\mathbb{F}_2}$  has an  $m \times m$  minor, call it  $\mathbf{A}_{\mathbb{F}_2}$  whose determinant is non-zero. Now consider the matrix  $\mathbf{M}$ , defined over  $\mathbb{C}$ , whose elements are the 0 and 1 real scalars corresponding to  $0_{\mathbb{F}_2}$  and  $1_{\mathbb{F}_2}$  values of  $\mathbf{M}_{\mathbb{F}_2}$ . Let  $\mathbf{A}$  be its minor corresponding to  $\mathbf{A}_{\mathbb{F}_2}$  of  $\mathbf{M}_{\mathbb{F}_2}$ . By lemma 2 (in Appendix B), we have  $\det(\mathbf{A}) \neq 0$ . Thus,  $\mathbf{M}$  is also left-invertible, hence its columns are linearly independent.  $\square$

## APPENDIX B LEMMA 2

*Lemma 2:* Let  $\mathbf{A}_{\mathbb{F}}$  and  $\mathbf{A}$  be  $n \times n$  matrices defined over  $\mathbb{F}_2$  and  $\mathbb{R}$ , respectively. Let  $a_{\mathbb{F},i,j}$ , the elements of  $\mathbf{A}_{\mathbb{F}}$ , be scalars in  $\{0_{\mathbb{F}_2}, 1_{\mathbb{F}_2}\}$ , while  $a_{i,j}$  the elements of  $\mathbf{A}$ , be scalars in  $\{0, 1\} \subseteq \mathbb{R}$ . Suppose that  $\mathbf{A}_{\mathbb{F}}$  has non-zero determinant, i.e.,  $\det(\mathbf{A}_{\mathbb{F}}) \neq 0_{\mathbb{F}_2}$ . If we define  $\mathbf{A}$  such that  $a_{i,j} = 0$  if  $a_{\mathbb{F},i,j} = 0_{\mathbb{F}_2}$ , and  $a_{i,j} = 1$  if  $a_{\mathbb{F},i,j} = 1_{\mathbb{F}_2}$  for all  $1 \leq i, j \leq n$ . Then,  $\det(\mathbf{A}) \neq 0$ .

*Proof:* Recall that the determinant of a square matrix defined over a commutative ring is given by the Leibniz formula [38]. Since  $\mathbb{F}_2$  is a finite field (with 2 elements), it constitutes a commutative ring. Moreover,  $\mathbb{R}$  is a commutative ring [38]. Therefore, both determinants of  $\mathbf{A}_{\mathbb{F}}$  and  $\mathbf{A}$  can be computed using the same exact formula. Since, finite field arithmetic over the prime field  $\mathbb{Z}_2$  is the integers modulo 2, then we can write  $\det(\mathbf{A}_{\mathbb{F}}) = \det(\mathbf{A}) \pmod{2}$ . Thus,  $\exists q \in \mathbb{Z}$  (the set of integers), such that  $\det(\mathbf{A}) = q \times 2 + \det(\mathbf{A}_{\mathbb{F}}) = q \times 2 + 1$ , were the latter equation follows from the fact that  $\det(\mathbf{A}_{\mathbb{F}}) \neq$

$0_{\mathbb{F}_2} \iff \det(\mathbf{A}_{\mathbb{F}}) = 1_{\mathbb{F}_2}$ . Therefore,  $\det(\mathbf{A})$  is an odd integer, which implies that  $\det(\mathbf{A}) \neq 0$ , concluding our proof.  $\square$

## APPENDIX C PROOF OF PROPOSITION 2

*Proof:* We will prove that adding an extra column  $\mathbf{p} \in \mathbb{R}^m$  to any **full rank** matrix  $\mathbf{M}$  of size  $m \times k$  with  $m \leq k$  (i.e.,  $\text{rank}(\mathbf{M}) = m$ ) **does not reduce** its singular values.

