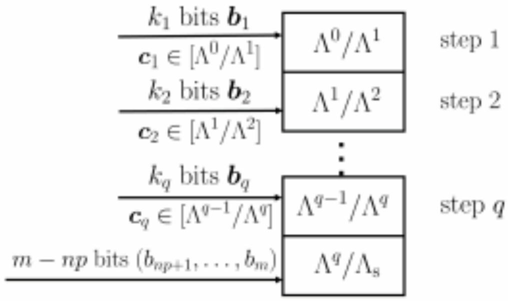


Fig. 1: Illustration of the labeling of a partition chain $\Lambda^0/\Lambda^1/\dots/\Lambda^{q-1}/\Lambda^q/\Lambda_s$.

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

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

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

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 $\boldsymbol{b} \in \{0,1\}^m$ is divided into n blocks according to $\boldsymbol{h}$,

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

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 $\boldsymbol{b}_i$ is converted to an integer u_i using the binary reflected Gray code (BRGC) [?] for $i = 1,\dots,n$, yielding

$$\boldsymbol{u} = (u_1, \dots, u_n). \tag{13}$$

The above procedures converting $\boldsymbol{b}$ to $\boldsymbol{u}$ according to the BRGC is denoted by $f_{\text{BRGC}}(\boldsymbol{b},\boldsymbol{h})$, and the inverse process of converting an integer vector $\boldsymbol{u}$ to a binary vector $\boldsymbol{b}$ is denoted by $f^{-1}_{\text{BRGC}}(\boldsymbol{u},\boldsymbol{h})$ 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

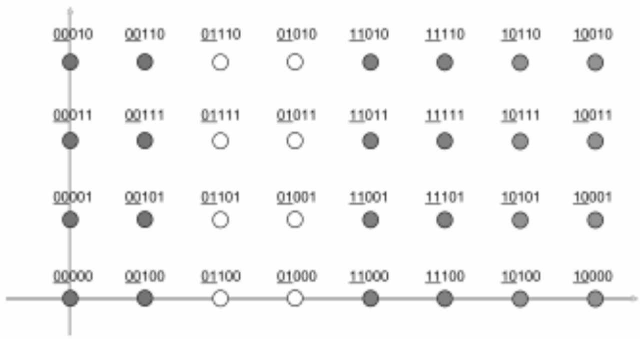
Ungerboeck's SP concept [?] maps binary labels to 1D or 2D constellation points by successively partitioning the constellation into two subsets at each bit level in order to maximize the intra-set minimum squared Euclidean distance (MSED) at each level, so that unequal error protection can be implemented 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 $\boldsymbol{b} \in \{0,1\}^m$ to integers $\boldsymbol{u} \in \mathcal{U}$ 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.

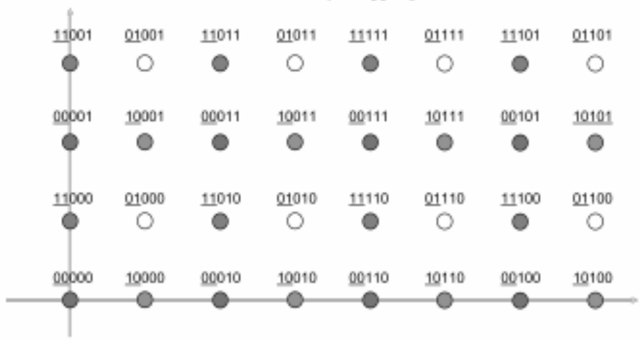
The preprocessing of the proposed SP mapping works as follows. First, q intermediate lattices $\Lambda^1,\dots,\Lambda^q$ are found to form the partition chain

