

# Coded Modulation Schemes for Voronoi Constellations

Shen Li, Ali Mirani, Magnus Karlsson, *Fellow, IEEE, Fellow, OSA*, and Erik Agrell, *Fellow, IEEE*

**Abstract**—Multidimensional Voronoi constellations (VCs) are shown to be more power-efficient than quadrature amplitude modulation (QAM) formats given the same uncoded bit error rate, and also have higher achievable information rates. However, a coded modulation scheme to sustain these gains after forward error correction (FEC) coding is still lacking. This paper designs coded modulation schemes with soft-decision FEC codes for VCs, including bit-interleaved coded modulation (BICM) and multilevel coded modulation (MLCM), together with three bit-to-integer mapping algorithms and log-likelihood ratio calculation algorithms. Simulation results show that VCs can achieve up to 1.84 dB signal-to-noise ratio (SNR) gains over QAM with BICM, and up to 0.99 dB SNR gains over QAM with MLCM for the additive white Gaussian noise channel, with a surprisingly low complexity.

**Index Terms**—Bit-interleaved coded modulation, constellation labeling, forward error correction coding, geometric shaping, information rates, lattices, multilevel coding, multidimensional modulation formats, Ungerboeck SP, Voronoi constellations.

## I. INTRODUCTION

ADVANCED multidimensional (MD) modulation formats are designed to have larger minimum Euclidean distance at the same average symbol energy than traditional two-dimensional (2D) quadrature amplitude modulation (QAM) formats. MD Voronoi constellations (VCs) are such a structured modulation format, comprising a coding lattice and a shaping lattice, the latter being a sublattice of the coding lattice [?], [?]. The coding lattice determines how constellation points are packed, resulting in a coding gain over the cubic packing. The shaping lattice of VCs determines the boundary shape of the constellation, achieving a shaping gain over a hypercubic boundary. When applying soft-decision (SD) forward error correction (FEC) codes to VCs, the coding gain of FEC coding might fully or partially cover the coding gain of VCs. On the other hand, the shaping gain, which comes from improved signal distribution and is asymptotically 1.53 dB over QAM for the average power-constrained additive white Gaussian noise (AWGN) channel, cannot be realized by FEC coding.

This research was funded in part by the Swedish Research Council (VR) under grants no. 2017-03702 and no. 2021-03709 and the Knut and Alice Wallenberg Foundation under grant no. 2018.0090.

S. Li and E. Agrell are with the Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden. e-mail: shenl@chalmers.se.

A. Mirani was with the Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden and is now with Ericsson AB, 417 56, Gothenburg, Sweden.

M. Karlsson is with the Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg.

Manuscript received July xx, 2023; revised xx xx, 2023.

VCs can have low-complexity encoding and decoding algorithms, i.e., mapping integers to constellation points and vice versa [?], [?], [?], [?], [?], which entirely avoid the need to store and process all constellation points individually in the transmitter and receiver. VCs have shown better bit error rate (BER) performance than Gray-labeled QAM in uncoded systems [?], [?], [?], [?], [?]. Mutual information (MI) and generalized mutual information (GMI) have also been studied for VCs in [?], [?], showing high gains over QAM.

In modern communication systems, SD FEC codes are usually used to provide significant power gains over uncoded systems. The joint design of the modulation format, labeling rule, and FEC codes is called a coded modulation (CM) scheme. The most widely used CM scheme is Gray-labeled QAM with bit-interleaved coded modulation (BICM), and serves as a benchmark for other CM schemes. In [?], a multilevel coded modulation (MLCM) scheme with SD FEC codes was proposed for the Hurwitz constellation, in which constellation points are a finite set of lattice points from the 4D checkerboard lattice  $D_4$ , and the boundary is hypercubic. The performance gains over QAM comes from the coding gain of  $D_4$ . In [?], [?], CM schemes with non-binary SD codes are designed for the 4D Welti constellation, which has constellation points from the  $D_4$  lattice and uses a hypersphere boundary. The performance gains over QAM with BICM comes from the shaping and coding gains of the Welti constellation itself, and the FEC codes (nonbinary codes or multilevel codes) as well.

However, a CM scheme to preserve VCs' high shaping gains and coding gains after FEC decoding is still lacking. Designing CM schemes for MD VCs that outperforms QAM with BICM is challenging, due to that no Gray labeling exists for MD VCs, and the resulting penalty from a non-Gray labeling might cancel out the shaping and coding gains of VCs.

In this paper, we focus on MD VCs with a cubic coding lattice, i.e., VCs having high shaping gains but no coding gain. The absent coding gain is instead achieved by FEC codes. We design several CM schemes with SD FEC codes for VCs for the first time, including BICM and MLCM. The considered VCs are of up to 24 dimensions and have up to  $5 \times 10^{27}$  constellation points with high spectral efficiencies, in order to achieve high shaping gains. However, the proposed labeling rule and log-likelihood ratio (LLR) calculation algorithm have a very low complexity. Moreover, the FEC overhead is lower than commonly used overheads (15%–25%) of high-performance SD FEC codes for optical communications. Thus, the application scenario of the proposed CM scheme would be ultra high-rate transmission systems, such as the upcoming 800

Gbps and 1.25 Tbps standards for fiber communications.

*Notation:* Bold lowercase symbols denote row vectors and bold uppercase symbols denote random vectors or matrices. All-zero and all-one vectors are denoted by  $\mathbf{0}$  and  $\mathbf{1}$ , respectively. Vector inequalities are performed element-wise, e.g., for vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , the inequality  $\mathbf{x} \leq \mathbf{y}$  refers to  $x_i \leq y_i$  for  $i = 1, \dots, n$ . The sets of integer, positive integer, real, complex, and natural numbers are denoted by  $\mathbb{Z}, \mathbb{Z}^+, \mathbb{R}, \mathbb{C}$ , and  $\mathbb{N}$ , respectively. Other sets are denoted by calligraphic symbols. Rounding a vector to its nearest integer vector is denoted by  $\lfloor \cdot \rfloor$ , in which ties are broken arbitrarily. The cardinality of a set or the order of a lattice partition is denoted by  $|\cdot|$ .

## II. LATTICES AND VCS

An  $n$ -dimensional lattice  $\Lambda$  is an infinite set of points spanned by the rows of its  $n \times n$  generator matrix  $\mathbf{G}_\Lambda$  with all integer coefficients, i.e.,

$$\Lambda \triangleq \{\mathbf{u}\mathbf{G}_\Lambda : \mathbf{u} \in \mathbb{Z}^n\}. \quad (1)$$

The *closest lattice point quantizer* of a lattice  $\Lambda$ , denoted by  $\mathcal{Q}_\Lambda(\cdot)$ , finds the closest lattice point in  $\Lambda$  of an arbitrary point  $\mathbf{x} \in \mathbb{R}^n$ , i.e.,

$$\mathcal{Q}_\Lambda(\mathbf{x}) = \arg \min_{\boldsymbol{\lambda} \in \Lambda} \|\mathbf{x} - \boldsymbol{\lambda}\|^2. \quad (2)$$

A sublattice  $\Lambda'$  of  $\Lambda$ , denoted by  $\Lambda' \subseteq \Lambda$ , contains a subset of the lattice points<sup>1</sup> of  $\Lambda$ , which is spanned by the generator matrix  $\mathbf{G}_{\Lambda'}$  satisfying

$$\mathbf{G}_{\Lambda'} = \mathbf{J}\mathbf{G}_\Lambda, \quad (3)$$

with an  $\mathbf{J} \in \mathbb{Z}^{n \times n}$ . The *lattice partition*  $\Lambda/\Lambda'$  partitions  $\Lambda$  into  $|\Lambda/\Lambda'| = |\det \mathbf{G}_{\Lambda'}|/|\det \mathbf{G}_\Lambda| = |\det \mathbf{J}|$  disjoint *cosets* of  $\Lambda'$  [?], and  $|\Lambda/\Lambda'|$  is called the *partition order*. If one arbitrary lattice point is selected from each of these cosets, a set of *coset representatives* (not unique) is formed, denoted by  $[\Lambda/\Lambda']$ . Then every lattice point  $\boldsymbol{\lambda} \in \Lambda$  can be written as

$$\boldsymbol{\lambda} = \mathbf{c} + \boldsymbol{\lambda}', \quad (4)$$

where  $\boldsymbol{\lambda}' \in \Lambda'$  and  $\mathbf{c} \in [\Lambda/\Lambda']$  can be uniquely labeled by  $k = \log_2(|\Lambda/\Lambda'|)$  bits if  $|\Lambda/\Lambda'|$  is a power of 2. The whole lattice  $\Lambda$  is decomposed as

$$\Lambda = [\Lambda/\Lambda'] + \Lambda'. \quad (5)$$

A *partition chain*, formed by a sequence of lattices  $\Lambda^0 \supseteq \Lambda^1 \supseteq \dots \supseteq \Lambda^q$  with  $q \in \mathbb{Z}^+$ , is denoted by  $\Lambda^0/\Lambda^1/\dots/\Lambda^q$  [?]. Every lattice point  $\boldsymbol{\lambda}_0 \in \Lambda^0$  can be written as

$$\boldsymbol{\lambda}_0 = \sum_{i=1}^q \mathbf{c}_i + \boldsymbol{\lambda}_q, \quad (6)$$

where  $\mathbf{c}_i \in [\Lambda^{i-1}/\Lambda^i]$  for  $i = 1, \dots, q$  and  $\boldsymbol{\lambda}_q \in \Lambda^q$ . If the partition orders  $|\Lambda^{i-1}/\Lambda^i|$  for  $i = 1, \dots, q$  are powers of 2,  $\boldsymbol{\lambda}_0$  can be uniquely labeled by the binary tuple

$$\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_q), \quad (7)$$

<sup>1</sup>Arbitrary points in  $\mathbb{R}^n$  are referred to “points” in this paper. To avoid ambiguity, “lattice point” is used when a point also belongs to a lattice. Later throughout the paper, “constellation points” refers to the points in VCs.

where  $\mathbf{b}_i$  is the bit labels of  $\mathbf{c}_i$  with the length of  $k_i$  for  $i = 1, \dots, q$  and the total length of  $\mathbf{b}$  is  $\sum_{i=1}^q k_i = \log_2(|\Lambda^0/\Lambda^q|)$ . The lattice  $\Lambda^0$  can be decomposed as

$$\Lambda^0 = [\Lambda^0/\Lambda^1] + \dots + [\Lambda^{q-1}/\Lambda^q] + \Lambda^q. \quad (8)$$

An  $n$ -dimensional VC is a set of coset representatives of a lattice partition  $\Lambda_c/\Lambda_s$ , where  $\Lambda_c$  is called the *coding lattice*,  $\Lambda_s$  is called the *shaping lattice*, and the partition order is  $M = |\Lambda_c/\Lambda_s|$ , which is a power of 2 to enable binary labeling with  $m = \log_2(M)$  bits. The VC points  $\Gamma$  are defined as all lattice points in the translated  $\Lambda_c$  having the all-zero point  $\mathbf{0}$  as their closest lattice point in  $\Lambda_s$ , i.e. [?],

$$\Gamma \triangleq \{\mathbf{x} \in (\Lambda_c - \mathbf{a}) : \mathcal{Q}_{\Lambda_s}(\mathbf{x}) = \mathbf{0}\}, \quad (9)$$

where the *offset vector*  $\mathbf{a} \in \mathbb{R}^n$  is usually optimized to minimize the average symbol energy [?]

$$E_s = \frac{1}{M} \sum_{\mathbf{x} \in \Gamma} \|\mathbf{x}\|^2. \quad (10)$$

The *spectral efficiency* [?], [?], [?] in bits per two-dimensional (2D) symbol for the uncoded system is defined as

$$\beta = \frac{2m}{n} \text{ [bits/2D-symbol]}. \quad (11)$$

The signal-to-noise ratio (SNR) is defined as  $E_s/\sigma_{\text{tot}}^2$ , where  $\sigma_{\text{tot}}^2$  is the total noise variance.

