Codebook Cardinality Spectrum of Distributed Arithmetic Codes for Stationary Memoryless Binary Sources

Yong Fang and Vladimir Stankovic

Abstract

It was demonstrated that, as a nonlinear implementation of Slepian-Wolf Coding (SWC), Distributed Arithmetic Coding (DAC) outperforms traditional Low-Density Parity-Check (LPDC) codes for short code length and biased sources. This fact triggers research efforts into theoretical analysis of DAC. In our previous work, we proposed two analytical tools, Codebook Cardinality Spectrum (CCS) and Hamming Distance Spectrum (HDS), to analyze DAC for Stationary Memoryless Binary Sources (SMBS) with uniform distribution. This paper extends our work on CCS from uniform SMBS to biased SMBS. We begin with the final CCS and then deduce each level of CCS backwards by recursion. The main finding of this paper is that the final CCS of biased SMBS is not uniformly distributed over $[0, 1)$. This paper derives the final CCS of biased SMBS and proposes a numerical algorithm for calculating CCS effectively in practice. All theoretical analyses are well verified by experimental results.

Index Terms

Distributed source coding, Slepian-Wolf coding, distributed arithmetic coding, codebook cardinality spectrum, biased sources.

Y. Fang is with the School of Information Engineering, Chang'an University, Xi'an, Shaanxi 710064, China (email: fy@chd.edu.cn). V. Stankovic is with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK (email: vladimir.stankovic@strath.ac.uk). *Corresponding author: Y. Fang*.

I. INTRODUCTION

Slepian-Wolf Coding (SWC) [1] is the lossless form of Distributed Source Coding (DSC) referring to **separate** compression and **joint** lossless reconstruction of two or more **correlated** discrete sources. Since the SWC can be mapped into a communication problem over a "virtual communication channel" [2], it is traditionally realized via channel codes, e.g., turbo codes [3], Low-Density Parity-Check (LDPC) codes [4], more recently polar codes [5], etc. Though channel coding has been recognized as a natural solution for the SWC problem, conventional source coding, e.g., Arithmetic Coding (AC) [6], [7], has also been tried. There are mainly two ideas for AC-based SWC, i.e., the Interval-Enlarging AC (IEAC) [8], [9], [10], which enlarges the mapping intervals of source symbols, and the Bit-Puncturing AC (BPAC) [11], which punctures a part of bits from AC bitstream regularly. We collectively refer to IEAC and BPAC as Distributed AC (DAC). The most significant feature of DAC distinguishing itself from channel-coding-based SWC is that DAC is a *nonlinear* code, while channel codes that are traditionally used for SWC, e.g., LDPC codes, are *linear* codes. Consequently, DAC decoding is sub-optimal and relatively complex, while state-of-the-art channel codes have optimal and low-complexity decoder designs, e.g., the Belief-Propagation (BP) algorithm. However, DAC has better adaptability to nonstationary source statistics and better performance for data blocks of short-to-medium length [8], [10], where traditional channel codes usually perform poorly.

A. Improvements and Extensions of DAC

Since its appearance in 2007 [8], [9], [10], [11], DAC attracted substantial research attention resulting in performance improvements and new applications.

The performance of DAC can be further improved. The decoding of DAC (including both IEAC and BPAC) will create a tree with exponentially-increasing complexity. To maintain linear complexity, after each decoding stage, the tree must be pruned. However, this pruning is based on the partial metrics of paths, so there is a risk that the proper path is mis-pruned. In [12], the **coupled BPAC** was proposed to handle this issue, which divides each source sequence into two coupled sub-sequences that are AC encoded and punctured independently. At the decoder, the bitstreams of two sub-sequences are decoded by two sub-decoders respectively. However, since two sub-sequences are coupled, their sub-decoders can exchange information between each other to avoid mis-pruning the proper path. Experiments show that with the coupled BPAC, better performance can be achieved as sequence length increases, when source and side information

are strongly correlated. Besides **mis-pruning of proper path**, another reason causing rate loss of DAC is the existence of **twin paths**, i.e., two paths differing from each other only in leaf nodes. This phenomenon was observed in [13]. Further, to remove twin paths, the **block DAC** was proposed in [13], which avoids mapping the sequences with small Hamming distances onto the same overlapped interval. Simulation results show that, for equally-likely memoryless sources, blocked DAC can achieve lower decoding error rate than the original DAC and can even outperform LDPC codes for short code lengths and highly correlated sources.

An important issue of DSC is that **the minimum achievable rate of source compression is unknown at the encoder**. This problem can be solved by two means: Rate-Compatible Coding and Decoder Correlation Estimation. On one hand, **DAC can be extended to realize rate-compatible coding**. In [14], the rate-compatible IEAC is studied in presence of a feedback channel. The encoder imitates decoder operations and stores all ambiguous symbols into a list. If decoding failure is detected, e.g., by Cyclic Redundancy Check (CRC), the decoder sends a request for a part of ambiguous symbols over the feedback channel. On receiving the requested ambiguous symbols, the decoder tries again and may select the proper path. Depending on in what order the ambiguous symbols are sent, there are three options: **direct ordering**, **inverse ordering**, and **hybrid ordering**. Experiments show that the rate loss of rate-compatible IEAC is negligible. On the other hand, **decoder correlation estimation can be realized with DAC**. Correlation estimation in DSC plays the same role as noise estimation in channel coding. In [15], a decoder-driven framework is proposed to estimate the unknown marginal and conditional probabilities of the source (given side information), which are fed back to the transmitter to drive the IEAC encoder. Compared with rate-compatible IEAC, an advantage of this framework is smaller decoding delay.

DAC can be extended to address the SWC of sources with memory, as shown in [16], where the **conditional BPAC** was proposed. First, the Markovian source is compressed with the conditional AC [7], and then the bitstream is punctured. At the decoder, both Markovian memory and correlated side information are made use of to aid the decoding of punctured bitstream. In addition, when the correlation between source and side information is non-stationary, the interleaving technique is used to avoid early mis-pruning of the proper path.

DAC can be extended to realize Distributed Joint Source-Channel Coding (DJSCC), as shown in [17], where the Distributed Joint Source-Channel Arithmetic Coding (DJSCAC) was proposed, which is an extension of IEAC by allowing the coexistence of overlapped intervals

and forbidden intervals. The overlapped intervals can remove the symbol-domain redundancy between the source and side information, while the forbidden intervals can counter the bit-domain noises of transmission channel. Experimental results show that the DJSCAC is better than parity-based turbo codes at relatively short block lengths.

DAC can serve as the SWC core for lossy DSC. In [18], the Nested Lattice Quantization (NLQ) is concatenated with multi-level DAC to implement Wyner-Ziv Coding (WZC) [19]—the asymmetric lossy DSC with decoder side information. The output of NLQ is first binarized and then the DAC is applied on each bitplane. The bitstreams of all bitplanes are decoded jointly by exploiting inter-bitplane dependencies and decoder side information. In addition, rate allocation among different bitplanes is studied.

DAC can be extended from binary sources to non-binary sources. The DAC was originally proposed for binary sources, so [20] proposed two non-binary DAC schemes: Multi-Interval DAC (MI-DAC) and Huffman-Coded DAC (HC-DAC). The error rate and decoding complexity of these two schemes are evaluated via computer simulations. The two schemes show similar error probability, but the MI-DAC is of lower decoding complexity. In addition, both schemes perform better when there is a strong correlation between source and side information.

B. Theoretical Analyses of DAC

However, it must be pointed out that most work on DAC summarized in sub-Sect. I-A is heuristic. We notice that DAC is a nonlinear code, so its theoretical analysis is more involved than that of linear codes, such as turbo codes, LDPC codes, and polar codes. Though some theoretical analyses have been seen for IEAC in our prior work [21], [22], [23], [24], [25], hardly any theoretical analysis is found for BPAC up to now. Actually, the analysis of BPAC is much more difficult than that of IEAC, so this paper will still focus on IEAC as our prior work. For simplicity, in the following by DAC, we refer to IEAC by default. Let us briefly overview our related advances on this issue in this sub-section.