$$\Lambda^0/\Lambda^1/\dots/\Lambda^q/\Lambda_s, \tag{14}$$

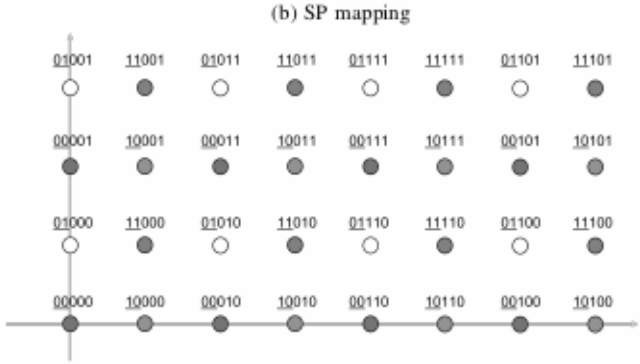
where $\Lambda^0 = \mathbb{Z}^n$ and $\Lambda^q = 2^p\mathbb{Z}^n$ is where to stop the partition. The partition chain should satisfy $\Lambda^0 \supset \Lambda^1 \supset \dots \supset \Lambda^q \supseteq \Lambda_s$ and have increasing MSEDs of $d_i^2 = 2^i$ for $i = 1,\dots,q$. The order of each partition 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 $\boldsymbol{C}_i$. 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



(a) Gray mapping



(b) SP mapping



(c) Hybrid mapping

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 $\mathbb{Z}^n/\Lambda_\mathrm{s}$, 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. 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 $\boldsymbol{X}$ labeled by m bits $(\boldsymbol{B}_1, \dots, \boldsymbol{B}_m)$ and output symbols $\boldsymbol{Y}$. The mutual information (MI) between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$\begin{aligned} I(\boldsymbol{Y};\boldsymbol{X}) &= I(\boldsymbol{Y};\boldsymbol{B}_1,\dots,\boldsymbol{B}_m) \\ &= I(\boldsymbol{Y};\boldsymbol{B}_1) + I(\boldsymbol{Y};\boldsymbol{B}_2|\boldsymbol{B}_1) + \dots \\ &\quad + I(\boldsymbol{Y};\boldsymbol{B}_m|\boldsymbol{B}_1,\dots,\boldsymbol{B}_{m-1}) \end{aligned} \tag{29}$$

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

$$I_{\mathrm{BMI}} = \sum_{i=1}^{m} I(\boldsymbol{Y};\boldsymbol{B}_i) \le I(\boldsymbol{Y};\boldsymbol{X}) \tag{30}$$

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 βR_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.

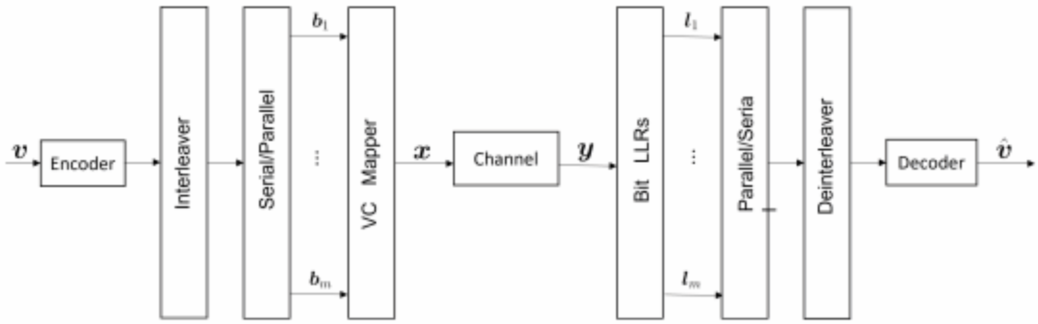
For a constellation Γ transmitted over the AWGN channel, the max-log approximation [**?**] of the kth bit after receiving a $\boldsymbol{y}^j \in \mathbb{R}^n$ for $j = 1, \dots, N/m$ is defined as