## III. LABELING OF VCS

### A. Encoding and decoding

A *labeling function* is a map from binary labels of length  $m$  to constellation points. For the considered VCs based on the lattice partition  $\mathbb{Z}^n/\Lambda_s$ , we divide the labeling algorithm into two steps: first from binary labels to integers and then from integers to VC points, i.e.,  $\{0, 1\}^m \rightarrow \mathcal{U} \rightarrow \Gamma$ . The integer set  $\mathcal{U}$  is formed by first writing the generator matrix of the shaping lattice  $\Lambda_s$  as a lower-triangular form  $\mathbf{G}_s$  with diagonal elements  $\mathbf{h} = (h_1, \dots, h_n)$ , and then letting

$$\mathcal{U} = \{\mathbf{u} : \mathbf{0} \leq \mathbf{u} \leq \mathbf{h} - \mathbf{1}\}. \quad (12)$$

*Encoding:* The function that maps binary labels to integers is denoted by  $f : \{0, 1\}^m \rightarrow \mathcal{U}$ , which will be discussed in Sections ??, ??, and ?. The algorithm that maps integers  $\mathcal{U}$  to constellation points  $\Gamma$  was proposed in [?] and summarized in [?, Alg. 1], which is denoted by  $g : \mathcal{U} \rightarrow \Gamma$  in this paper.

*Decoding:* After receiving a noisy version of a VC point, the algorithm that maps it back to an estimate of the transmitted VC point was proposed in [?] and summarized in [?, Alg. 2], which is denoted by a function  $w : \mathbb{R}^n \rightarrow \mathcal{U}$  in this paper. Then the binary labels are obtained by the inverse of  $f$ , i.e.,  $f^{-1} : \mathcal{U} \rightarrow \{0, 1\}^m$ .

The rest of this section introduces three different mapping functions  $f$  for the considered VCs based on the lattice partition  $\mathbb{Z}^n/\Lambda_s$ . Section ?? reviews the Gray mapping proposed in [?]. Section ?? proposes a new mapping function, the set partitioning (SP) mapping based on Ungerboeck's SP and lattice partition chains. Another new hybrid mapping function combining the SP mapping and pseudo-Gray mapping is then proposed in Section ??.

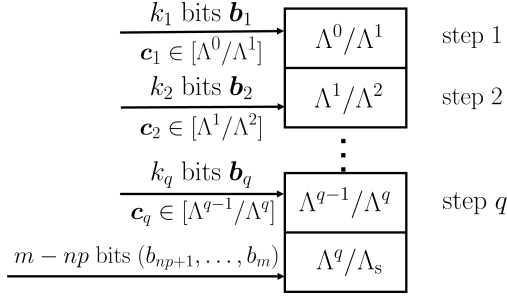


Fig. 1: Illustration of the labeling of a partition chain  $\Lambda^0/\Lambda^1/\dots/\Lambda^{q-1}/\Lambda^q/\Lambda_s$ .

TABLE I: Example partition chains in the SP mapping for MD VCs with a cubic coding lattice  $\mathbb{Z}^n$ .

$n = 2$	$\mathbb{Z}^2/D_2/2\mathbb{Z}^2/2D_2/4\mathbb{Z}^2/4D_2/\dots$				
Step $i$	1	2	3	4	5
$k_i$	1	1	1	1	1
$d_i^2$	2	4	8	16	32
$n = 4$	$\mathbb{Z}^4/D_4/2\mathbb{Z}^4/2D_4/4\mathbb{Z}^4/4D_4/\dots$				
Step $i$	1	2	3	4	5
$k_i$	1	1	1	1	1
$d_i^2$	2	4	8	16	32
$n = 8$	$\mathbb{Z}^8/D_8/E_8\mathbf{R}_8/2E_8/2E_8\mathbf{R}_8/4E_8/\dots$				
Step $i$	1	2	3	4	5
$k_i$	1	3	4	4	4
$d_i^2$	2	4	8	16	32
$n = 16$	$\mathbb{Z}^{16}/D_{16}/D_{16}\mathbf{R}_{16}/\Lambda_{16}/\Lambda_{16}\mathbf{R}_{16}/2\Lambda_{16}/\dots$				
Step $i$	1	2	3	4	5
$k_i$	1	8	3	8	8
$d_i^2$	2	4	8	16	32

### B. Gray mapping

In [?], a mapping method between binary labels and integers is proposed in order to minimize the uncoded BER of VCs, which works in the following way.

First, the binary label  $\mathbf{b} \in \{0, 1\}^m$  is divided into  $n$  blocks according to  $\mathbf{h}$ ,

$$\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n),$$

each of which has  $\log_2(h_i)$  bits for  $i = 1, \dots, n$ , and  $\sum_{i=1}^n \log_2(h_i) = m$ . Then  $\mathbf{b}_i$  is converted to an integer  $u_i$  using the binary reflected Gray code (BRGC) [?] for  $i = 1, \dots, n$ , yielding

$$\mathbf{u} = (u_1, \dots, u_n). \quad (13)$$

The above procedures converting  $\mathbf{b}$  to  $\mathbf{u}$  according to the BRGC is denoted by  $f_{\text{BRGC}}(\mathbf{b}, \mathbf{h})$ , and the inverse process of converting an integer vector  $\mathbf{u}$  to a binary vector  $\mathbf{b}$  is denoted by  $f_{\text{BRGC}}^{-1}(\mathbf{u}, \mathbf{h})$  in this paper. After mapping integers to VC points using function  $g$  defined in section ??, the labeling is not true Gray, but close to Gray, which is called ‘‘pseudo-Gray’’ labeling.

### C. SP mapping

Ungerboeck’s SP concept [?] maps binary labels to 1D or 2D constellation points by successively partitioning the

TABLE II: An example look-up table for the coset representatives of the lattice partition  $D_8/E_8\mathbf{R}_8$  and their bit labels.

$[D_8/E_8\mathbf{R}_8]$	labels
(00000000)	(000)
(01010000)	(001)
(00011000)	(010)
(01001000)	(011)
(11000000)	(100)
(10010000)	(101)
(11011000)	(110)
(10001000)	(111)

constellation into two subsets at each bit level in order to maximize the intra-set minimum squared Euclidean distance (MSED) at each level, so that unequal error protection can be implemented on different bit levels. Since all partition orders are 2, Ungerboeck’s SP is also called binary SP. Binary SP has been applied to 1D, 2D [?], [?], [?], and 4D [?], [?], [?], [?] signal constellations. When binary SP is applied in larger than 2 dimensions, the MSED might not increase at every bit level. Binary SP requires one encoder and one decoder at each bit level, which has a high complexity in FEC for large constellations. Generalized from the binary SP, signal sets can be partitioned into multiple subsets based on the concept of cosets [?], [?], [?], [?], which enables SP in higher dimensions [?], [?], [?] and increasing MSED at every partition level. How the coset representatives are labeled at each partition level is not specified. In this section, we introduce a systematic algorithm for mapping bits  $\mathbf{b} \in \{0, 1\}^m$  to integers  $\mathbf{u} \in \mathcal{U}$  based on SP such that the MSED doubles at every partition level for very large MD VCs based on the lattice partition  $\mathbb{Z}^n/\Lambda_s$ .

For large constellations, after getting a sufficiently large intra-set MSED, it is reasonable to not partition the remaining subsets and leave the corresponding bit levels uncoded. One convenient way is to stop partitioning when a scaled integer lattice  $2^p\mathbb{Z}^n$  is obtained, where  $p \in \mathbb{N}$ . Then we map the last  $m - np$  bits to integers according to BRGC to minimize the BER for the uncoded bits.

The preprocessing of the proposed SP mapping works as follows. First,  $q$  intermediate lattices  $\Lambda^1, \dots, \Lambda^q$  are found to form the partition chain

$$\Lambda^0/\Lambda^1/\dots/\Lambda^q/\Lambda_s, \quad (14)$$

where  $\Lambda^0 = \mathbb{Z}^n$  and  $\Lambda^q = 2^p\mathbb{Z}^n$  is where to stop the partition. The partition chain should satisfy  $\Lambda^0 \supset \Lambda^1 \supset \dots \supset \Lambda^q \supseteq \Lambda_s$  and have increasing MSEDs of  $d_i^2 = 2^i$  for  $i = 1, \dots, q$ . The order of each partition step  $|\Lambda^{i-1}/\Lambda^i| = 2^{k_i}$  for  $i = 1, \dots, q$  and  $\sum_{i=1}^q k_i = \log_2(|\Lambda^0/\Lambda^q|) = np$ . At every partition step  $i$ , all coset representatives  $[\Lambda^{i-1}/\Lambda^i]$  are labeled by  $k_i$  bits and the mapping rules are stored in a look-up table  $\mathbf{C}_i$ . Conventionally, the set of coset representatives contains the all-zero lattice point labeled by the all-zero binary tuple. Fig. ?? illustrates the mapping for the partition chain in general. Table ?? lists some example partition chains and their intra-set MSEDs for MD VCs with a cubic coding lattice. These partition chains contain commonly used lattices

as intermediate lattices including the  $n$ -dimensional checkerboard lattice  $D_n$ , 8-dimensional (8D) Gosset lattice  $E_8$ , 16-dimensional (16D) Barnes–Wall lattice  $\Lambda_{16}$  [?], and the 24-dimensional (24D) Leech lattice  $\Lambda_{24}$  [?, Ch. 4]. The  $n \times n$  matrix  $\mathbf{R}_n$  is an integer orthonormal rotation matrix with a determinant of  $\det \mathbf{R}_n = 2^{n/2}$  [?], [?]. When multiplied with a lattice generator matrix on the right, it rotates every two dimensions of the lattice by  $45^\circ$  and rescales it by  $\sqrt{2}$ . The size of the look-up table  $\mathbf{C}_i$  is  $2^{k_i}$ , which is not more than  $2^8$  in Table ?? and much smaller than a table for the whole VC.

As an example, a set of coset representatives of the partition  $D_8/E_8\mathbf{R}_8$  and one set of possible bit labels are listed in Table ?. Note that neither the choice of the set of coset representatives nor the mapping within the look-up table is unique, and both of them are arbitrarily selected. The effects of different choices on the performance are not studied. However, we conjecture that there would be no big difference since the intra-set MSED cannot be increased by further partitioning the set of coset representatives.

The SP demapping  $f_{\text{SP}}^{-1}$  from an integer vector  $\mathbf{u}$  to its bit labels  $\mathbf{b}$  works as follows. Given an integer vector  $\mathbf{u} \in \mathcal{U}$ , starting from the first partition step  $\Lambda^0/\Lambda^1$ , we know that  $\mathbf{u}$  belongs to one coset of this partition since  $\mathbf{u} \in \Lambda^0 = \mathbb{Z}^n$ . By full search among a certain set of coset representatives  $[\Lambda^0/\Lambda^1]$ , there must be only one  $\mathbf{c}_1 \in [\Lambda^0/\Lambda^1]$  such that  $(\mathbf{u} - \mathbf{c}_1) \cdot \mathbf{G}_{\Lambda^1}^{-1}$  yields an integer vector, where  $\mathbf{G}_{\Lambda^1}$  is the generator matrix of  $\Lambda^1$ . The bit labels  $\mathbf{b}_1$  corresponding to  $\mathbf{c}_1$  is found in the look-up table  $\mathbf{C}_1$ . Then  $\mathbf{u} - \mathbf{c}_1$  is a lattice point of  $\Lambda^1$ , which must belong to a certain coset of  $\Lambda^1/\Lambda^2$ . A vector  $\mathbf{c}_2 \in [\Lambda^1/\Lambda^2]$  is found such that  $(\mathbf{u} - \mathbf{c}_1 - \mathbf{c}_2) \cdot \mathbf{G}_{\Lambda^2}^{-1}$  yields an integer vector, where  $\mathbf{G}_{\Lambda^2}$  is the generator matrix of  $\Lambda^2$  and the bit labels  $\mathbf{b}_2$  corresponding to  $\mathbf{c}_2$  is found in the look-up table  $\mathbf{C}_2$ . The procedure is repeated until all  $\mathbf{c}_i$  and  $\mathbf{b}_i$  are obtained for  $i = 1, \dots, q$ .