It is recognized that DAC is to some extent like syndrome coding as it partitions source space into cosets or codebooks, so an important problem is how source space is partitioned (equally or unequally)? Considering the difficulty of this problem, we begin with the asymptotic case that block length is infinite. In the letter [21], the formula is deduced to calculate the relative size of codebooks, which is exactly verified with experimental results. For the first time, it is found that DAC partitions source space into codebooks of unequal sizes, which is quite different from

AC. The full paper [22] is an extension of [21] by supplying strict proofs. Another contribution of [22] is proving that codebook size increases exponentially as block length increases. Though this conclusion coincides with our intuition, its strict proof is still necessary.

In [23], it is recognized that the non-uniform distribution of codebook sizes implies a rate loss of DAC, so we deduce a formula to make use of the distribution of codebook sizes to improve the DAC. A theoretical contribution of [23] is that it strictly proves the existence of twin paths in the same codebook, as found in [13], and gives a loose lower bound of decoding error rate for DAC. It is also shown that random permutation can remove twin paths.

The other important issue of DAC is how codewords are spaced (densely or sparsely) within each codebook. This problem is answered by the Hamming Distance Spectrum (HDS) in [24], which is a Probability Mass Function (PMF) of the Hamming distance between sibb codewords, where *sibb codewords* are those codewords belonging to the same codebook (Note that DAC is a many-to-one mapping that partitions source space into codebooks, each of which includes one or more codewords). Different from the study on codebook size, which begins from infinite-length blocks, the study on HDS begins directly from finite-length blocks. We deduce the formula for calculating the HDS, which is exactly verified by experimental results.

The deduction of HDS in [24] is very complex, so it is simplified in [25]. Besides HDS, [25] defines the Codebook Cardinality Spectrum (CCS), which is an ensemble of Probability Density Functions (PDF). Remember that the AC encoder recursively shrinks the initial interval $[0,1)$ according to source symbols and finally outputs an arbitrary number within the final interval as the bitstream. Let $[L_i, H_i)$, where $i \geq 0$, be the mapping interval after coding the i-th source symbol, and U be the output real number in the final interval. Then $[0,1) = [L_0, H_0) \supset \cdots \supset [L_i, H_i) \supset \cdots \supset [L_n, H_n) \ni U$, where n is the length of source block. Let $U_i \triangleq \frac{U-L_i}{H_i-L_i} \in [0,1)$. The level-$i$ CCS is defined as the PDF of U_i. Especially, the PDF of U_0 is called the *initial CCS* and the PDF of U_n is called the *final CCS*. Actually, the relative size of codebooks in [21], [22], [23] is just the asymptotic form of initial CCS as block length $n \to \infty$. It is proved in [25] that by carefully tuning coding parameters, the final CCS will always be a uniform function over $[0,1)$. Then other levels of CCS can be deduced backwards by recursion.

In the original paper on DAC [8], it is found that if all symbols of each block are mapped onto enlarged intervals, the performance of DAC is very poor. So the authors of [8] modify the DAC by mapping the last few symbols of each block onto non-overlapped intervals as traditional AC. However, no theoretical explanation is given to answer why such special handling can bring

benefits. In [25], this problem is deeply studied. By tracing the change of HDS with the number of ending symbols mapping onto non-overlapped intervals, it is reasonably explained why the DAC can be improved by increasing the number of ending symbols [25].

C. Motivations and Contributions

The theoretical analyses summarized in sub-Sect. I-B are limited to Stationary Memoryless Binary Sources (SMBS) with uniform distribution. Compared to LDPC codes, DAC performs better for short data blocks and biased sources [10]. Hence, the extension of CCS and HDS from **uniform** SMBS to **biased** SMBS is important as it would potentially lead to short SWC code designs. Unfortunately, this extension is not straightforward, and the main difficulty comes from the fact that *DAC behaves as channel coding for uniform SMBS, but it looks like Joint Source-Channel Coding (JSCC) for biased SMBS*, where the difference between channel coding and JSCC is whether the length of output bitstream is fixed or not. Let R be the average rate per symbol. For a binary source with bias probability p, the lengths of mapping intervals for symbols 0 and 1 are proportional to $(1-p)^R$ and p^R, respectively. Then for each length-n binary source block, the length of output bitstream is $\lceil -n_0 \log_2 (1-p)^R - n_1 \log_2 p^R \rceil = \lceil -R(n_0 \log_2(1-p) + n_1 \log_2 p) \rceil$, where n_0 is the number of 0's and n_1 is the number of 1's in the block, i.e., $n = n_0 + n_1$. It can be seen that *if and only if* (iff) $p = 0.5$, symbols 0 and 1 are mapped onto equal-length intervals, and the length of output bitstream is always $\lceil nR \rceil$, irrespective of n_0 and n_1. Hence for uniform SMBS, the concept of codebook partitioning can be directly borrowed from channel coding: DAC partitions source space $\mathbb{B}^n$ into $2^{\lceil nR \rceil}$ codebooks. However, if $p \neq 0.5$, symbols 0 and 1 are mapped onto unequal-length intervals, and the length of output bitstream is variable (depending on n_0 and n_1). Thus for biased SMBS, it is impossible to build the concept of codebook partitioning as done for channel coding.

This paper extends our prior work on CCS from uniform SMBS to biased SMBS. The main contributions of this paper are as follows. First, we derive the algebraic form of the *final* CCS for *biased* SMBS, and show that in contrast to uniform SMBS, the *final* CCS of biased SMBS is not uniformly distributed over $[0, 1)$. Second, we derive the backward-recursion formula for other levels of CCS of biased SMBS. Finally, a numerical algorithm is proposed to calculate effectively the CCS of biased SMBS in practice.

Note that the CCS of DAC for biased SMBS was first discussed in [26]. However, strictly speaking, what addressed in [26] is actually not DAC because unequally-likely symbols are

mapped onto equal-length intervals, which leads to a channel coding-like scheme, i.e., the length of output bitstream is fixed, irrespective of n_0 and n_1, the numbers of symbols 0 and 1 in each block. Hence, the analysis in [26] is based on a simpler assumption. In contrast to [26], in this paper, unequally-likely symbols are mapped onto unequal-length intervals such that the length of output bitstream depends on n_0 and n_1, leading to a JSCC-like scheme that requires a completely new analysis approach to that taken in [26].

The rest of this paper is arranged as below. Sect. II outlines this paper. Sect. III shows some properties of arithmetic sequences and geometric sequences, which lay a foundation for the proofs of this paper. Sects. IV and V study two important issues of AC, i.e., bitstream termination and fractional-bit rate loss, both of which have significant impacts on the final CCS of biased SMBS. Sect. VI derives the final CCS, and Sect. VII deduces the backward-recursion formula of CCS. A numerical algorithm for practically calculating CCS is also given in Sect. VII. Simulation results are reported in Sect. VIII and finally, Sect. IX concludes this paper.

II. Paper Outline

This section will give an outline of this paper. First, we define some notations that will be frequently used. Then we review the principle of DAC and define some important concepts. Finally, we highlight some key steps of our proofs to be detailed in the following sections.

A. Notations

Let $\mathbb{N}$ be the set of *natural* numbers, $\mathbb{Z}$ the set of *integers*, $\mathbb{Q}$ the set of *rational* numbers, and $\mathbb{R}$ the set of *real* numbers. Hence, $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$. Let $[i:j] \triangleq \{i, \ldots, j\} \subset \mathbb{Z}$ be a closed discrete interval and $(i:j) \triangleq \{(i+1), \ldots, (j-1)\} \subset [i:j]$ an open discrete interval. Similarly, $[i:j)$ and $(i:j]$ denote two half-open discrete intervals. If $\mathcal{I} = [i:j) \subset \mathbb{Z}$, we define $a\mathcal{I} + b \triangleq \{ai + b, \ldots, a(j-1) + b\}$; if $\mathcal{I} = [l, h) \subset \mathbb{R}$, we define $a\mathcal{I} + b \triangleq [al + b, ah + b)$. The hybrid scaling-shifting operation of other forms (open or closed) of discrete or continuous intervals can be defined by analogy.