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

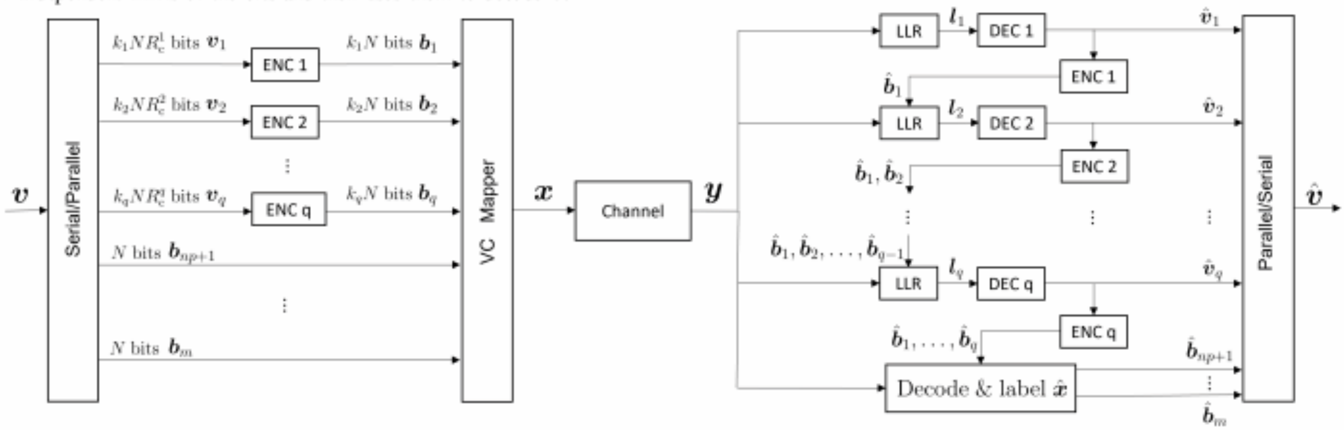
where σ^2 is the noise power per two dimensions, $\Gamma^{(k,0)}$ and $\Gamma^{(k,1)}$ are the sets of constellation points with 0 and 1 at bit position k, respectively, and $\Gamma^{(k,0)} \cup \Gamma^{(k,1)} = \Gamma$. Computing (**??**) needs a full search in Γ, which is infeasible for very large constellations. In [**?**], an LLR approximation method based on importance sampling is proposed and exemplified for very large VCs based on the lattice partition $\mathbb{Z}^n/\Lambda_\mathrm{s}$ 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 $\mathbb{Z}^n - \boldsymbol{a}$ that are inside a "Euclidean ball" centered at $\lfloor \boldsymbol{y}^j + \boldsymbol{a} \rceil$, i.e.,

$$\mathcal{B}(\boldsymbol{y}^j, R^2) \triangleq \{\boldsymbol{e} : \|\boldsymbol{e} + \boldsymbol{a} - \lfloor \boldsymbol{y}^j + \boldsymbol{a} \rceil\|^2 \le R^2, \boldsymbol{e} + \boldsymbol{a} \in \mathbb{Z}^n\}, \tag{32}$$

where $\lfloor\cdot\rceil$ represents rounding a vector to its nearest integer vector and $R^2 \ge 0$ 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 $\mathcal{B}(\boldsymbol{y}^j, R^2) \cap \Gamma$, searching among all points from $\mathcal{B}(\boldsymbol{y}^j, R^2)$ might return a point outside Γ, which causes a loss in decoding performance. However, the



(a) BICM: A block of NR_c information bits $\boldsymbol{v}$ 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 $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ of length N/m. At the time slot $j = 1, \ldots, N/m$, a VC mapper first maps m bits $\boldsymbol{b}^j = (b_1^j, \ldots, b_m^j)$ to an integer $\boldsymbol{u}^j$ by $\boldsymbol{u}^j = f_{\text{BRGC}}(\boldsymbol{b}^j, \boldsymbol{h})$, and then maps $\boldsymbol{u}^j$ to a VC point $\boldsymbol{x}^j \in \Gamma$ by $\boldsymbol{x}^j = g(\boldsymbol{u}^j)$. The receiver deinterleaves N independent LLRs of the bits and then uses them to decode $\hat{\boldsymbol{v}}$.