Next, the remaining  $m - np$  bit labels for the partition  $2^p\mathbb{Z}^n/\Lambda_s$  are found as follows. First, the coset representatives of the partition  $\mathbb{Z}^n/2^p\mathbb{Z}^n$  are set as

$$\mathcal{S} = [\mathbb{Z}^n/2^p\mathbb{Z}^n] = \{\mathbf{s} \in \mathbb{Z}^n : \mathbf{0} \leq \mathbf{s} \leq (2^p - 1) \cdot \mathbf{1}\}. \quad (15)$$

There must be a unique  $\mathbf{s} \in \mathcal{S}$  such that  $\mathbf{u} - \mathbf{s} \in 2^p\mathbb{Z}^n$ , and  $\mathbf{s}$  can be easily found by

$$\mathbf{s} = \mathbf{u} \bmod 2^p, \quad (16)$$

where mod is the modulo operator that takes the remainder of a vector element-wise and returns a vector. The purpose of choosing such a set of coset representatives is to make sure that  $\mathbf{u} - \mathbf{s}$  still falls within the range of  $\mathcal{U}$ , i.e.,  $\mathbf{u} - \mathbf{s} \in \mathcal{U} \cap 2^p\mathbb{Z}^n$ . Then we know that

$$\frac{\mathbf{u} - \mathbf{s}}{2^p} \in \mathbb{Z}^n \quad (17)$$

with the range

$$\mathbf{0} \leq \frac{\mathbf{u} - \mathbf{s}}{2^p} \leq \frac{\mathbf{h}}{2^p} - \mathbf{1}. \quad (18)$$

The bit labels of  $(\mathbf{u} - \mathbf{s})/2^p$  can be obtained by converting each decimal element to bits according to BRGC, i.e.,

$$(b_{np+1}, \dots, b_m) = f_{\text{BRGC}}^{-1} \left( \frac{\mathbf{u} - \mathbf{s}}{2^p}, \frac{\mathbf{h}}{2^p} \right). \quad (19)$$

---

### Algorithm 1 SP mapping $f_{\text{SP}}$

---

Input:  $\mathbf{b}$ . Output:  $\mathbf{u}$ .

Preprocessing: The partition chain  $\Lambda^0/\Lambda^1/\dots/\Lambda^q/\Lambda_s$  is given, where all partition orders  $|\Lambda^{i-1}/\Lambda^i|$  for  $i = 1, \dots, q$  are powers of 2. Set the look-up tables  $\mathbf{C}_i$  between all coset representatives and their corresponding bit labels for all partition steps. Divide the first  $np$  bits of  $\mathbf{b}$  into  $q$  blocks  $\mathbf{b}_i$  for  $i = 1, \dots, q$ , each of which has  $k_i$  bits. Find a lower-triangular generator matrix  $\mathbf{G}_s$  of  $\Lambda_s$  and denote the diagonal elements of  $\mathbf{G}_s$  as  $\mathbf{h}$ .

- 1: Find the corresponding  $\mathbf{c}_i$  of  $\mathbf{b}_i$  according to  $\mathbf{C}_i$  for  $i = 1, \dots, q$ .
  - 2: Let  $\mathbf{c} \leftarrow \sum_{i=1}^q \mathbf{c}_i$
  - 3: Let  $\mathbf{s} \leftarrow \mathbf{c} - \lfloor \mathbf{c}/2^p \rfloor \cdot 2^p$
  - 4: Let  $\mathbf{t} \leftarrow f_{\text{BRGC}}((b_{np+1}, \dots, b_m), \mathbf{h}/2^p)$
  - 5: Let  $\mathbf{u} \leftarrow \mathbf{s} + 2^p\mathbf{t}$
- 

---

### Algorithm 2 SP demapping $f_{\text{SP}}^{-1}$

---

Input:  $\mathbf{u}$ . Output:  $\mathbf{b}$ .

The partition chain  $\Lambda^0/\Lambda^1/\dots/\Lambda^q/\Lambda_s$  is given, where all partition orders  $|\Lambda^{i-1}/\Lambda^i|$  are powers of 2. Set the look-up tables  $\mathbf{C}_i$  between all coset representatives and their corresponding bit labels for all steps  $i = 1, \dots, q$ . Set the set of coset representatives of the partition  $\Lambda^0/\Lambda^q$  as  $\mathcal{S}$  defined in (??). Find a lower-triangular generator matrix  $\mathbf{G}_s$  of  $\Lambda_s$  and denote the diagonal elements of  $\mathbf{G}_s$  as  $\mathbf{h}$ .

- 1: Let  $\mathbf{v} \leftarrow \mathbf{u}$
  - 2: **for**  $i = 1, \dots, q$  **do**
  - 3: Find the only  $\mathbf{c}_i \in [\Lambda^{i-1}/\Lambda^i]$  such that  $\mathbf{u} - \mathbf{c}_i \in \Lambda^i$
  - 4: Let  $\mathbf{b}_i$  be the bit labels of  $\mathbf{c}_i$  according to  $\mathbf{C}_i$
  - 5: Let  $\mathbf{u} \leftarrow \mathbf{u} - \mathbf{c}_i$
  - 6: **end for**
  - 7: Let  $\mathbf{s} \leftarrow \mathbf{v} \bmod 2^p$
  - 8: Let  $(b_{np+1}, \dots, b_m) = f_{\text{BRGC}}^{-1}((\mathbf{v} - \mathbf{s})/2^p, \mathbf{h}/2^p)$
  - 9: Let  $\mathbf{b} \leftarrow (\mathbf{b}_1, \dots, \mathbf{b}_q, b_{np+1}, \dots, b_m)$
- 

Now we describe the SP mapping  $f_{\text{SP}}$  from the bit labels  $\mathbf{b}$  to the integer vector  $\mathbf{u}$ . Given bit labels  $\mathbf{b} \in \{0, 1\}^m$ , the first  $np$  bits are divided into  $q$  blocks  $\mathbf{b}_i$  for  $i = 1, \dots, q$ , each of which has  $k_i$  bits and indicates a coset representative  $\mathbf{c}_i$  according to the look-up  $\mathbf{C}_i$ . Then  $\mathbf{c} = \sum_{i=1}^q \mathbf{c}_i$  indicates a coset representative of the lattice partition  $\mathbb{Z}^n/2^p\mathbb{Z}^n$ , but  $\mathbf{c}$  might not belong to  $\mathcal{S}$ , which can be converted to  $\mathbf{s} \in \mathcal{S}$  by

$$\mathbf{s} = \mathbf{c} \bmod 2^p. \quad (20)$$

Now we know that  $\mathbf{u} - \mathbf{s} \in 2^p\mathbb{Z}^n$ . The remaining  $m - np$  bits of  $\mathbf{b}$  indicate an integer vector

$$\mathbf{t} = f_{\text{BRGC}} \left( (b_{np+1}, \dots, b_m), \frac{\mathbf{h}}{2^p} \right). \quad (21)$$

Finally,  $\mathbf{u}$  is obtained by

$$\mathbf{u} = \mathbf{s} + 2^p\mathbf{t}. \quad (22)$$

Algorithms ?? and ?? summarize the SP mapping process between  $\mathbf{u}$  and  $\mathbf{b}$  for VCs with a cubic coding lattice.

### D. Hybrid mapping

This mapping is a special case of the SP mapping, which is carefully designed for the considered VCs based on the lattice partition  $\mathbb{Z}^n/\Lambda_s$ . The idea is to only consider  $q$  intermediate lattices which are a multiple of the cubic lattice, i.e.,  $\Lambda^i = 2^{p_i}\mathbb{Z}^n$  for  $i = 0, \dots, q$  with positive integers  $p_1 < p_2 < \dots < p_q = p$  and  $p_0 = 0$ . This yields a partition chain  $2^{p_0}\mathbb{Z}^n/2^{p_1}\mathbb{Z}^n/2^{p_2}\mathbb{Z}^n/\dots/2^{p_q}\mathbb{Z}^n/\Lambda_s$ . Thus, the order of each partition step is  $|\Lambda^{i-1}/\Lambda^i| = 2^{k_i} = 2^{n(p_i-p_{i-1})}$  for  $i = 1, \dots, q$  and  $\sum_{i=1}^q k_i = \log_2(|\Lambda^0/\Lambda^q|) = np$ . The intra-set MSSED is  $d_i^2 = 2^{p_i}$  at the  $i$ th partition step. The coset representatives in each partition step is simply set as

$$C_i = [2^{p_{i-1}}\mathbb{Z}^n/2^{p_i}\mathbb{Z}^n] = \{c : 0 \leq c \leq (2^{p_i-p_{i-1}} - 1) \cdot \mathbf{1}\}, \quad (23)$$

for  $i = 1, \dots, q$ , which is labeled by  $k_i = n(p_i - p_{i-1})$  bits for  $i = 1, \dots, q$ . Thanks to that these intermediate lattices are a multiple of  $\mathbb{Z}^n$ , no full search from  $C_i$  is needed to find the unique coset representative  $c_i \in C_i$  as in the SP mapping. The coset representative  $c_i$  can be mapped to  $n(p_i - p_{i-1})$  bits by directly converting each element of  $c_i$  to  $p_i - p_{i-1}$  bits according to BRGC, i.e.,

$$b_i = f_{\text{BRGC}}^{-1}(c_i, 2^{p_i-p_{i-1}} \cdot \mathbf{1}), \quad (24)$$

for  $i = 1, \dots, q$ .

The hybrid demapping  $f_H^{-1}$  from an integer vector  $u$  to its bit labels  $b$  works as follows. Given an integer vector  $u \in \mathcal{U}$ , starting from the first partition step  $\mathbb{Z}^n/2^{p_1}\mathbb{Z}^n$ , the unique  $c_1 \in C_1$  can be easily found by

$$c_1 = u \bmod 2^{p_1}. \quad (25)$$

Then  $u - c_1$  is a lattice point of  $2^{p_1}\mathbb{Z}^n$ , which must belong to a certain coset of  $2^{p_1}\mathbb{Z}^n/2^{p_2}\mathbb{Z}^n$ . Then the unique  $c_2$  is obtained by

$$c_2 = (u - c_1) \bmod 2^{p_2}. \quad (26)$$

Repeating this procedure,  $c_i$  is found successively by

$$c_i = \left(u - \sum_{j=1}^{i-1} c_j\right) \bmod 2^{p_i} \quad (27)$$

for  $i = 1, \dots, q$ . Then the first  $np$  bit labels of  $u$  can be obtained by (??). Similar to the SP mapping, the remaining  $m - np$  bits are obtained by (??), where  $s = \sum_{i=1}^p c_i$  in this case.

The hybrid mapping  $f_H$  finding the corresponding integer vector  $u$  of bit labels  $b$  works as follows. Given bit labels  $b \in \{0, 1\}^m$ ,  $c_i \in C_i$  can be directly obtained by

$$c_i = f_{\text{BRGC}}(b_i, 2^{p_i-p_{i-1}} \cdot \mathbf{1}). \quad (28)$$

The integer vector  $s = \sum_{i=1}^p c_i$  must belong to the set  $\mathcal{S}$  defined in (??) due to the definition of  $C_i$  in (??). Then  $u$  is obtained combining (??) and (??).

Algorithms ?? and ?? summarize the hybrid mapping process between  $u$  and  $b$  for VCs with a cubic coding lattice.

---

### Algorithm 3 Hybrid mapping $f_H$

---

Input:  $b$ . Output:  $u$ .

Preprocessing: Given the partition chain  $2^{p_0}\mathbb{Z}^n/2^{p_1}\mathbb{Z}^n/2^{p_2}\mathbb{Z}^n/\dots/2^{p_q}\mathbb{Z}^n/\Lambda_s$  with positive integers  $p_1 < p_2 < \dots < p_q = p$  and  $p_0 = 0$ , set  $q$  sets of coset representatives  $C_i$  as in (??) for  $i = 1, \dots, q$ . Divide the first  $np$  bits of  $b$  into  $q$  blocks  $b_i$  for  $i = 1, \dots, q$ , each of which has  $n(p_i - p_{i-1})$  bits. Find a lower-triangular generator matrix  $G_s$  and denote the diagonal elements of  $G_s$  as  $h$ .