We use $\lfloor \cdot \rfloor$, $\lceil \cdot \rceil$, and $\lfloor \cdot \rceil$ to denote the flooring, ceiling, and rounding functions, respectively. If there is only one operand inside and the operand is a real number, $\{\cdot\}$ denotes the fractional part of this real number. Thus, $x = \lfloor x \rfloor + \{x\} \in \mathbb{R}$, where $\lfloor x \rfloor \in \mathbb{Z}$ and $\{x\} \in [0, 1)$. If there are two or more operands inside, $\{\cdot\}$ denotes the set formed by the operands inside, e.g., $\mathbb{B} = \{0, 1\}$ is the binary set. Depending on the operand, $|\cdot|$ may denote the absolute value of a scalar,

Fig. 1. Mapping from symbols to intervals. The left is for body and the right is for tail.

the cardinality of a set, the length of a continuous interval, or the support (i.e., the number of nonzero elements) of a vector.

Let X denote a random variable whose realization is $x \in \mathcal{X}$, where $\mathcal{X}$ is the alphabet of X. Let $f(X)$ denote a function of X whose realization is denoted by $f(x)$. Let $X_i^j \triangleq (X_i, \ldots, X_j)$. If $i = 1$, the subscript i can be dropped, i.e., $X^j = X_1^j$. If $i > j$, $X_i^j = \emptyset$, e.g., $X^0 = \emptyset$. Correspondingly, x_i^j denotes a realization of X_i^j. Similarly, $x^j = x_1^j$ and $x_i^j = \emptyset$ if $i > j$.

The Dirac delta function is denoted by $\delta(x)$. For $a, b \in \mathbb{Z}$, $\gcd(a, b)$ denotes the *greatest common divisor* and $\mathrm{lcm}(a, b)$ denotes the *least common multiple*. For $a \in [0, 1]$ or $a \in \mathbb{B}$ and $b \in [0, 1]$ or $b \in \mathbb{B}$, we define $(a \circ b) \triangleq ab + (1 - a)(1 - b)$. Thus for a binary random variable X with bias probability $\Pr(X = 1) = p \in [0, 1]$, we have $\Pr(X = x) = (x \circ p)$, where $x \in \mathbb{B}$. The uniform function over interval $\mathcal{I}$ is denoted by

$$\Pi_{\mathcal{I}}(x) \triangleq \begin{cases} \frac{1}{|\mathcal{I}|}, & x \in \mathcal{I} \\ 0, & x \notin \mathcal{I} \end{cases}. \tag{1}$$

B. Overview on DAC

Let X^n be a block of n independent and identically-distributed (i.i.d.) binary random variables with $\Pr(X_i = 1) = p$ for all $i \in [1 : n]$. The DAC divides X^n into body X^{n-t} and tail X_{n-t+1}^n, where t is called tail length. The mapping from symbols 0 and 1 to intervals obeys the rules $0 \to [0, (1 - p)^{\gamma(i)})$ and $1 \to [(1 - p^{\gamma(i)}), 1)$, where

$$\gamma(i) \triangleq \begin{cases} r, & i \in [1 : (n - t)] \\ 1, & i \in [(n - t + 1) : n] \end{cases}. \tag{2}$$

We call r the overlap factor of body symbols. The symbol-interval mapping can be visualized by Fig. 1, where the left is for body and the right is for tail. Once the symbol-interval mapping rules are set, the encoder shrinks interval $[0, 1)$ repeatedly according to source symbols and output an arbitrary number within the final interval. The overall code rate is then

$$R = \frac{(n - t)r + t}{n} H(p), \tag{3}$$

where $H(p) \triangleq -p\log_2 p-(1-p)\log_2(1-p)$. Thus, each DAC bitstream can be parameterized by (n,t,r,p). Especially, if $p=0.5$, then $H(p)=1$ and $R=\frac{(n-t)r+t}{n}$. If $t=0$, it is called **tailless DAC**, and if $t>0$, it is called **tailed DAC**.

The decoding is an inverse of encoding. For tail symbols, the decoding is the same as AC because there is no ambiguous interval. For body symbols, if the bitstream

- falls into $[0,1-p^r)$, then the 0-branch is created;
- falls into $[(1-p)^r,1)$, then the 1-branch is created;
- falls into $[1-p^r,(1-p)^r)$, then two branches are created.

The decoding will form an incomplete binary tree, and each path from the root node to a leaf node corresponds to a binary sequence. Now we need the help of side information Y^n, which is correlated to the source X^n. If the *max a posteriori* decoder is used, then the path nearest (in Hamming distance) to Y^n will be output as the estimate of X^n.

C. Definitions

The DAC encoder repeatedly scales down the primitive interval $[0,1)$ according to X^n. After coding X^i, the lower and upper bounds of the resulting interval are functions of X^i, so they are denoted by $l(X^i)$ and $h(X^i)$, respectively. Let $\mathcal{I}(X^i)\triangleq[l(X^i),h(X^i))$. Obviously, $\mathcal{I}(X^n)\subset\cdots\subset\mathcal{I}(X^0)=[0,1)$. We define an important function:

$$\eta(a,b)\triangleq -(ar+b)\log_2 p-(((n-t)-a)r+(t-b))\log_2(1-p), \tag{4}$$

where $a\in[0:(n-t)]$ and $b\in[0:t]$. According to the symbol-interval mapping rules (the detailed deduction can be found in sub-Sect. IV-A), it is easy to get $-\log_2|\mathcal{I}(X^n)|=\eta(A,B)$, where $|\mathcal{I}(X^n)|$ is the length of $\mathcal{I}(X^n)$, $A\triangleq|X^{n-t}|$ is the support of X^{n-t}, and $B\triangleq|X_{n-t+1}^n|$ is the support of X_{n-t+1}^n. Finally, $m\geq n'\triangleq\lceil\eta(A,B)\rceil$ bits Z^m will be output.

Definition II.1 (**Normalized Mapping Interval** (NMI))**.** The NMI of X^n is defined as

$$\mathcal{N}(X^n)\triangleq 2^{n'}\mathcal{I}(X^n)=[2^{n'}l(X^n),2^{n'}h(X^n))\subset[0,2^{n'}). \tag{5}$$

Definition II.2 (**Fractional-Bit Rate Loss** (FBRL))**.** Let X be an SMBS with bias probability p compressed by an (n,t,r) DAC encoder. Let $A\triangleq|X^{n-t}|$ and $B\triangleq|X_{n-t+1}^n|$. The FBRL is

$$\tau(A,B)\triangleq\lceil\eta(A,B)\rceil-\eta(A,B)\in[0,1). \tag{6}$$

According to the definition of FBRL, the length of $\mathcal{N}(X^n)$ is $|\mathcal{N}(X^n)| = 2^{\tau(A,B)} \in [1,2)$, so there may be one or two integers in $\mathcal{N}(X^n)$. Specifically, if we define $\ell(X^n) \triangleq 2^{n'} l(X^n)$, then $\lceil \ell(X^n) \rceil$ must belong to $\mathcal{N}(X^n)$, and $\lceil \ell(X^n) \rceil + 1$ may or may not belong to $\mathcal{N}(X^n)$.