Let  $\mathbf{M}_{\mathbf{p}} = (\mathbf{M} \mid \mathbf{p})$  be an  $m \times k+1$  matrix. Then, we can obtain the singular values of  $\mathbf{M}_{\mathbf{p}}$  as the positive square roots of the eigenvalues of  $\mathbf{M}_{\mathbf{p}}\mathbf{M}_{\mathbf{p}}^T$ , which can be written as:

$$\mathbf{M}_{\mathbf{p}}\mathbf{M}_{\mathbf{p}}^T = (\mathbf{M} \mid \mathbf{p})(\mathbf{M} \mid \mathbf{p})^T \quad (36)$$

$$= \mathbf{M}\mathbf{M}^T + \mathbf{p}\mathbf{p}^T \quad (37)$$

Since  $\mathbf{p}\mathbf{p}^T \succeq 0$  (i.e., positive semidefinite), then we must have  $\mathbf{M}_{\mathbf{p}}\mathbf{M}_{\mathbf{p}}^T - \mathbf{M}\mathbf{M}^T \succeq 0$ . Let  $\sigma_i(\cdot)$  denote the  $i^{\text{th}}$  largest singular value of a matrix. Then, we have  $\sigma_i(\mathbf{M}_{\mathbf{p}}\mathbf{M}_{\mathbf{p}}^T) \geq \sigma_i(\mathbf{M}\mathbf{M}^T) \forall i = 1, \dots, m$ , which implies

$$\implies \sigma_{\min}(\mathbf{M}_{\mathbf{p}}\mathbf{M}_{\mathbf{p}}^T) \geq \sigma_{\min}(\mathbf{M}\mathbf{M}^T) \quad (38)$$

$$\implies \sigma_{\min}(\mathbf{M}_{\mathbf{p}}) \geq \sigma_{\min}(\mathbf{M}) \quad (39)$$

Define  $\mathbf{G}^{(i)} \triangleq (\mathbf{g}_1 \mid \mathbf{g}_2 \mid \dots \mid \mathbf{g}_i)$ , where  $\mathbf{g}_j$  is the  $j^{\text{th}}$  column of  $\mathbf{G}$ . Then, by sequentially applying the result shown in Eq. (39) (by adding columns of  $\mathbf{P}$  in Eq. (34)), we obtain

$$\begin{aligned} \sigma_{\min}(\mathbf{G}) &= \sigma_{\min}(\mathbf{G}^{(n)}) \geq \sigma_{\min}(\mathbf{G}^{(n-1)}) \geq \dots \\ &\geq \sigma_{\min}(\mathbf{G}^{(m+1)}) \geq \sigma_{\min}(\mathbf{G}^{(m)}) = \sigma_{\min}(\mathbf{I}_m) = 1 \end{aligned} \quad \square$$

## 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] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.
- [3] M. R. Akdeniz *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014.
- [4] (Mar. 2017). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. [Online]. Available: <https://goo.gl/U1eQNM>
- [5] G. R. MacCartney, J. Zhang, S. Nie, and T. S. Rappaport, "Path loss models for 5G millimeter wave propagation channels in urban microcells," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 3948–3953.
- [6] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.
- [7] K. Haneda *et al.*, "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–7.
- [8] S.-E. Chiu, N. Ronquillo, and T. Javidi, "Active learning and CSI acquisition for mmWave initial alignment," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2474–2489, Nov. 2019.
- [9] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1527–1550, 3rd Quart., 2017.
- [10] K. Venugopal, A. Alkhateeb, N. G. Prelcic, and R. W. Heath, "Channel estimation for hybrid architecture-based wideband millimeter wave systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1996–2009, Sep. 2017.