- 1: Let  $c_i \leftarrow f_{\text{BRGC}}(b_i, 2^{p_i-p_{i-1}} \cdot \mathbf{1})$  for  $i = 1, \dots, q$
  - 2: Let  $s \leftarrow \sum_{i=1}^q c_i$
  - 3: Let  $t \leftarrow f_{\text{BRGC}}((b_{np+1}, \dots, b_m), h/2^p)$
  - 4: Let  $u \leftarrow s + 2^p t$
- 

---

### Algorithm 4 Hybrid demapping $f_H^{-1}$

---

Input:  $u$ . Output:  $b$ .

Preprocessing: Given the partition chain  $2^{p_0}\mathbb{Z}^n/2^{p_1}\mathbb{Z}^n/2^{p_2}\mathbb{Z}^n/\dots/2^{p_q}\mathbb{Z}^n/\Lambda_s$  with positive integers  $p_1 < p_2 < \dots < p_q = p$  and  $p_0 = 0$ , set  $q$  sets of coset representatives  $C_i$  as in (??) for  $i = 1, \dots, q$ . Find a lower-triangular generator matrix  $G_s$  and denote the diagonal elements of  $G_s$  as  $h$ .

- 1: **for**  $i = 1, \dots, q$  **do**
  - 2:   Let  $c_i = u \bmod 2^{p_i}$
  - 3:   Let  $b_i \leftarrow f_{\text{BRGC}}^{-1}(c_i, 2^{p_i-p_{i-1}} \cdot \mathbf{1})$
  - 4:   Let  $u \leftarrow u - c_i$
  - 5: **end for**
  - 6: Let  $(b_{np+1}, \dots, b_m) \leftarrow f_{\text{BRGC}}^{-1}(u/2^p, h/2^p)$
  - 7: Let  $b = (b_1, \dots, b_q, b_{np+1}, \dots, b_m)$
- 

*Example 1:* A simple example is a 2D VC based on the lattice partition  $\mathbb{Z}^n/4D_2$ , where  $4D_2$  is the scaled 2D checkerboard lattice with the generator matrix

$$G_s = \begin{pmatrix} 8 & 0 \\ 4 & 4 \end{pmatrix}.$$

This VC does not provide any shaping gain, which is just for simplicity of illustration. Fig. ?? illustrates the three different mapping rules  $f$  for this example VC. It can be observed that all integer points have only 1-bit difference from their nearest neighbors in the Gray mapping. The SP mapping is based on the lattice partition chain  $\mathbb{Z}^2/D_2/2\mathbb{Z}^2/4D_2$  ( $q = 2, p = 1$ ). The hybrid mapping is based on the lattice partition chain  $\mathbb{Z}^2/2\mathbb{Z}^2/4D_2$  ( $q = p = 1$ ), and  $[\mathbb{Z}^n/2\mathbb{Z}^n] = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ . For both SP and hybrid mapping, within each coset of  $2\mathbb{Z}^2/4D_2$ , the points have only 1-bit difference among the last three bits from their closest neighbors.

## IV. CM SCHEMES

The joint design of forward error correction (FEC) coding and modulation formats is called coded modulation (CM), which plays a vital part in modern communication systems. Designing a CM scheme involves a trade-off among the spectral efficiency, power, and complexity. In this section, we propose three SD CM schemes for VCs based on the

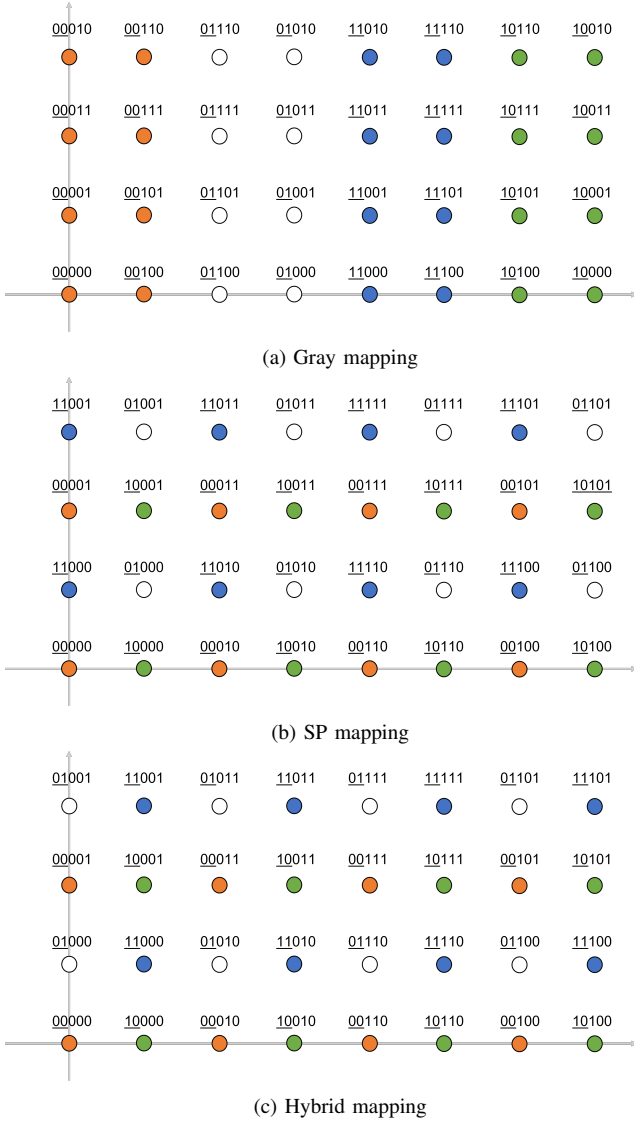


Fig. 2: Example 1: Different mapping rules  $f$  between integer vectors and bit labels for the VC based on the lattice partition  $\mathbb{Z}^2/4D_2$ . Integer points having the same first two bits are filled with the same color for better visualization and comparison.

lattice partition  $\mathbb{Z}^n/\Lambda_s$ , adopting the three labeling schemes introduced in section ??, and the computation of the log-likelihood ratios (LLRs) for SD decoding is discussed.

The designed CM schemes can be combined with an outer hard-decision (HD) code, known as concatenated coding [?], [?]. Concatenated codes are widely used in many communication standards nowadays, such as the DVB-S2 standards [?] for satellite communications and the 400ZR [?] and upcoming 800G standards for fiber-optical communications [?]. The inner CM scheme brings the uncoded BER down to a certain target BER (e.g., around  $10^{-3}$  for fiber-optic communications). Then the outer code can further eliminate the error floor and achieve a very low BER as needed. Commonly used outer codes include Reed–Solomon codes [?], turbo product codes [?], Bose–Chaudhuri–Hocquenghem codes [?], staircase codes [?], and zipper codes [?].

#### A. BICM for VCs with Gray mapping

Consider a channel with input symbols  $\mathbf{X}$  labeled by  $m$  bits  $(B_1, \dots, B_m)$  and output symbols  $\mathbf{Y}$ . The mutual information (MI) between  $\mathbf{X}$  and  $\mathbf{Y}$  is

$$\begin{aligned} I(\mathbf{Y}; \mathbf{X}) &= I(\mathbf{Y}; B_1, \dots, B_m) \\ &= I(\mathbf{Y}; B_1) + I(\mathbf{Y}; B_2|B_1) + \dots \\ &\quad + I(\mathbf{Y}; B_m|B_1, \dots, B_{m-1}) \end{aligned} \quad (29)$$

according to the chain rule. If the conditions in all conditional MIs of (??) are neglected, the BICM capacity [?]

$$I_{\text{BICM}} = \sum_{i=1}^m I(\mathbf{Y}; B_i) \leq I(\mathbf{Y}; \mathbf{X}) \quad (30)$$

is obtained. The channel is regarded as  $m$  independent bit subchannels, which can be encoded and decoded independently. BICM utilizes this concept and contains only one binary component code to protect all bit subchannels. An interleaver is added between the encoder and symbol mapper to distribute the coded bits evenly to all bit subchannels.

The Gray mapping in Section ?? maps each bit level independently, and is suitable to be combined with BICM. Fig. ?? illustrates the BICM scheme for VCs. The total rate of BICM for VCs is  $\beta R_c$  [bits/2D-symbol], where  $R_c$  is the code rate of the inner code. Decoding is based on bit LLRs, which is described below.

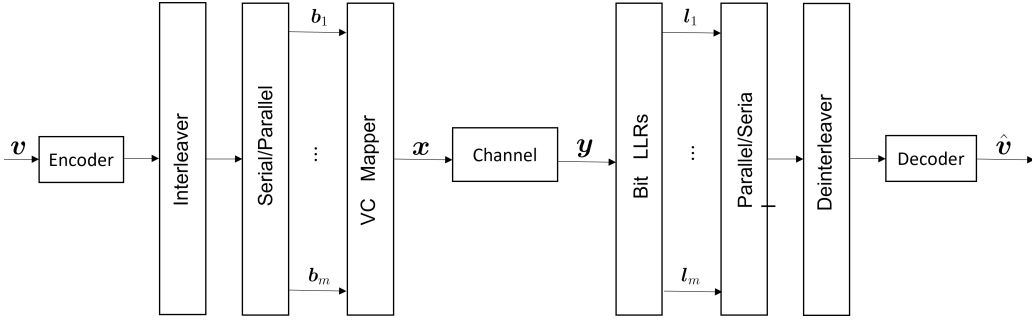
For a constellation  $\Gamma$  transmitted over the AWGN channel, the max-log approximation [?] of the  $k$ th bit after receiving a  $\mathbf{y}^j \in \mathbb{R}^n$  for  $j = 1, \dots, N/m$  is defined as

$$\text{LLR}(b_k|\mathbf{y}^j) = -\frac{1}{\sigma^2} \left( \min_{\mathbf{x} \in \Gamma^{(k,0)}} \|\mathbf{y}^j - \mathbf{x}\|^2 - \min_{\mathbf{x} \in \Gamma^{(k,1)}} \|\mathbf{y}^j - \mathbf{x}\|^2 \right), \quad (31)$$

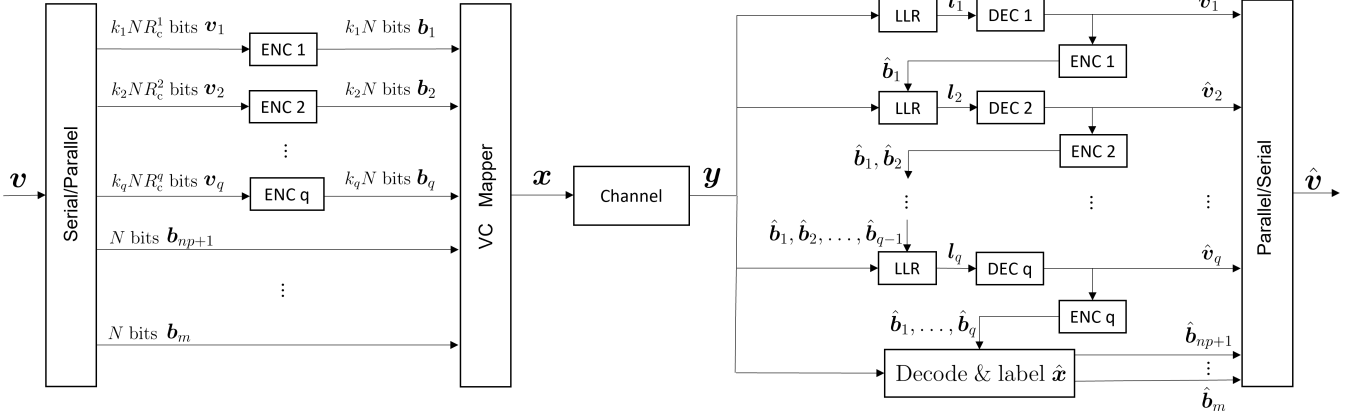
where  $\sigma^2$  is the noise power per two dimensions,  $\Gamma^{(k,0)}$  and  $\Gamma^{(k,1)}$  are the sets of constellation points with 0 and 1 at bit position  $k$ , respectively, and  $\Gamma^{(k,0)} \cup \Gamma^{(k,1)} = \Gamma$ . Computing (??) needs a full search in  $\Gamma$ , which is infeasible for very large constellations. In [?], an LLR approximation method based on importance sampling is proposed and exemplified for very large VCs based on the lattice partition  $\mathbb{Z}^n/\Lambda_s$  for the AWGN channel. The idea is to only search from a small portion of the whole constellation, which is called “importance set”. In this paper, instead of searching from a subset of the VC, we further reduce the complexity of the approximation in [?, Eq. (33)] by searching from a finite number of lattice points from  $\mathbb{Z}^n - \mathbf{a}$  that are inside a “Euclidean ball” centered at  $\lfloor \mathbf{y}^j + \mathbf{a} \rfloor$ , i.e.,

$$\mathcal{B}(\mathbf{y}^j, R^2) \triangleq \{e : \|e + \mathbf{a} - \lfloor \mathbf{y}^j + \mathbf{a} \rfloor\|^2 \leq R^2, e + \mathbf{a} \in \mathbb{Z}^n\}, \quad (32)$$

where  $\lfloor \cdot \rfloor$  represents rounding a vector to its nearest integer vector and  $R^2 \geq 0$  is the squared radius of the Euclidean ball. When the SNR is low, the points in the Euclidean ball might partially or fully fall outside of the constellation boundary. Compared with the method in [?, Eq. (33)], where the importance set is defined as  $\mathcal{B}(\mathbf{y}^j, R^2) \cap \Gamma$ , searching among all points from  $\mathcal{B}(\mathbf{y}^j, R^2)$  might return a point outside  $\Gamma$ , which causes a loss in decoding performance. However, the