Definition II.3 (**Prefix Bitstream** and **Overall Bitstream**)**.** We call Z^m the **prefix** bitstream of X^n because a potentially infinite-length bitstream z_{m+1}^{∞} may follow Z^m at the decoder. The **overall** bitstream of X^n is defined as

$$u(X^n; z_{m+1}^{\infty}) \triangleq \sum_{j=1}^{m} Z_j 2^{-j} + \sum_{j=m+1}^{\infty} z_j 2^{-j} \in \mathcal{I}(X^n). \tag{7}$$

Definition II.4 (**Normalized Overall Bitstream** (NOB))**.** The NOB of X^n is defined as

$$\varphi(X^n; z_{m+1}^{\infty}) \triangleq 2^{n'} u(X^n; z_{m+1}^{\infty}) = \sum_{j=1}^{m} Z_j 2^{n'-j} + \sum_{j=m+1}^{\infty} z_j 2^{n'-j} \in \mathcal{N}(X^n). \tag{8}$$

Definition II.5 (**Projection of Bitstream**)**.** The projection of $u(X^n; z_{m+1}^{\infty})$ onto $[l,h) \subseteq [0,1)$ is defined as $\frac{u(X^n; z_{m+1}^{\infty}) - l}{h-l}$. Especially, the **projection** of $u(X^n; z_{m+1}^{\infty})$ onto $\mathcal{I}(X^i) = [l(X^i), h(X^i))$ is called the level-i **self-projection** of $u(X^n; z_{m+1}^{\infty})$:

$$U_i \triangleq \frac{u(X^n; z_{m+1}^{\infty}) - l(X^i)}{h(X^i) - l(X^i)} \in [0,1). \tag{9}$$

Definition II.6 (**Codebook Cardinality Spectrum** (CCS))**.** The PDF of U_i is called the level-i CCS and denoted by $f_i(u)$, where $u \in [0,1)$.

Definition II.7 (**Conditional CCS**)**.** The conditional PDF of U_i given X_{i+1} is called the level-i conditional CCS and denoted by $f_i(u|x)$, where $u \in [0,1)$ and $x \in \mathbb{B}$.

According to the definition and properties of PDF, we have $f_i(u) = f_i(u|x) = 0$ for $u \notin [0,1)$, $f_i(u) \geq 0$ and $f_i(u|x) \geq 0$ for $u \in [0,1)$, and $\int_0^1 f_i(u)du = \int_0^1 f_i(u|x)du = 1$. Specifically, we call $f_0(u)$ the *initial* CCS and $f_n(u)$ the *final* CCS.

D. Road Map of the Proofs

The main contributions of this paper are deriving the analytical form of CCS for theoretical analyses (see Theorem VI.1, Corollary VI.1, and Theorem VII.1) and proposing a numerical algorithm of CCS for empirical applications (see sub-Sect. VII-C). Our deduction mainly includes three key issues.

First, we determine decoder-appending bits z_{m+1}^{∞}. As shown by Def. II.5, the CCS is closely linked with z_{m+1}^{∞}. Different z_{m+1}^{∞} may cause different overall bitstream $u(X^n; z_{m+1}^{\infty})$ and self-projections U_i's. The effect of z_{m+1}^{∞} is especially significant for the final CCS $f_n(u)$. As analyzed in Sect. IV, what bits can follow Z^m depends on how AC bitstream is terminated. The classical termination method in [7] is the most **conservative** in the sense that it allows arbitrary bits z_{m+1}^{∞} to follow Z^m, while ensuring $u(X^n; z_{m+1}^{\infty}) \in \mathcal{I}(X^n)$ in any case. Instead, an alternative to the classical termination is proposed in sub-Sect. IV-E of this paper, which is the most **progressive** in the sense that it allows only $z_{m+1}^{\infty} = 10\ldots$ to follow Z^m to ensure $u(X^n; z_{m+1}^{\infty}) \in \mathcal{I}(X^n)$. It will be proved in Sect. IV that the most conservative termination will produce the longest prefix bitstream Z^m with $m = n'$ or $n'+1$; while the most progressive termination will produce the shortest prefix bitstream Z^m with $m < n'$. A prerequisite of the most progressive termination is that the decoder must know m, while the most conservative termination has no such prerequisite. For simplicity, the deduction of this paper will be based on the most progressive termination and z_{m+1}^{∞} is fixed to $10\ldots$, where m is assumed to be known at the decoder. In this special case, we shorten $u(X^n; 10\ldots)$ to $u(X^n)$ and $\varphi(X^n; 10\ldots)$ to $\varphi(X^n)$ for conciseness.

Second, we derive the analytical form of the final CCS $f_n(u)$. This is the most complex step that is subject to three sub-problems.

- The first sub-problem is about how to explain $\varphi(X^n)$, which is closely linked with the used termination method. For the most progressive termination, $\varphi(X^n) = \sum_{j=1}^{m} Z_j 2^{n'-j} + 2^{n'-m-1}$. Since $m < n'$, we have $\varphi(X^n) \in \mathbb{Z}$. Hence, $\varphi(X^n) = \lceil \ell(X^n) \rceil$ or $\lceil \ell(X^n) \rceil + 1$, where $\ell(X^n) \triangleq 2^{n'} l(X^n)$. To derive the analytical form of $f_n(u)$, we must know under what cases, $\varphi(X^n) = \lceil \ell(X^n) \rceil$, and under what cases, $\varphi(X^n) = \lceil \ell(X^n) \rceil + 1$. In addition, we also need to know the probability of $\varphi(X^n) = \lceil \ell(X^n) \rceil$ and the probability of $\varphi(X^n) = \lceil \ell(X^n) \rceil + 1$. All these questions will be answered in Sect. IV.
- The second sub-problem is about the distribution of FBRL $\tau(A, B)$, which will be solved in Sect. V by exploiting the properties of arithmetic sequences.
- The third sub-problem is about the distribution of $c(X^n) \triangleq \lceil \ell(X^n) \rceil - \ell(X^n)$, which will be solved in sub-Sect. VI-B by exploiting the properties of geometric sequences.

After these three sub-problems are solved, the final CCS will be derived in sub-Sects. VI-C and VI-D in two different cases, respectively.

Finally, after the final CCS $f_n(u)$ is obtained, $f_i(u)$ for $i < n$ can be recursively deduced backwards. The proof for this step is quite simple, as shown in Sect. VII.

III. PRELIMINARIES

This section will give some properties of the Flooring, Ceiling, and Rounding (F/C/R) errors of arithmetic sequences and geometric sequences that are necessary for the proofs in the following sections. Before our discussion, the bijective relation between the *fractional part* of any real number and its F/C/R errors will be revealed, so that the properties of fractional parts can be directly extended to F/C/R errors. Then only the properties of fractional parts will be further addressed. In addition, we also prove a property of binomial distribution.

A. Relation of F/C/R Error and Fractional Part

Let $x = \lfloor x \rfloor + \{x\} \in \mathbb{R}$, where $\lfloor x \rfloor \in \mathbb{Z}$ and $\{x\} \in [0, 1)$. Then the flooring error of x is $f \triangleq \lfloor x \rfloor - x = -\{x\}$, the ceiling error of x is

$$c \triangleq \lceil x \rceil - x = \begin{cases} 1 - \{x\}, & 0 < \{x\} < 1 \\ 0, & \{x\} = 0 \end{cases}, \tag{10}$$

and the rounding error of x is

$$r \triangleq \lfloor x \rceil - x = \begin{cases} -\{x\}, & 0 \le \{x\} < 0.5 \\ 1 - \{x\}, & 0.5 \le \{x\} < 1 \end{cases}. \tag{11}$$

It is easy to get $f \in (-1, 0]$, $c \in [0, 1)$, and $r \in (-0.5, 0.5]$. Obviously, the F/C/R errors of any real number are all linear *bijective* functions of its factional part, so F/C/R errors must have the same properties as fractional parts. For example, if $\{x\}$ is *uniformly distributed* (u.d.) over $[0, 1)$, then f, c, and r are also u.d. over $[0, 1)$.

B. Uniform Distribution of Sequences

Let $\omega = (x_i)$, $i \in \mathbb{N}$, be a sequence of real numbers. Let $C(\mathcal{I}; n; \omega)$ be the counting function defined as the number of terms x_i, $1 \le i \le n$, for which $\{x_i\} \in \mathcal{I} \subseteq [0, 1)$.