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

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

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

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

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

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

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

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

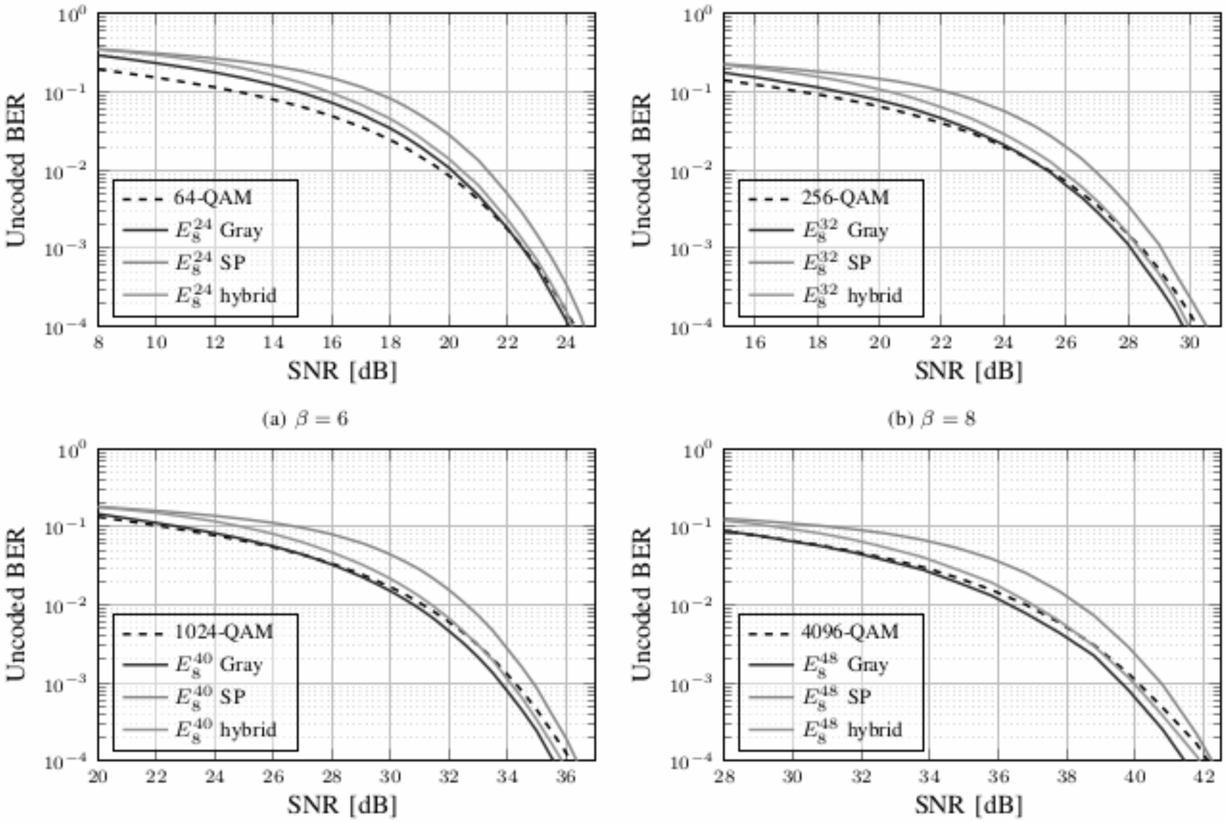
B. *MLCM for VCs with SP mapping*

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

$$I_k = I(\boldsymbol{Y}; \boldsymbol{B}_k | \boldsymbol{B}_1, \ldots, \boldsymbol{B}_{k-1}) \tag{35}$$

for $k = 1, \ldots, m$. This concept directly implies an MLCM 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 $\boldsymbol{B}_1$ to $\boldsymbol{B}_m$ provided that the previous bits are given. Practical design rules of the code rates can be found in [?]. The suitable labeling for MLCM is Ungerboeck's SP labeling.

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



(a) $\beta = 6$

(b) $\beta = 8$

(c) $\beta = 10$

(d) $\beta = 12$

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 β 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 1.81×10^{-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 β 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