(a) BICM: A block of  $NR_c$  information bits  $\mathbf{v}$  are encoded into  $N$  bits by the encoder and then permuted by the interleaver to avoid burst errors [?]. The serial bits after interleaving are converted to  $m$  parallel bit streams  $\mathbf{b}_1, \dots, \mathbf{b}_m$  of length  $N/m$ . At the time slot  $j = 1, \dots, N/m$ , a VC mapper first maps  $m$  bits  $\mathbf{b}^j = (b_1^j, \dots, b_m^j)$  to an integer  $\mathbf{u}^j$  by  $\mathbf{u}^j = f_{\text{BRGC}}(\mathbf{b}^j, \mathbf{h})$ , and then maps  $\mathbf{u}^j$  to a VC point  $\mathbf{x}^j \in \Gamma$  by  $\mathbf{x}^j = g(\mathbf{u}^j)$ . The receiver deinterleaves  $N$  independent LLRs of the bits and then uses them to decode  $\hat{\mathbf{v}}$ .



(b) MLMC: A block of serial information bits  $\mathbf{v}$  are partitioned into  $q$  parallel bit streams  $\mathbf{v}_i$  with  $k_i NR_c^i$  bits for  $i = 1, \dots, q$  and  $m - np$  parallel uncoded bit streams  $\mathbf{b}_{np+1}, \dots, \mathbf{b}_m$  with length  $N$ . Then  $\mathbf{v}_i$  is encoded into  $k_i N$  bits  $\mathbf{b}_i$  by encoder (ENC)  $i$  for  $i = 1, \dots, q$ . The VC mapper first maps bits to  $N$  integer vectors by the SP or hybrid mapping, and then encode these integer vectors into  $N$  VC points  $\mathbf{x}$ . Multistage decoding is performed at the receiver after receiving  $N$  noisy symbols  $\mathbf{y}$ . Decoder (DEC) 1 first decodes  $k_1 NR_c^1$  bits  $\hat{\mathbf{v}}_1$  back based on  $\mathbf{y}$  and  $k_1 N$  LLRs  $\mathbf{l}_1$ . Then  $\hat{\mathbf{v}}_1$  is encoded into  $\hat{\mathbf{b}}_1$  by encoder 1. Decoder  $i = 2, \dots, q$  successively decodes  $\hat{\mathbf{v}}_i$  and reencodes it into  $\hat{\mathbf{b}}_i$  based on  $\mathbf{y}$  and LLRs  $\mathbf{l}_i$ , given all previous bits  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_{i-1}$ . The estimation of the uncoded bits  $\hat{\mathbf{b}}_{np+1}, \dots, \hat{\mathbf{b}}_m$  is obtained after getting  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_q$ .

Fig. 3: Block diagrams of BICM and MLMC for VCs.

computation complexity is reduced a lot since no closest lattice point quantizer needs to be applied to all points in  $\mathcal{B}(\mathbf{y}^j, R^2)$  in order to determine the intersection with  $\Gamma$ .

The bit labels of the points in the Euclidean ball  $\mathbf{e} \in \mathcal{B}(\mathbf{y}^j, R^2)$  can be obtained by

$$f_{\text{BRGC}}^{-1}(w(\mathbf{e}), \mathbf{h}). \quad (33)$$

Then for each bit position  $k$ ,  $\mathcal{B}(\mathbf{y}^j, R^2)$  can be divided into two subsets  $\mathcal{B}(\mathbf{y}^j, R^2) = \mathcal{B}^{(k,0)}(\mathbf{y}^j, R^2) \cup \mathcal{B}^{(k,1)}(\mathbf{y}^j, R^2)$ , containing points within  $\mathcal{B}(\mathbf{y}^j, R^2)$  with 0 and 1 at bit position  $k$ , respectively. Thus, the approximated max-log LLRs  $\mathbf{l}^j$  of the  $j$ th channel realization  $\mathbf{y}^j$  contain  $m$  independent values  $l_k^j$  for  $k = 1, \dots, m$ . The LLR of the  $k$ th bit is computed as

$$l_k^j = -\frac{1}{\sigma^2} \left( \min_{\mathbf{e} \in \mathcal{B}^{(k,0)}(\mathbf{y}^j, R^2)} \|\mathbf{y} + \mathbf{a} - \mathbf{e}\|^2 - \min_{\mathbf{e} \in \mathcal{B}^{(k,1)}(\mathbf{y}^j, R^2)} \|\mathbf{y} + \mathbf{a} - \mathbf{e}\|^2 \right). \quad (34)$$

If either  $\mathcal{B}^{(k,0)}(\mathbf{y}^j, R^2)$  or  $\mathcal{B}^{(k,1)}(\mathbf{y}^j, R^2)$  is empty, then the corresponding minimum in (34) is set to a large default value  $r > R^2$ . Here only integer values of  $R^2$  are considered, because the  $\|\mathbf{e} + \mathbf{a} - \lfloor \mathbf{y}^j + \mathbf{a} \rfloor\|^2$  in (34) is always an integer.

The choice of  $R^2$  involves a trade-off between computation complexity and decoding performance. For high-dimensional VCs, the LLR approximation can have high complexity when the Euclidean ball contains a larger number of points. Given  $R^2$ ,  $r$  can be roughly optimized by testing which value gives the best decoding performance.

### B. MLMC for VCs with SP mapping

Denoting the terms of (??) as  $I_1, \dots, I_m$ , a channel can be regarded as  $m$  virtual independent “equivalent subchannels” with MIs

$$I_k = I(\mathbf{Y}; \mathbf{B}_k | \mathbf{B}_1, \dots, \mathbf{B}_{k-1}) \quad (35)$$

for  $k = 1, \dots, m$ . This concept directly implies an MLMC scheme proposed by Imai *et al.* in [?], where the bit subchannels are protected unequally with different component channel codes and a multistage decoder decodes the bits successively from  $\mathbf{B}_1$  to  $\mathbf{B}_m$  provided that the previous bits are given. Practical design rules of the code rates can be found in [?]. The suitable labeling for MLMC is Ungerboeck’s SP labeling.

Fig. ?? shows an MLMC scheme for VCs, which contains  $q$  component codes with code rates  $R_c^i$  for  $i = 1, \dots, q$  and

the same codeword length  $N$  to protect the first  $np$  bit levels of the VC symbols, and the last  $m - np$  bit levels remain uncoded. Thus, the MLCM for VCs has a total rate of

$$R_{\text{tot}} = \frac{\sum_{i=1}^q k_i R_c^i + (m - np)}{n/2} [\text{bits}/2\text{D-symbol}]. \quad (36)$$

The transmitter forms  $m$  bits  $\mathbf{b}^j = (\mathbf{b}_1^j, \dots, \mathbf{b}_q^j, b_{np+1}^j, \dots, b_m^j)$  for  $j = 1, \dots, N$ , where  $\mathbf{b}_i^j = (b_i^{k_i(j-1)+1}, \dots, b_i^{k_i j})$  are the  $(k_i(j-1)+1)$ th to  $k_i j$ th bits of  $\mathbf{b}_i$  for  $i = 1, \dots, q$ . The  $\mathbf{b}_i$  is illustrated in Fig. ?? with length  $k_i N$  for  $i = 1, \dots, q$ . For the SP mapping, the VC mapper first maps  $\mathbf{b}^j$  to an integer by  $\mathbf{u}^j = f_{\text{SP}}(\mathbf{b}^j)$  and then encodes the integer into a VC point by  $\mathbf{x}^j = g(\mathbf{u}^j)$ .

At the receiver side, after getting  $(\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_q^j)$ , the coset representative  $\hat{\mathbf{c}}_i^j$  is obtained according to  $\mathbf{C}_i$  for  $i = 1, \dots, q$ , and the  $\hat{\mathbf{c}}^j \in [\mathbb{Z}^n/2^p\mathbb{Z}^n]$  is found by  $\hat{\mathbf{c}}^j = \sum_{i=1}^q \hat{\mathbf{c}}_i^j$ . Then the estimation of the transmitted point should be found by searching a point within the subset  $2^p\mathbb{Z}^n + \hat{\mathbf{c}}^j$  that is closest to  $\mathbf{y}^j + \mathbf{a}$ . This is equivalent to

$$\begin{aligned} \hat{\mathbf{x}}^j &= \mathcal{Q}_{2^p\mathbb{Z}^n + \hat{\mathbf{c}}^j}(\mathbf{y}^j + \mathbf{a}) \\ &= 2^p \left\lfloor \frac{\mathbf{y}^j + \mathbf{a} - \hat{\mathbf{c}}^j}{2^p} \right\rfloor + \hat{\mathbf{c}}^j - \mathbf{a}. \end{aligned} \quad (37)$$

Finally, the bit labels of  $\hat{\mathbf{x}}^j$  are obtained by

$$\hat{\mathbf{b}}^j = (\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_q^j, \hat{b}_{np+1}^j, \dots, \hat{b}_m^j) = f_{\text{SP}}^{-1}(w(\hat{\mathbf{x}}^j)), \quad (38)$$

where the last  $m - np$  bits  $(\hat{b}_{np+1}^j, \dots, \hat{b}_m^j)$  are the estimation of the uncoded bits mapped to  $\hat{\mathbf{x}}^j$ .