Definition III.1 (Definition 1.1 of [27])**.** The sequence $\omega = (x_i)$, $i \in \mathbb{N}$, is *uniformly distributed modulo 1* (u.d. mod 1) if for every pair of a and b with $0 \le a < b \le 1$, we have

$$\lim_{n \to \infty} \frac{C([a, b); n; \omega)}{n} = b - a. \tag{12}$$

Proposition III.1 (Lemma 1.1 and Example 2.1 of [27])**.** Let θ be an irrational number. Then the sequence $(\alpha + i\theta)$, $i \in \mathbb{N}$, is u.d. mod 1 for any real constant α.

Proposition III.2. Let $d = l/k$, where $l, k \in \mathbb{Z}$ and $\gcd(l, k) = 1$. The sequence $(\alpha + id)$, $i \in \mathbb{N}$, is u.d. mod 1 over the discrete space $\{\beta, \beta + 1/k, \ldots, \beta + \frac{k-1}{k}\}$, where $\beta = \frac{\{k\alpha\}}{k} \in [0, 1/k)$.

Proof. This is because $(\{\alpha + id\})$, $i \in \mathbb{N}$, is a periodic sequence with circle k. □

Proposition III.3 (Corollary 4.1 and Corollary 4.2 of [27])**.** The sequence (αx^i), $i \in \mathbb{N}$, where α is a non-zero real constant, is u.d. mod 1 for almost all non-integral $x > 1$, and the exceptional set has Lebesgue measure zero.

C. Jointly Uniform Distribution of Sequences

Let $\mathbf{a} = (a_1, \ldots, a_s)$ and $\mathbf{b} = (b_1, \ldots, b_s)$. If $a_i < b_i$ for all $i \in [1 : s]$, we say $\mathbf{a} < \mathbf{b}$. The set of points $\mathbf{x} = (x_1, \ldots, x_s) \in \mathbb{R}^s$ such that $\mathbf{a} \leq \mathbf{x} < \mathbf{b}$ is denoted by $[\mathbf{a}, \mathbf{b})$. Let $\omega = (\mathbf{x}_i)$, $i \in \mathbb{N}$, where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,s})$, be a sequence of real vectors. Let $C(\mathcal{I}; n; \omega)$ be the number of terms $\mathbf{x}_i$, $1 \leq i \leq n$, for which $\{\mathbf{x}_i\} \in \mathcal{I} \subseteq [0, 1)^s$.

Definition III.2 (Definition 6.1 of [27])**.** The sequence $\omega = (\mathbf{x}_i)$, $i \in \mathbb{N}$, is u.d. mod 1 in $\mathbb{R}^s$ if for every interval hypercube $[\mathbf{b}, \mathbf{a}) \subseteq [0, 1)^s$, we have

$$\lim_{n\to\infty} \frac{C([\mathbf{a}, \mathbf{b}); n; \omega)}{n} = \prod_{j=1}^{s} (b_j - a_j). \tag{13}$$

Proposition III.4 (Theorem 6.3 of [27])**.** The sequence $(\mathbf{x}_i)$, $i \in \mathbb{N}$, where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,s})$, is u.d. mod 1 in $\mathbb{R}^s$ iff for every non-zero lattice point $\mathbf{h} \in \mathbb{Z}^s \setminus \mathbf{0}$, the sequence of real numbers (y_i), $i \in \mathbb{N}$, where $y_i = \sum_{j=1}^{s} h_j x_{i,j}$, is u.d. mod 1.

Definition III.3. We say that the sequence (x_i), $i \in \mathbb{N}$, is **jointly** u.d. mod 1 if for every $s \in \mathbb{N}$, the sequence $(\mathbf{y}_i)$, $i \in \mathbb{N}$, where $\mathbf{y}_i = (x_i, \ldots, x_{i+s-1})$, is u.d. mod 1 in $\mathbb{R}^s$.

Proposition III.5. If the sequence (x^i), $i \in \mathbb{N}$, is u.d. mod 1 and x is a transcendental number, then the sequence (x^i), $i \in \mathbb{N}$, is **jointly** u.d. mod 1.

Proof. According to Prop. III.4, the sequence $(\mathbf{y}_i)$, $i \in \mathbb{N}$, where $\mathbf{y}_i = (x^i, \ldots, x^{i+s-1})$, is u.d. mod 1 in $\mathbb{R}^s$ iff for all non-zero lattice point $\mathbf{h} \in \mathbb{Z}^s \setminus \mathbf{0}$, the sequence of real numbers (z_i), $i \in \mathbb{N}$, where $z_i = x^i \sum_{j=1}^{s} h_j x^{j-1}$, is u.d. mod 1. According to Prop. III.3, if the sequence (x^i), $i \in \mathbb{N}$, is u.d. mod 1, then the sequence (αx^i), $i \in \mathbb{N}$, is also u.d. mod 1 if the real constant $\alpha \neq 0$. Thus, the sequence (x^i), $i \in \mathbb{N}$, is jointly u.d. mod 1 if $\alpha = \sum_{j=1}^{\infty} h_j x^{j-1} \neq 0$ for every non-zero lattice point $\mathbf{h} \in \mathbb{Z}^\infty \setminus \mathbf{0}$, which implies that x is a transcendental number. □

D. Extensions of Uniform Distribution

Proposition III.6 (Example 2.9 of [27])**.** Let θ be an irrational number and d an arbitrary (rational or irrational) real number. Then the sequence $(\alpha + i\theta + jd)$, $i, j \in \mathbb{N}$, is u.d. mod 1. This property can be extended to more general cases.

Proposition III.7 (Extension of Prop. III.2)**.** Let $d_1 = l_1/k_1$ and $d_2 = l_2/k_2$, where $l_1, k_1, l_2, k_2 \in \mathbb{Z}$ and $\gcd(l_1, k_1) = \gcd(l_2, k_2) = 1$. Then the 2D arithmetic sequence $(\alpha + id_1 + jd_2)$, $i, j \in \mathbb{N}$, is u.d. mod 1 over the discrete space $\{\beta, \beta + 1/\kappa, \ldots, \beta + \frac{\kappa-1}{\kappa}\}$, where $\kappa = \mathrm{lcm}(k_1, k_2)$ and $\beta = \frac{\{\kappa\alpha\}}{\kappa} \in [0, 1/\kappa)$. This property can be extended to higher dimensional cases.

Proposition III.8 (Extension of Prop. III.3)**.** The 2D power sequence $(\alpha x^i y^j)$, $i, j \in \mathbb{N}$, where α is a non-zero real constant, is u.d. mod 1 for almost all non-integral $x > 1$ and $y > 1$, and the exceptional set has Lebesgue measure zero. This property holds in higher dimensional cases.

Proposition III.9 (Extension of Prop. III.5)**.** If the 2D power sequence $(x^i y^j)$, $i, j \in \mathbb{N}$, is u.d. mod 1 and $\alpha = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h_{i,j} x^{i-1} y^{j-1} \neq 0$ for all non-zero lattice point $\mathbf{h} \in \mathbb{Z}^{\infty \times \infty} \setminus \mathbf{0}$, then the sequence $(x^i y^j)$, $i, j \in \mathbb{N}$, is **jointly** u.d. mod 1. This property can be further extended to higher dimensional cases.