- [11] 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.
- [12] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R. W. Heath, "Frequency-domain compressive channel estimation for frequency-selective hybrid millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 2946–2960, May 2018.
- [13] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, and P. Indyk, "Fast millimeter wave beam alignment," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2018, pp. 432–445.
- [14] X. Li and A. Alkhateeb, "Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems," 2019, *arXiv:1905.13212*. [Online]. Available: <http://arxiv.org/abs/1905.13212>
- [15] H. He, C.-K. Wen, S. Jin, and G. Ye Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [16] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.
- [17] 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, 2018.
- [18] Y. Shabara, C. E. Koksal, and E. Ekici, "Beam discovery using linear block codes for millimeter wave communication networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1446–1459, Aug. 2019.
- [19] J. Li, Y. Sun, L. Xiao, S. Zhou, and C. E. Koksal, "Analog beam tracking in linear antenna arrays: Convergence, optimality, and performance," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 1193–1198.
- [20] M. Hashemi, A. Sabharwal, C. Emre Koksal, and N. B. Shroff, "Efficient beam alignment in millimeter wave systems using contextual bandits," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 2393–2401.
- [21] A. Alkhateeb, G. Leus, and R. W. Heath, "Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?" in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2909–2913.
- [22] X. Lu, W. Yang, Y. Cai, and X. Guan, "Comparison of CS-based channel estimation for millimeter wave massive MIMO systems," *Appl. Sci.*, vol. 9, no. 20, p. 4346, Oct. 2019.
- [23] C.-K. Wen, C.-J. Wang, S. Jin, K.-K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs," *IEEE Trans. Signal Process.*, vol. 64, no. 10, pp. 2541–2556, May 2016.
- [24] J. Mo, P. Schniter, and R. W. Heath, "Channel estimation in broadband millimeter wave MIMO systems with few-bit ADCs," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1141–1154, Mar. 2018.
- [25] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Deep learning for compressed sensing based channel estimation in millimeter wave massive MIMO," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6.
- [26] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Nov. 2017.
- [27] *IEEE Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band*, Standard 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as Amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012, Dec. 2012).
- [28] W.-T. Li, Y.-C. Chiang, J.-H. Tsai, H.-Y. Yang, J.-H. Cheng, and T.-W. Huang, "60-GHz 5-bit phase shifter with integrated VGA phase-error compensation," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 3, pp. 1224–1235, Mar. 2013.
- [29] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [30] J. Mo, P. Schniter, N. G. Prelcic, and R. W. Heath, "Channel estimation in millimeter wave MIMO systems with one-bit quantization," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 957–961.
- [31] M. Kokshoorn, H. Chen, P. Wang, Y. Li, and B. Vucetic, "Millimeter wave MIMO channel estimation using overlapped beam patterns and rate adaptation," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 601–616, Feb. 2017.
- [32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [33] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: <https://keras.io>
- [34] Y. Shabara. (2021). *Simulation Codes for Source Coding Based Beam Discovery*. [Online]. Available: <https://github.com/yahiaShabara/beamDiscoveryPublic>
- [35] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling robust 60 GHz links under blockage," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2016, pp. 193–206.
- [36] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
- [37] A. Tomkins *et al.*, "A 60 GHz, 802.11ad/WiGig-compliant transceiver for infrastructure and mobile applications in 130 nm SiGe BiCMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 10, pp. 2239–2255, Oct. 2015.
- [38] D. Grinberg, "Notes on the combinatorial fundamentals of algebra," Tech. Rep., 2016.



**Yahia Shabara** (Student Member, IEEE) received the B.Sc. degree in electrical engineering from Alexandria University, Alexandria, Egypt, in 2012, and the M.Sc. degree in wireless communications from Nile University, Giza, Egypt, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA. His research interests include wireless communications, computer networks, machine learning, information theory, and network security.



**Eylem Ekici** (Fellow, IEEE) received the B.S. and M.S. degrees in computer engineering from Bogazici University, Turkey, in 1997 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, in 2002. He is currently a Professor with the Department of Electrical and Computer Engineering, The Ohio State University. His current research interests include wireless communication systems and networks, with a focus on algorithm design and resource management for mmWave, dynamic spectrum, and vehicular communication systems. He served as the General Co-Chair of ACM MobiCom 2012 and ACM MobiHoc 2021. He was also the TPC Co-Chair of the IEEE INFOCOM 2017. He is an Associate Editor-in-Chief of IEEE TRANSACTIONS ON MOBILE COMPUTING, and a former Associate Editor of IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, and *Computer Networks* (Elsevier).



**Can Emre Koksal** (Senior Member, IEEE) received the B.S. degree in electrical engineering from Middle East Technical University, in 1996, and the S.M. and Ph.D. degrees in electrical engineering and computer science from MIT, in 1998 and 2002, respectively.

He was a Postdoctoral Fellow with MIT, until 2004, and a Senior Researcher with EPFL, until 2006. Since 2006, he has been with the Electrical and Computer Engineering Department, The Ohio State University, where he is currently a Professor. He is also the Founder and CEO of DATAnchor and several technologies he has built have been licensed to a few enterprises and start-ups. His research interests include wireless communications, cybersecurity, communication networks, information theory, stochastic processes, and financial economics. He was a recipient of the Columbus Business First—Inventor of the Year Award, in 2020, the National Science Foundation CAREER Award, in 2011, a finalist of the Bell Labs Prize, in 2015, the OSU CoE Innovator Award, in 2016 and 2020, and the OSU CoE Lumley Research Award, in 2011 and 2017. He was a co-recipient of an HP Labs—Innovation Research Award, in 2011. He received the Best Paper Award in IEEE WiOpt 2018 and the Best Student Paper Candidate in ACM MOBICOM 2005. He has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and *Computer Networks* (Elsevier).