The max-log LLRs of  $\mathbf{b}_i^j$  contains  $k_i$  LLR values independent of each other, denoted by  $\mathbf{l}_i^j = (l_{i,1}^j, \dots, l_{i,k_i}^j)$ , which is computed by the following procedure. Given  $\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_{i-1}^j$ , the coset representatives  $\hat{\mathbf{c}}_1^j, \dots, \hat{\mathbf{c}}_{i-1}^j$  are directly obtained according to look-up tables  $\mathbf{C}_1, \dots, \mathbf{C}_{i-1}$ . Then we know that the corresponding integer belongs to the lattice  $\Lambda^{i-1} + \sum_{t=1}^{i-1} \hat{\mathbf{c}}_t^j$ . In the  $i$ th partition step, the coset representatives  $[\Lambda^{i-1}/\Lambda^i]$  have been labeled by the look-up table  $\mathbf{C}_i$ . Then we can divide  $\mathbf{C}_i$  into two subsets  $\mathbf{C}_i = \mathbf{C}_i^{(e,0)} \cup \mathbf{C}_i^{(e,1)}$ , representing coset representatives having a bit 0 and 1 at the  $e$ th bit of  $\mathbf{b}_i$ , respectively. For all  $\hat{\mathbf{c}}_i^j \in \mathbf{C}_i^{(e,0)}$ , we find the closest point to  $\mathbf{y} + \mathbf{a}$  from the lattice  $\Lambda^i + \sum_{t=1}^{i-1} \hat{\mathbf{c}}_t^j + \hat{\mathbf{c}}_i^j$ , and denote all such closest points as the set

$$\mathcal{Z}_i^{(e,0)} = \{\mathbf{z} = \mathcal{Q}_{\Lambda^i + \sum_{t=1}^{i-1} \hat{\mathbf{c}}_t^j}(\mathbf{y} + \mathbf{a}) : \hat{\mathbf{c}}_i^j \in \mathbf{C}_i^{(e,0)}\}. \quad (39)$$

The set  $\mathcal{Z}_i^{(k,1)}$  is defined analogously. Then the max-log LLR of the  $e$ th bit of  $\mathbf{b}_i^j$  can be approximated as

$$\begin{aligned} \text{LLR}(\mathbf{b}_i^j | \mathbf{y}, \hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_{i-1}^j) &\approx l_{i,e}^j \\ &= -\frac{1}{\sigma^2} \left( \min_{\mathbf{z} \in \mathcal{Z}_i^{(e,0)}} \|\mathbf{y} + \mathbf{a} - \mathbf{z}\|^2 - \min_{\mathbf{z} \in \mathcal{Z}_i^{(e,1)}} \|\mathbf{y} + \mathbf{a} - \mathbf{z}\|^2 \right). \end{aligned} \quad (40)$$

The computation complexity of the LLRs in MLCM depends on the partition orders  $k_i$ , which is much lower than the complexity of (??) in BICM. However, MLCM uses  $q$  component codes, which adds complexity and delay compared with BICM.

TABLE III: The considered VCs and TDHQ formats in the simulation.

Name	$n$	$\Lambda/\Lambda_s$	$M$	$m$	$\beta$
$E_8^{24}$	8	$\mathbb{Z}^8/8E_8$	16, 777, 216	24	6
$\Lambda_{24}^{72}$	24	$\mathbb{Z}^{24}/2\Lambda_{24}\mathbf{R}_{24}$	$\approx 4.7 \times 10^{21}$	72	6
$E_8^{32}$	8	$\mathbb{Z}^8/16E_8$	$\approx 4.3 \times 10^9$	32	8
$\Lambda_{24}^{96}$	24	$\mathbb{Z}^{24}/4\Lambda_{24}\mathbf{R}_{24}$	$\approx 7.9 \times 10^{28}$	96	8
$E_8^{40}$	8	$\mathbb{Z}^8/32E_8$	$\approx 1.1 \times 10^{12}$	40	10
$\Lambda_{24}^{120}$	24	$\mathbb{Z}^{24}/8\Lambda_{24}\mathbf{R}_{24}$	$\approx 1.3 \times 10^{36}$	120	10
$E_8^{48}$	8	$\mathbb{Z}^8/64E_8$	$\approx 2.8 \times 10^{14}$	48	12
$\Lambda_{24}^{144}$	24	$16\mathbb{Z}^{24}/4\Lambda_{24}\mathbf{R}_{24}$	$\approx 2.2 \times 10^{43}$	144	12
$\Lambda_{16}^{76}$	16	$\mathbb{Z}^{16}/16\Lambda_{16}$	$\approx 7.9 \times 10^{22}$	76	9.5
$\Lambda_{16}^{92}$	16	$\mathbb{Z}^{16}/32\Lambda_{16}$	$\approx 5.0 \times 10^{27}$	92	11.5
Name	$t_1, t_2$	$M_1, M_2$	$m_{\text{QAM}}$	$\beta_{\text{QAM}}$	
TDHQ1	4, 4	512, 1024	76	9.5	
TDHQ1	4, 4	2048, 1096	92	11.5	

### C. MLCM for VCs with hybrid mapping

The MLCM scheme for VCs with the hybrid mapping in Section ?? is a special case of Fig. ?? with  $p = p_q$  and  $k_i = n(p_i - p_{i-1})$ . At time step  $j = 1, \dots, N$ , the VC mapper maps  $m$  bits  $\mathbf{b}^j = (\mathbf{b}_1^j, \dots, \mathbf{b}_q^j, b_{np+1}^j, \dots, b_m^j)$  to an integer  $\mathbf{u}^j = f_{\text{H}}(\mathbf{b}^j)$  and then maps  $\mathbf{u}^j$  to a VC point  $\mathbf{x}^j = g(\mathbf{u}^j)$ . At the receiver side, successive decoding is performed based on  $\mathbf{y}^j$  and all previous bits  $(\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_{i-1}^j)$  for decoder  $i$ . After decoding  $(\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_q^j)$ , the coset representative  $\hat{\mathbf{c}}_i^j$  is obtained by (??) for  $i = 1, \dots, q$ . The estimation of the coset representative of the partition  $\mathbb{Z}^n/2^p\mathbb{Z}^n$  is calculated as  $\hat{\mathbf{s}}^j = \sum_{i=1}^q \hat{\mathbf{c}}_i^j$ . The estimation of the transmitted VC point  $\hat{\mathbf{x}}^j$  is decoded by (??), where  $\hat{\mathbf{c}}^j$  is replaced by  $\hat{\mathbf{s}}^j$ . Finally, the estimation of bit labels of  $\hat{\mathbf{x}}^j$  is obtained by

$$\hat{\mathbf{b}}^j = (\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_q^j, \hat{b}_{np+1}^j, \dots, \hat{b}_m^j) = f_{\text{H}}^{-1}(w(\hat{\mathbf{x}}^j)), \quad (41)$$

where the last  $m - np$  bits  $(\hat{b}_{np+1}^j, \dots, \hat{b}_m^j)$  are the estimation of the uncoded bits mapped to  $\hat{\mathbf{x}}^j$ . The hybrid CM scheme for VCs has a total rate of

$$R_{\text{tot}} = \frac{\sum_{i=1}^q n(p_i - p_{i-1})R_c^i + (m - np)}{n/2} [\text{bits}/2\text{D-symbol}]. \quad (42)$$

The max-log LLRs of  $\mathbf{b}_i^j = (b_{i,1}^j, \dots, b_{i,n(p_i - p_{i-1})}^j)$  contain  $n(p_i - p_{i-1})$  independent LLR values for  $i = 1, \dots, q$ , denoted by  $\mathbf{l}_i^j = (l_{i,1}^j, \dots, l_{i,n(p_i - p_{i-1})}^j)$  and calculated as follows. Given the previous estimated bits  $\hat{\mathbf{b}}_1^j, \dots, \hat{\mathbf{b}}_{i-1}^j$ , the coset representatives  $\mathbf{c}_t$  for  $t = 1, \dots, i-1$  are obtained by (??). If  $|2^{p_{i-1}}\mathbb{Z}^n/2^{p_i}\mathbb{Z}^n|$  is not a very large number, the max-log LLR of the  $e$ th bit of  $\mathbf{b}_i^j$ , denoted by  $l_{i,e}^j$ , can be calculated using (??) with  $\Lambda^{i-1} = 2^{p_{i-1}}\mathbb{Z}^n$  and  $\Lambda^i = 2^{p_i}\mathbb{Z}^n$ . If  $|2^{p_{i-1}}\mathbb{Z}^n/2^{p_i}\mathbb{Z}^n|$  is large, then  $l_{i,e}^j$  can be calculated by enumerating a scaled Euclidean ball centered at the closest



TABLE IV: The parameters of the considered CM schemes in simulation.

Constellation Mapping	$\beta$	CM	Partition chain	#LDPC codes	Code rates	#Coded bit levels/ $m$	$R_{\text{tot}}$ [2D-symbol]	
$E_8^{24}$	Gray	6	BICM	-	1	$R_c = 8/9$	24/24	5.33
64-QAM	Gray	6	BICM	-	1	$R_c = 8/9$	6/6	5.33
$E_8^{24}$	hybrid	6	MLCM	$\mathbb{Z}^8/2\mathbb{Z}^8/E_8^{24}$	1	$R_c^1 = 2/3$	8/16	5.33
$\Lambda_{24}^{72}$	hybrid	6	MLCM	$\mathbb{Z}^{24}/2\mathbb{Z}^{24}/\Lambda_{24}^{72}$	1	$R_c^1 = 2/3$	24/48	5.33
64-QAM	hybrid	6	MLCM	-	1	$R_c^1 = 2/3$	2/6	5.33
$E_8^{32}$	Gray	8	BICM	-	1	$R_c = 9/10$	32/32	7.2
256-QAM	Gray	8	BICM	-	1	$R_c = 9/10$	8/8	7.2
$E_8^{32}$	hybrid	8	MLCM	$\mathbb{Z}^8/2\mathbb{Z}^8/E_8^{32}$	1	$R_c^1 = 3/5$	8/24	7.2
$\Lambda_{24}^{96}$	hybrid	8	MLCM	$\mathbb{Z}^{24}/2\mathbb{Z}^{24}/\Lambda_{24}^{96}$	1	$R_c^1 = 3/5$	24/72	7.2
256-QAM	hybrid	8	MLCM	-	1	$R_c^1 = 3/5$	2/8	7.2
256-QAM	SP	8	MLCM	-	2	$R_c^1 = 1/3, R_c^2 = 8/9$	2/8	7.22
$E_8^{40}$	Gray	10	BICM	-	1	$R_c = 9/10$	40/40	9
1024-QAM	Gray	10	BICM	-	1	$R_c = 9/10$	10/10	9
$E_8^{40}$	hybrid	10	MLCM	$\mathbb{Z}^8/2\mathbb{Z}^8/E_8^{40}$	1	$R_c^1 = 1/2$	8/40	9
$\Lambda_{24}^{120}$	hybrid	10	MLCM	$\mathbb{Z}^{24}/2\mathbb{Z}^{24}/\Lambda_{24}^{120}$	1	$R_c^1 = 1/2$	24/120	9
1024-QAM	hybrid	10	MLCM	-	1	$R_c^1 = 1/2$	2/10	9
$E_8^{48}$	Gray	12	BICM	-	1	$R_c = 9/10$	48/48	10.8
4096-QAM	Gray	12	BICM	-	1	$R_c = 9/10$	12/12	10.8
$E_8^{48}$	hybrid	12	MLCM	$\mathbb{Z}^8/2\mathbb{Z}^8/E_8^{48}$	1	$R_c^1 = 2/5$	8/48	10.8
$\Lambda_{24}^{144}$	hybrid	12	MLCM	$\mathbb{Z}^{24}/2\mathbb{Z}^{24}/\Lambda_{24}^{144}$	1	$R_c^1 = 2/5$	24/144	10.8
4096-QAM	hybrid	12	MLCM	-	1	$R_c^1 = 2/5$	2/12	10.8
$E_8^{48}$	SP	12	MLCM	$\mathbb{Z}^8/D_8/E_8\mathbf{R}_8/2\mathbb{Z}^8/E_8^{48}$	1	$R_c^1 = 0, R_c^2 = 0, R_c^3 = 4/5$	8/48	10.8
$\Lambda_{16}^{92}$	Gray	11.5	BICM	-	1	$R_c = 9/10$	92/92	10.35
TDHQ2	Gray	11.5	BICM	-	1	$R_c = 9/10$	92/92	10.35
$\Lambda_{16}^{92}$	hybrid	11.5	MLCM	$\mathbb{Z}^{16}/2\mathbb{Z}^{16}/\Lambda_{16}^{92}$	1	$R_c^1 = 2/5$	16/92	10.3
TDHQ2	hybrid	11.5	MLCM	-	1	$R_c^1 = 2/5$	16/92	10.3

lattice point of  $2^{p_i-1}\mathbb{Z}^n$  to  $\mathbf{y}$ , i.e.,

$$\mathcal{D}(\mathbf{y}, R^2) \triangleq \left\{ \mathbf{e} : \|\mathbf{e} + \mathbf{a} - \left\lfloor \frac{\mathbf{y} + \mathbf{a}}{2^{p_i-1}} \right\rfloor \cdot 2^{p_i-1}\|^2 \leq 2^{2p_i-1} R^2, \right. \\ \left. \mathbf{e} + \mathbf{a} \in 2^{p_i-1}\mathbb{Z}^n \right\}, \quad (43)$$

which consists of two subsets  $\mathcal{D}(\mathbf{y}, R^2) = \mathcal{D}(\mathbf{y}, R^2)^{(e,0)} \cup \mathcal{D}(\mathbf{y}, R^2)^{(e,1)}$ , representing points with 0 and 1 at the  $e$ th bit of  $\mathbf{b}_i$ , respectively. Then  $l_{i,e}^j$  is computed as

$$l_{i,e}^j = -\frac{1}{\sigma^2} \left( \min_{\mathbf{e} \in \mathcal{D}^{(e,0)}(\mathbf{y}^j, R^2)} \|\mathbf{y} + \mathbf{a} - \mathbf{e}\|^2 \right. \\ \left. - \min_{\mathbf{e} \in \mathcal{D}^{(e,1)}(\mathbf{y}^j, R^2)} \|\mathbf{y} + \mathbf{a} - \mathbf{e}\|^2 \right). \quad (44)$$