E. Equipartition Property of Binomial Distribution

Proposition III.10. Consider a binomial distribution with parameters $n \in \mathbb{N}$ and $p \in (0, 1)$. For conciseness, we assume that n can be exactly divided by k. Then as $n \to \infty$, we have $q_0 = \cdots = q_{k-1} = 1/k$, where

$$q_i \triangleq \sum_{m=0}^{n/k-1} \binom{n}{mk+i} p^{mk+i} (1-p)^{n-mk-i}. \tag{14}$$

Proof. An (n, p) binomial process includes n independent Bernoulli trials with success probability p. Let X_n be the number of successes in n consecutive Bernoulli trials and $Y_n = X_n \bmod k \in [0 : k)$. Then $(\ldots, Y_{n-1}, Y_n, Y_{n+1}, \ldots)$ forms a 1-order time-homogeneous Markov chain with transition matrix $\mathbf{T} = (t_{i,j})_{k \times k}$, where $t_{i,i} = (1-p)$ for all $i \in [0 : k)$, $t_{i,i+1} = t_{k-1,0} = p$ for all $i \in [0 : (k-1))$, and $t_{i,j} = 0$ in other cases. The steady state of $(\ldots, Y_{n-1}, Y_n, Y_{n+1}, \ldots)$ exists and its stationary distribution is $\pi = (1/k, \ldots, 1/k)$. Therefore, $\lim_{n\to\infty} q_i = \pi_i = 1/k$. □

IV. Termination of AC Bitstream

In this and the following sections, we will explain in detail how to obtain the analytical form of CCS for biased SMBS. Our deduction includes two steps. First, the analytical form of the final CCS is derived, and then, the backward-recursion formula of CCS is derived. However, before doing so, it is necessary to address two important issues—Bitstream Termination and FBRL. These two issues will be tackled in this and the next sections, respectively.

A. Normalized Mapping Interval (NMI)

Let $\mathcal{I}(X^i) \triangleq [l(X^i), h(X^i))$ be the interval of X^i. Initially, $\mathcal{I}(X^0) = \mathcal{I}(\emptyset) = [0, 1)$. Clearly, $\mathcal{I}(X^i) \subset \mathcal{I}(X^{i-1})$. If $X_i = 0$, then $l(X^i) = l(X^{i-1})$ and $\frac{|\mathcal{I}(X^i)|}{|\mathcal{I}(X^{i-1})|} = (1-p)^{\gamma(i)}$; if $X_i = 1$, then $h(X^i) = h(X^{i-1})$ and $\frac{|\mathcal{I}(X^i)|}{|\mathcal{I}(X^{i-1})|} = p^{\gamma(i)}$, where $\gamma(i)$ is defined by (2). Thus, for $i \in [1:n]$,

$$|\mathcal{I}(X^i)| = (p \circ X_i)^{\gamma(i)} \cdot |\mathcal{I}(X^{i-1})| \tag{15}$$

$$= \prod_{i'=1}^{i} (p \circ X_{i'})^{\gamma(i')} \tag{16}$$

and

$$l(X^i) = l(X^{i-1}) + X_i(1 - p^{\gamma(i)})|\mathcal{I}(X^{i-1})| \tag{17}$$

$$= \sum_{i'=1}^{i} X_{i'}(1 - p^{\gamma(i')})|\mathcal{I}(X^{i'-1})|. \tag{18}$$

From (16), we have $\eta(A, B) = -\log_2 |\mathcal{I}(X^n)|$, where $A = |X^{n-t}|$, $B = |X_{n-t+1}^n|$, and $\eta(\cdot, \cdot)$ is defined by (4). Thus $\mathcal{I}(X^n) = [l(X^n), l(X^n) + 2^{-\eta(A,B)})$. To uniquely indicate $\mathcal{I}(X^n)$, we need $m \geq n' \triangleq \lceil \eta(A, B) \rceil \geq \eta(A, B)$ bits. Let us scale $\mathcal{I}(X^n)$ up by $2^{n'}$ times to get the NMI $\mathcal{N}(X^n)$ of X^n as defined by (5). Obviously,

$$\mathcal{N}(X^n) = [\ell(X^n), \ell(X^n) + 2^{\tau(A,B)}), \tag{19}$$

where $\tau(A, B) \triangleq n' - \eta(A, B) \in [0, 1)$ as defined by (6) and

$$\ell(X^n) \triangleq 2^{n'} l(X^n) = 2^{n'} \sum_{i=1}^{n} X_i(1 - p^{\gamma(i)})|\mathcal{I}(X^{i-1})|$$

$$= 2^{\tau(A,B)} \sum_{i=0}^{n-1} X_{i+1}(1 - p^{\gamma(i+1)}) \frac{|\mathcal{I}(X^i)|}{|\mathcal{I}(X^n)|}. \tag{20}$$

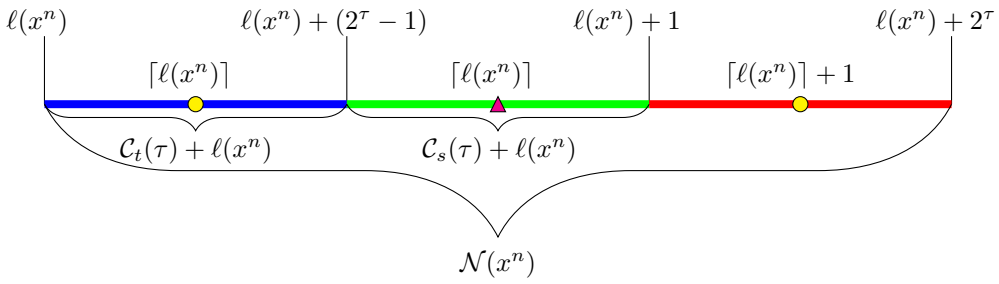


Fig. 2. Example of $\ell(x^n)$, $\lceil\ell(x^n)\rceil$, $\mathcal{N}(x^n)$, $\mathcal{C}_t(\tau)$ and $\mathcal{C}_s(\tau)$, where $\tau(a,b)$ is shortened to τ.

B. Integers in NMI

Though arbitrary number in the NMI can be output, the analysis on final CCS will be simple if **the encoder output is just an integer in the NMI**. As shown by (19), the length of NMI is $|\mathcal{N}(X^n)| = 2^{\tau(A,B)} \in [1,2)$. If $\tau(A,B)=0$, we have $|\mathcal{N}(x^n)| \equiv 1$ and there is only one integer $\lceil\ell(X^n)\rceil$ in the NMI $\mathcal{N}(X^n)$. However, if $|\mathcal{N}(X^n)|>1$, there may be one integer, i.e., $\lceil\ell(X^n)\rceil$, or two integers, i.e., $\lceil\ell(X^n)\rceil$ and $\lceil\ell(X^n)\rceil+1$, in the NMI $\mathcal{N}(X^n)$. The following questions are very important because they will impact the final CCS:

- If $|\mathcal{N}(X^n)|>1$, under what conditions, there is only one integer $\lceil\ell(X^n)\rceil$ in the NMI, and under what conditions, there are two integers $\lceil\ell(X^n)\rceil$ and $\lceil\ell(X^n)\rceil+1$ in the NMI?
- If both $\lceil\ell(X^n)\rceil$ and $\lceil\ell(X^n)\rceil+1$ belong to the NMI, which integer will be output?
- If both $\lceil\ell(X^n)\rceil$ and $\lceil\ell(X^n)\rceil+1$ belong to the NMI, how possible will $\lceil\ell(X^n)\rceil$ be output, and how possible will $\lceil\ell(X^n)\rceil+1$ be output?

The first problem is closely linked with the ceiling error of $\ell(X^n)$:

$$c(X^n) \triangleq \lceil\ell(X^n)\rceil - \ell(X^n) \in [0,1). \tag{21}$$

Proposition IV.1. For any $x^n \in \mathbb{B}^n$, if $c(x^n) \in \mathcal{C}_t(\tau(a,b))$, there will be two integers $\lceil\ell(x^n)\rceil$ and $(\lceil\ell(x^n)\rceil+1)$ in its NMI $\mathcal{N}(x^n)$; if $c(x^n) \in \mathcal{C}_s(\tau(a,b))$, there is only one integer $\lceil\ell(x^n)\rceil$ in its NMI $\mathcal{N}(x^n)$, where $|x^{n-t}|=a$, $|x_{n-t+1}^n|=b$, $\tau(a,b)$ is defined by (6), and

$$\begin{cases} \mathcal{C}_t(\tau) \triangleq [0,(2^\tau-1)) \subseteq [0,1) \\ \mathcal{C}_s(\tau) \triangleq [(2^\tau-1),1) \subseteq [0,1) \end{cases}. \tag{22}$$

Proof. Given $\tau \in [0,1)$, we have $\mathcal{C}_t(\tau) \cup \mathcal{C}_s(\tau) = [0,1)$ and $\mathcal{C}_t(\tau) \cap \mathcal{C}_s(\tau) = \emptyset$. We actually divide

the space $[0,1)$ of ceiling error $c(X^n)$ into two non-overlapping subspaces $\mathcal{C}_t(\tau)$ and $\mathcal{C}_s(\tau)$, as shown by Fig. 2. If $c(x^n) \in \mathcal{C}_t(\tau(a,b))$, we have

$$\begin{cases} \lceil \ell(x^n) \rceil \in [\ell(x^n), \ell(x^n) + (2^{\tau(a,b)} - 1)) \subset \mathcal{N}(x^n) \\ \lceil \ell(x^n) \rceil + 1 \in [\ell(x^n) + 1, \ell(x^n) + 2^{\tau(a,b)}) \subset \mathcal{N}(x^n) \end{cases}. \tag{23}$$

Thus, there are two integers $\lceil \ell(x^n) \rceil$ and $(\lceil \ell(x^n) \rceil + 1)$ in $\mathcal{N}(x^n)$, as shown by the two yellow circles in Fig. 2. If $c(x^n) \in \mathcal{C}_s(\tau(a,b))$, we have

$$\begin{cases} \lceil \ell(x^n) \rceil \in [\ell(x^n) + (2^{\tau(a,b)} - 1), \ell(x^n) + 1) \subseteq \mathcal{N}(x^n) \\ \lceil \ell(x^n) \rceil + 1 \in [\ell(x^n) + 2^{\tau(a,b)}, \ell(x^n) + 2) \not\subset \mathcal{N}(x^n) \end{cases}. \tag{24}$$

Thus, there is only one integer $\lceil \ell(x^n) \rceil$ in $\mathcal{N}(x^n)$, as shown by the magenta triangle in Fig. 2. □

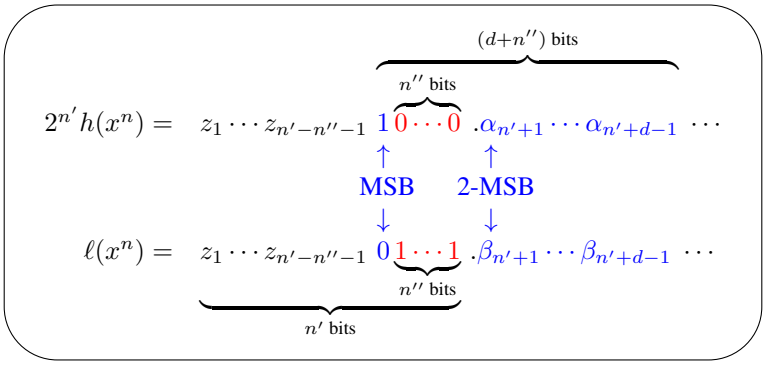
We refer to $\mathcal{C}_t(\tau(a,b))$ as the conditional **twin-integer subspace** of $c(X^n)$ and $\mathcal{C}_s(\tau(a,b))$ as the conditional **single-integer subspace** of $c(X^n)$, given $|X^{n-t}| = a$ and $|X_{n-t+1}^n| = b$. If $\tau = 0$, we have $\mathcal{C}_t(\tau) = \emptyset$ and $\mathcal{C}_s(\tau) = [0,1)$. As τ increases from 0 to 1, $\mathcal{C}_t(\tau)$ will change from $\emptyset$ to $[0,1)$ and $\mathcal{C}_s(\tau)$ will change from $[0,1)$ to $\emptyset$.

C. Interval Bounds

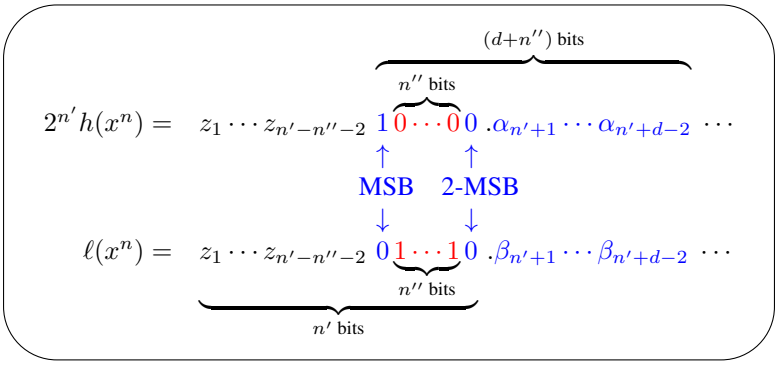
However, Prop. IV.1 is far from enough for answering the second and third questions in sub-Sect. IV-B, because the output of a real AC encoder is subject to the used termination rule that heavily depends on interval bounds $h(x^n)$ and $l(x^n)$. For conciseness, we let $\mathrm{comb}(h(x^n), l(x^n))$ denote the combination of $h(x^n)$ and $l(x^n)$. Since $|\mathcal{N}(x^n)| \in [1,2)$, there are only three possible patterns for $\mathrm{comb}(h(x^n), l(x^n))$, as shown by Fig. 3, where $n' = \lceil \eta(a,b) \rceil$, d is buffer depth (in bits) for interval bounds, and $n'' \geq 0$ is the number of underflow bits. For simplicity, we define $\xi_\alpha \triangleq \sum_{i'=n'+1}^{\infty} \alpha_{i'} 2^{n'-i'} \in [0,1)$ and $\xi_\beta \triangleq \sum_{i'=n'+1}^{\infty} \beta_{i'} 2^{n'-i'} \in [0,1)$, where $\alpha_{i'}, \beta_{i'} \in \mathbb{B}$ for all $i' \in [(n'+1) : \infty)$. Obviously, $\xi_\beta = \{\ell(x^n)\}$. The following proposition is an alternate of Prop. IV.1 that links $\mathrm{comb}(h(x^n), l(x^n))$ with $\mathcal{C}_s(\tau)$ and $\mathcal{C}_t(\tau)$ defined by (22).

Proposition IV.2. For any $x^n \in \mathbb{B}^n$, if $\mathrm{comb}(h(x^n), l(x^n))$ belongs to the **N**-pattern and $c(x^n) > 0$, then $c(x^n) \in \mathcal{C}_s(\tau(a,b))$; otherwise, $c(x^n) \in \mathcal{C}_t(\tau(a,b))$.

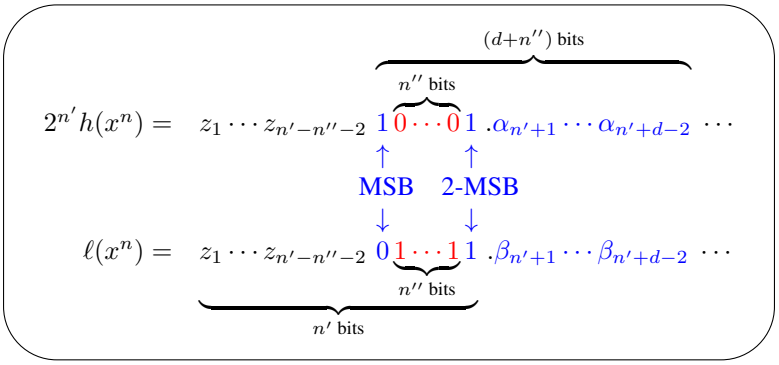
Proof. The so-called **N**-pattern is defined by Fig. 3(a), which means $|\mathcal{N}(x^n)| = 1 + \xi_\alpha - \xi_\beta$. Since $\xi_\alpha, \xi_\beta \in [0,1)$, it is sure that $0 < |\mathcal{N}(x^n)| < 2$. The condition for $1 \leq |\mathcal{N}(x^n)|$ is $0 \leq \xi_\beta \leq \xi_\alpha$. Further, depending on ξ_β, there are one or two integers in $\mathcal{N}(x^n)$.