It is worth noting that, when  $p_i = i$  for  $i = 1, \dots, q$  (i.e., the partition chain  $\mathbb{Z}^n/2\mathbb{Z}^n/\dots/2^q\mathbb{Z}^n/\Lambda_s$  is considered), setting  $R^2 = 1$  in (??) is sufficient, thanks to (??) and (??) in the hybrid labeling. The approximation complexity will be very low since  $\mathcal{D}(\mathbf{y}, 1)$  contains only  $2n+1$  points. Also,  $\mathcal{D}^{(e,0)}(\mathbf{y}, 1)$  or  $\mathcal{D}^{(e,1)}(\mathbf{y}, 1)$  can never be an empty set for  $i = 1, \dots, n$ , due to (??) and (??) again.

## V. PERFORMANCE ANALYSIS

In this section, we present the coded BER performance in the AWGN channel for VCs with the three proposed CM schemes introduced in Section ??, and compare them with the most commonly used benchmark at the same rate: Gray-labeled QAM with BICM [?], [?], [?], [?]. In order to see

how much gain is actually from shaping, we also apply the proposed hybrid mapping in Section ?? to QAM and combine it with MLCM. This scheme is new, but other types of hybrid mapping for QAM with MLCM exist in the literature [?], [?]. However, optimizing the design of MLCM for QAM in concatenated CM schemes is not the focus of this paper.

For fairness of comparison,  $n/2$  2D QAM formats are multiplexed in the time domain to fill the same number of dimensions and to achieve the same uncoded spectral efficiencies as VCs. The traditional way of realizing non-integer spectral efficiencies for QAM formats is through time-domain hybrid QAM (TDHQ) [?], [?], [?], [?]. To form an  $n$ -dimensional TDHQ format,  $t_1$  and  $t_2$  2D QAM formats with cardinalities  $M_1$  and  $M_2$ , respectively, are used, satisfying

$$t_1 + t_2 = \frac{n}{2} \\ \beta_{\text{QAM}} = \beta = \frac{t_1 \log_2(M_1) + t_2 \log_2(M_2)}{t_1 + t_2}.$$

For example, one TDHQ symbol  $\mathbf{x}$  having the same spectral efficiency as  $\Lambda_{16}^{92}$  ( $\beta = 11.5$  [bits/2D-symbol]) consists of  $t_1 = 4$  4096-QAM and  $t_2 = 4$  2048-QAM symbols:

$$\mathbf{x} = (\underbrace{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4}_{\in 4096\text{-QAM}}, \underbrace{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8}_{\in 2048\text{-QAM}}).$$

The two constituent QAM constellations are scaled to the same minimum distance, which maximizes the minimum distance of the resulting hybrid QAM constellation for a given  $n$ -dimensional symbol energy  $E_s$  [?, Ch. 4.3].

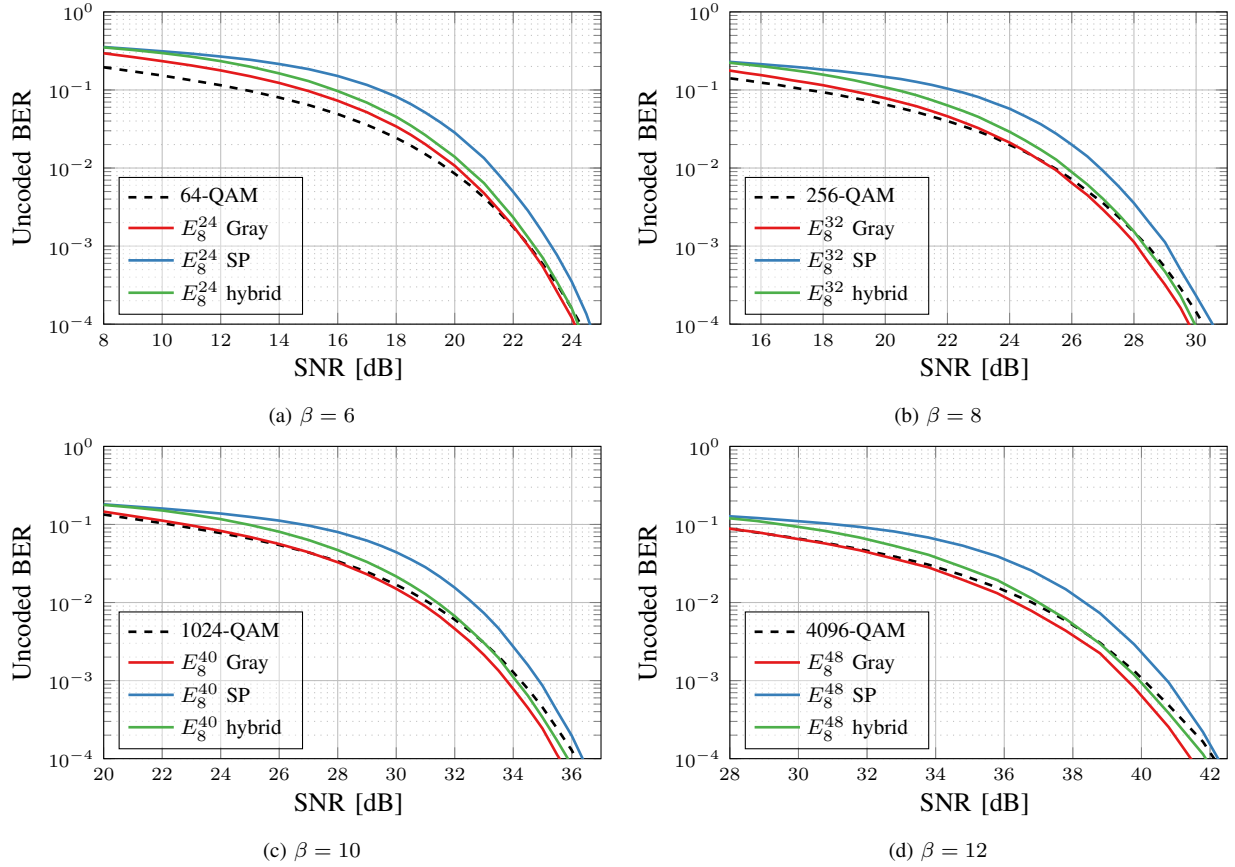


Fig. 4: Uncoded BER performance of 8D VCs compared with QAM at the same spectral efficiency.

VCs show high uncoded BER gains at high dimensions and spectral efficiencies [?, Fig. 5]. Thus, we investigate the performance of 8D, 16D, and 24D VCs with high spectral efficiencies of up to 12 bits/2D-symbol. The parameters of the considered VCs and the benchmark QAM formats are listed in Table ??.

Fig. ?? shows the uncoded BER for 8D VCs with three different mapping rules, compared with Gray-labeled QAM. For VCs in uncoded systems, the Gray mapping has the lowest uncoded BER among the three mappings and achieves an increasing SNR gain over QAM as  $\beta$  increases, which implies that VCs with Gray mapping can outperform QAM in systems with a single HD FEC code [?, Fig. 5]. The hybrid mapping has marginal SNR gains over QAM at high SNRs, since the penalty of a non-Gray mapping for the VC almost counteracts its shaping gains. The SP mapping yields the worst performance and shows no gain over Gray-labeled QAM due to not efficient labeling.

The performance of 8D and 24D VCs compared with QAM constellations with both BICM and MLCM in coded systems is shown in Fig. ?. A set of LDPC codes from the digital video broadcasting (DVB-S2) standard [?] with multiple code rates is considered as the inner code. The codeword length is  $N = 64800$  and 50 decoding iterations are used. Table ?? lists the parameters of the considered CM schemes in this paper. For all the VCs with hybrid mapping listed in Table ??,  $p = q = 1$  and  $k_1 = n$ . If we target a BER of  $1.81 \times 10^{-3}$

when a zipper code [?] is used as the outer code, 8D VCs with MLCM and hybrid mapping yield an increasing SNR gain over QAM with MLCM and hybrid mapping from 0.22 to 0.59 dB as  $\beta$  increases. These gains mainly come from shaping. When compared with Gray-labeled QAM with BICM, the most commonly used benchmark,  $0.22 + 0.18 = 0.4$  to  $0.59 + 0.27 + 0.40 = 1.26$  dB SNR gains are achieved by 8D VCs with MLCM and hybrid mapping. The MLCM achieves SNR gains over BICM due to its effective utilization of FEC overheads to protect the most significant bit levels. However, MLCM has a high error floor at a BER around  $10^{-3}$  due to the uncoded bit levels, whereas the BICM scheme does not, as all bit levels are protected by FEC codes.

Fig. ?? also shows that 8D VCs with BICM do not outperform QAM with BICM at  $\beta = 6$  and  $\beta = 8$ , and start to achieve 0.19 and 0.40 dB SNR gains at  $\beta = 10$  and  $\beta = 12$ , respectively. This observation is consistent with [?, Fig. 6, Fig. 9] that the GMI performance of VCs outperform QAM only at high spectral efficiencies.

In Fig. ??, we also show the BER performance of 256-QAM with MLCM and SP mapping. In this scheme, the code rates are selected according to the “capacity rule” from [?, Sec. IV-A], i.e., the MIs of the equivalent subchannels  $I_i$  defined in (?). As the first 2 bit levels are protected and the left 6 bit levels are uncoded, the MIs of the 8 equivalent subchannels

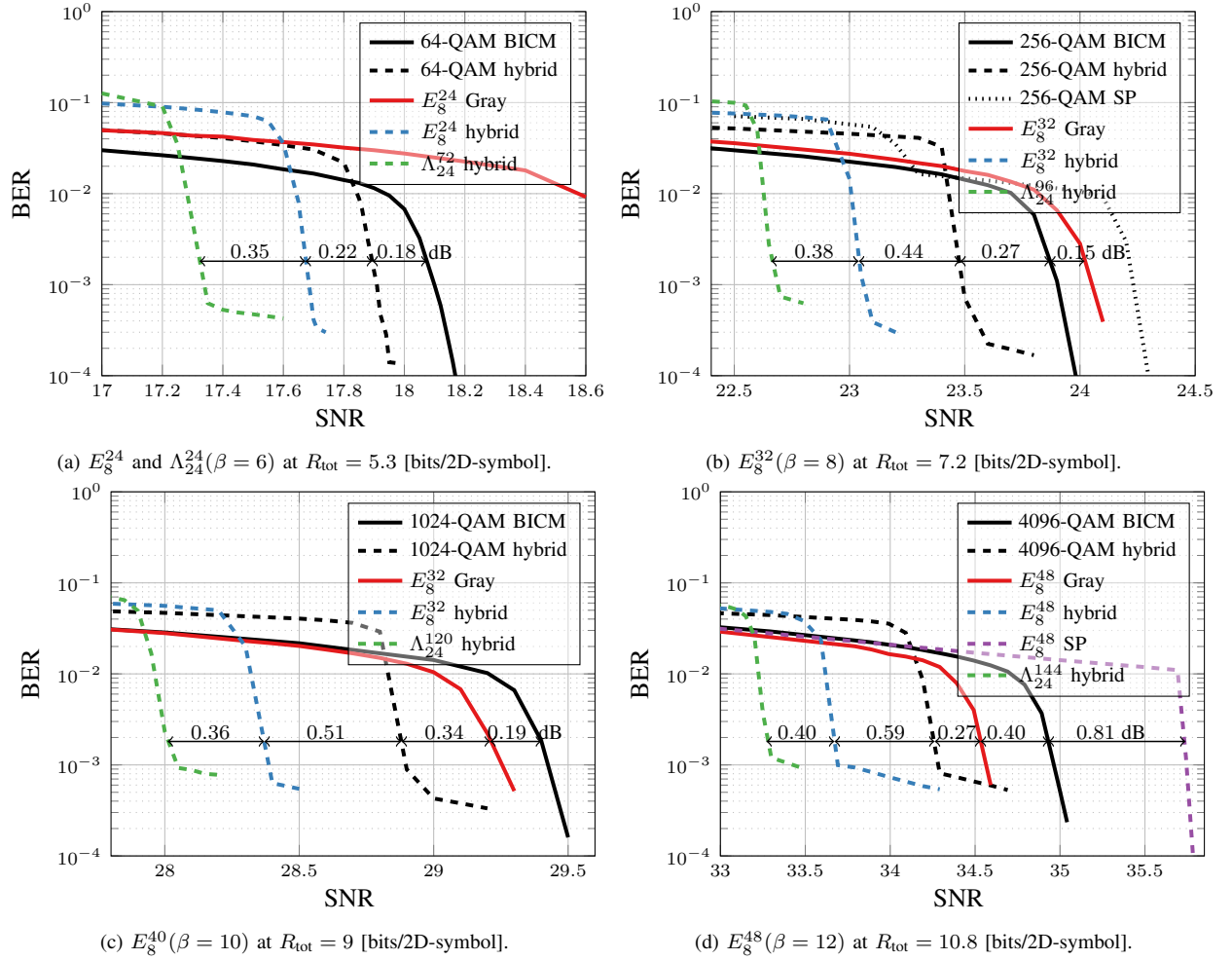


Fig. 5: Coded BER performance of 8D and 24D VCs compared with QAM at the same total rate. Solid lines represent BICM performance and dashed lines represent MLCM performance with hybrid mapping. The LLRs of VCs with BICM are calculated using (??) with  $R^2 = 6$  and  $r = 20$ ; the LLRs of VCs with MLCM and hybrid mapping are calculated using (??) with  $R^2 = 1$ ; the LLRs of VCs with MLCM and SP mapping are calculated using (??).

are lower-bounded as

$$\begin{aligned}
 I(Y; X) &= I(Y; B_1, \dots, B_8) \\
 &\geq I(Y; B_1) + I(Y; B_2|B_1) + \sum_{k=3}^8 I(Y; B_k|B_1, B_2). \quad (45)
 \end{aligned}$$

We define the eight conditional MI values of each bit level  $I_k$  for  $k = 1, \dots, 8$  as the eight terms in (??) (6 terms are in the summation) and Fig. ?? shows  $I_k$  for  $k = 1, \dots, 8$ . To target a total rate of 7.2 [bits/2D-symbol], the MIs at SNR = 23.9 dB in Fig. ?? suggests  $R_c^1 = 1/3$  and  $R_c^2 = 8/9$  approximately according to the capacity rule. The 256-QAM with MLCM and SP mapping suffers a large performance loss compared with other schemes, although two component codes are used.

In Fig. ??, we show the BER performance of  $E_8^{48}$  with MLCM and SP mapping. The partition chain is  $\mathbb{Z}^8/D_8/E_8\mathbf{R}_8/2\mathbb{Z}^8/E_8^{48}$  with parameters  $p = 1, q = 3, k_1 = 1, k_2 = 3$  and  $k_3 = 4$ . A bit different from (??) and [?], since we can have multiple bits per partition level, the bits at the same partition level are considered independent of each other,

and protected by the same code. Thus, the MIs of the first 8 equivalent subchannels are lower-bounded as

$$\begin{aligned}
 I(Y; B_1, \dots, B_8) &\geq I(Y; B_1) + I(Y; B_2, B_3, B_4|B_1) \\
 &\quad + I(Y; B_5, B_6, B_7, B_8|B_1, B_2, B_3, B_4) \quad (46) \\
 &\geq I(Y; B_1) + \sum_{k=2}^4 I(Y; B_k|B_1) + \sum_{k=5}^8 I(Y; B_k|B_1, \dots, B_4) \quad (47)
 \end{aligned}$$

The eight conditional MI values of each bit level  $I_k$  for  $k = 1, \dots, 8$  are defined as the eight terms in (??). Fig. ?? shows the estimated  $I_i$  for  $i = 1, \dots, 8$  using the method based on importance sampling proposed in [?], [?]. Bits at the same partition level are protected by the same code. Thus, three different code rates should be used for the lattice partition  $\mathbb{Z}^8/D_8/E_8\mathbf{R}_8/2\mathbb{Z}^8$ . For a bit level with an estimated conditional MI lower than 0.2, we do not use that subchannel to carry information, and set the code rate to 0. If we look at the MIs at SNR = 24.2 dB in Fig. ??, the first four bits are not used to transmit information with  $R_c^1 = R_c^2 = 0$ , and  $R_c^3 = 4/5$  according to the capacity rule. From Fig. ??, a 0.81

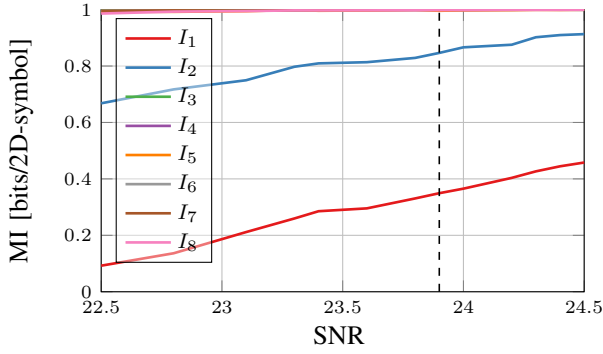


Fig. 6: The MIs of all 8 bit levels for 256-QAM with the SP mapping. The  $I_k$  for  $k = 3, \dots, 8$  overlap at approximately 1 bit/2D-symbol for all SNRs.

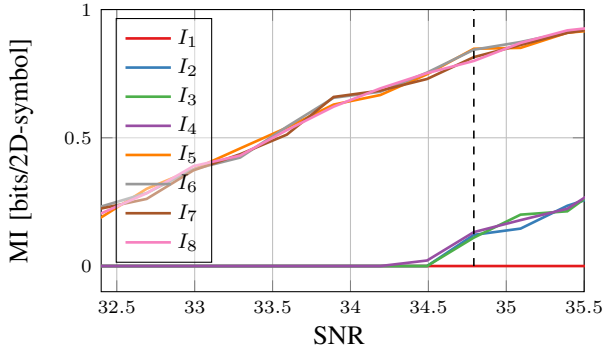


Fig. 7: The MIs of the first 8 bit levels for  $E_8^{48}$  with the SP mapping.

dB SNR loss is observed for  $E_8^{48}$  with MLCM and SP mapping compared with 4096-QAM with BICM. This is due to the bad uncoded BER performance for VCs with SP mapping resulting from the high penalty of non-Gray labeling, and the FEC code cannot sufficiently reduce such a high uncoded BER. Thus, we do not consider SP mapping for 16D VCs in the following results.

Among the 8D results, VCs with MLCM and hybrid mapping always yield the best performance. In Fig. ??, we also illustrate the performance of 24D VCs with MLCM and hybrid mapping. It shows that 24D VCs can achieve 0.57 to 0.99 dB gains over QAM with MLCM and hybrid mapping at different  $\beta$ . When compared with QAM with BICM, up to 0.75–1.66 dB gains are achieved by 24D VCs. Larger SNR gains over QAM formats are observed than in the 8D case, since 24D VCs inherently have a higher asymptotic shaping gain than 8D VCs [?, Table I].

For 16D VCs, which have noninteger spectral efficiencies, Fig. ?? presents the uncoded and coded BER performance compared with TDHQ formats. In Fig. ??,  $\Lambda_{16}^{92}$  with MLCM and hybrid mapping achieves 0.99 dB SNR gain over TDHQ2 with MLCM and hybrid mapping. In total, it achieves up to  $0.99 + 0.19 + 0.66 = 1.84$  dB SNR gain over TDHQ2 with BICM.

VCs with MLCM and hybrid mapping have the lowest complexity, as the scheme needs only one FEC code and only some of the bit levels are encoded. Moreover, the computation of LLRs is the fastest. In general, VCs with MLCM and

SP mapping use more than one component code, which increases the complexity, and estimating the MIs of equivalent subchannels has a high complexity. VCs with BICM also use just one component code and has good performance gains at high  $\beta$  thanks to the small loss in the LLR approximation. However, its LLR approximation complexity is higher than for the two MLCM schemes for VCs.

## VI. CONCLUSION

In this paper, we propose three CM schemes for very large MD VCs, including bit-to-integer mapping algorithms and LLR computation algorithms. This makes very large VCs adoptable in practical communication systems with SD FEC codes. Among them, one MLCM scheme for VCs with hybrid mapping has even lower complexity than BICM. The simulation results for the AWGN channel show that even with some penalty from the non-Gray labeling, VCs achieve high shaping gains over QAM with both BICM and MLCM, especially at high spectral efficiencies. Higher power gains are expected for nonlinear fiber channels, which remains as future work.

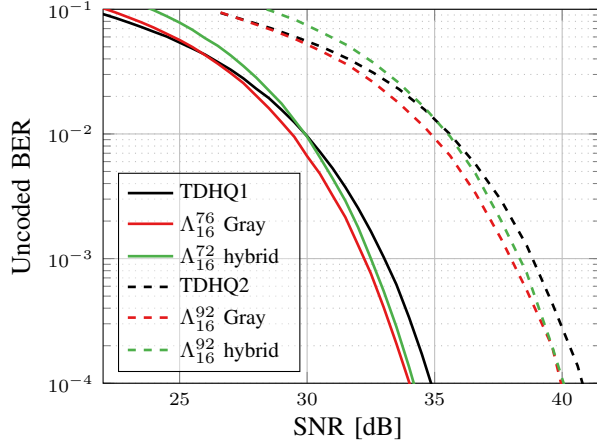
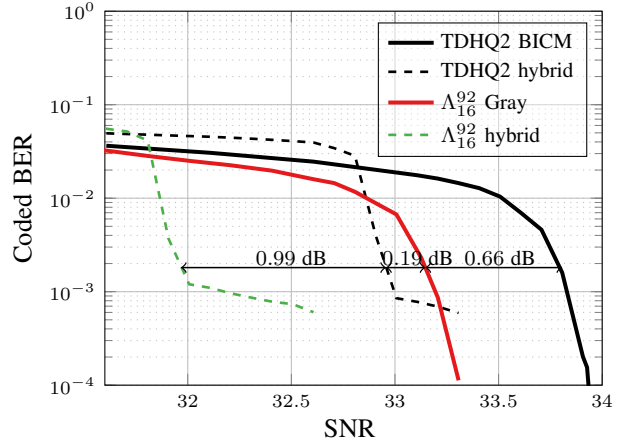
(a)  $\Lambda_{16}^{76}$  ( $\beta = 9.5$ ) and  $\Lambda_{16}^{92}$  ( $\beta = 11.5$ ).(b)  $\Lambda_{16}^{92}$  ( $\beta = 11.5$ ) at  $R_{\text{tot}} = 10.3$  [bits/2D-symbol].

Fig. 8: Uncoded and coded BER performance of 16D VCs compared with TDHQ formats. Solid lines represent BICM performance and dashed lines represent MLCM performance. The LLRs of VCs with BICM are calculated using (??) with  $R^2 = 2$  and  $r = 20$ ; the LLRs of VCs with MLCM and hybrid mapping are calculated using (??) with  $R^2 = 1$ .