(a) $\mathrm{comb}(h(x^n), l(x^n)) \in$ **N**-Pattern, where $0 \leq \xi_\beta \leq \xi_\alpha$.



(b) $\mathrm{comb}(h(x^n), l(x^n)) \in$ **Z**-Pattern, where $0 \leq \xi_\alpha < \xi_\beta$.



(c) $\mathrm{comb}(h(x^n), l(x^n)) \in$ **O**-Pattern, where $0 \leq \xi_\alpha < \xi_\beta$.

Fig. 3. Three patterns of binarized bounds of NMI, where $n' = \lceil \eta(a,b) \rceil$, d is buffer depth (in bits) for binarized bounds, and $n'' \geq 0$ is the number of underflow bits. The MSB stands for the Most Significant Bit and the 2-MSB stands for the 2-nd Most Significant Bit. (a) **N**-pattern, where **N** means that there is **N**o 2-MSB before the decimal point. (b) **Z**-pattern, where **Z** means that both 2-MSBs are **Z**ero. (c) **O**-pattern, where **O** means that both 2-MSBs are **O**ne.

- If $\xi_\beta > 0$, then $c(x^n) > 0$ and $c(x^n) \in \mathcal{C}_s(\tau(a,b))$ as there is only one integer in $\mathcal{N}(x^n)$:

$$\lceil \ell(x^n) \rceil = z_1 \cdots z_{n'-n''-1}\, 1 \overbrace{0 \cdots 0}^{n'' \text{ bits}}. \tag{25}$$

- If $\xi_\beta = 0$, then $c(x^n) = 0 \in \mathcal{C}_t(\tau(a,b))$ as there are two integers in $\mathcal{N}(x^n)$:

$$\begin{cases} \lceil \ell(x^n) \rceil + 1 = & z_1 \cdots z_{n'-n''-1}\, 1 \overbrace{0 \cdots 0}^{n'' \text{ bits}} \\ \lceil \ell(x^n) \rceil = & z_1 \cdots z_{n'-n''-1}\, 0 \underbrace{1 \cdots 1}_{n'' \text{ bits}} \end{cases}. \tag{26}$$

The so-called **Z**-pattern is defined by Fig. 3(b), which means $|\mathcal{N}(x^n)| = 2 + \xi_\alpha - \xi_\beta$. Since $\xi_\alpha, \xi_\beta \in [0,1)$, it is sure that $1 < |\mathcal{N}(x^n)|$, and the condition for $|\mathcal{N}(x^n)| < 2$ is $0 \le \xi_\alpha < \xi_\beta$. Clearly, $c(x^n) > 0$ and $c(x^n) \in \mathcal{C}_t(\tau(a,b))$ as there are always two integers in $\mathcal{N}(x^n)$:

$$\begin{cases} \lceil \ell(x^n) \rceil + 1 = & z_1 \cdots z_{n'-n''-2}\, 1 \overbrace{0 \cdots 0}^{n'' \text{ bits}} 0 \\ \lceil \ell(x^n) \rceil = & z_1 \cdots z_{n'-n''-2}\, 0 \underbrace{1 \cdots 1}_{n'' \text{ bits}} 1 \end{cases}. \tag{27}$$

The so-called **O**-pattern is defined by Fig. 3(c), which is actually an up-shifted form of the **Z**-pattern. Thus $c(x^n) > 0$ and $c(x^n) \in \mathcal{C}_t(\tau(a,b))$ as there are always two integers in $\mathcal{N}(x^n)$:

$$\begin{cases} \lceil \ell(x^n) \rceil + 1 = & z_1 \cdots z_{n'-n''-2}\, 1 \overbrace{0 \cdots 0}^{n'' \text{ bits}} 1 \\ \lceil \ell(x^n) \rceil = & z_1 \cdots z_{n'-n''-2}\, 1 \underbrace{0 \cdots 0}_{n'' \text{ bits}} 0 \end{cases}. \tag{28}$$

□

D. Classical Termination of AC Bitstream

Let Z^m be the encoder output of X^n. At the decoder, Z^m may be followed by $z_{m+1}^\infty \in \mathbb{B}^\infty$. If only $\varphi(X^n; z_{m+1}^\infty) \in \mathcal{N}(X^n)$, the correctness of decoding can be guaranteed, where $\mathcal{N}(X^n)$ is the NMI of X^n defined by (5) and $\varphi(X^n; z_{m+1}^\infty)$ is the NOB of X^n defined by (8). Further, to simplify the analysis, we hope to explain $\varphi(X^n; z_{m+1}^\infty)$ as an integer. Unfortunately, as analyzed below, though the classical termination in [7] can ensure $\varphi(X^n; z_{m+1}^\infty) \in \mathcal{N}(X^n)$ for arbitrary appending bits z_{m+1}^∞, it fails to explain $\varphi(X^n; z_{m+1}^\infty)$ as an integer in all cases.

A real AC codec is subject to a certain computation precision, so two d-bit integers, $high$ and low, are defined to store windowed binarized NMI bounds. The termination rule in [7] exploits the second Most Significant Bits (2-MSBs) of $high$ and low to remove the ambiguity of encoder output when it is followed by arbitrary bits. Let MSB_h, MSB_l, 2-MSB_h, and 2-MSB_l be the MSB or 2-MSB of $high$ or low. As shown by the first column of Tab. I, depending on the combination of 2-MSB_h and 2-MSB_l, the **N**-pattern defined in Fig. 3 can be further divided into three sub-

TABLE I
FINAL VALUES OF WINDOWED BINARIZED INTERVAL BOUNDS AND ENCODER OUTPUT

	$high$ and low (d bits)	encoder output [7]	encoder output*
N0-pattern ($\alpha_{n'+1} = \beta_{n'+1} = 0$)	$\begin{cases} 1.0\,\alpha_{n'+2}\cdots\alpha_{n'+d-1} \\ 0.0\,\beta_{n'+2}\cdots\beta_{n'+d-1} \end{cases}$	$\overbrace{z_1\cdots z_{n'-n''-1}\,0\underbrace{1\cdots1}_{n''\text{ bits}}.1}^{(n'+1)\text{ bits}=\lceil\ell(x^n)\rceil\pm0.5}$	$\overbrace{z_1\cdots z_{n'-n''-1}}^{(n'-n''-1)\text{ bits}}$
N1-pattern ($\alpha_{n'+1} = \beta_{n'+1} = 1$)	$\begin{cases} 1.1\,\alpha_{n'+2}\cdots\alpha_{n'+d-1} \\ 0.1\,\beta_{n'+2}\cdots\beta_{n'+d-1} \end{cases}$	$\overbrace{z_1\cdots z_{n'-n''-1}\,1\underbrace{0\cdots0}_{n''\text{ bits}}.0}^{(n'+1)\text{ bits}=\lceil\ell(x^n)\rceil}$	
NX-pattern ($\alpha_{n'+1} = 1$, $\beta_{n'+1} = 0$)	$\begin{cases} 1.1\,\alpha_{n'+2}\cdots\alpha_{n'+d-1} \\ 0.0\,\beta_{n'+2}\cdots\beta_{n'+d-1} \end{cases}$	$\overbrace{z_1\cdots z_{n'-n''-1}\,0\underbrace{1\cdots1}_{n''\text{ bits}}.1}^{(n'+1)\text{ bits}=\lceil\ell(x^n)\rceil\pm0.5}$ or $\underbrace{z_1\cdots z_{n'-n''-1}\,1\overbrace{0\cdots0}^{n''\text{ bits}}.0}_{(n'+1)\text{ bits}=\lceil\ell(x^n)\rceil(+1)}$	
Z-pattern	$\begin{cases} 10.\alpha_{n'+1}\cdots\alpha_{n'+d-2} \\ 00.\beta_{n'+1}\cdots\beta_{n'+d-2} \end{cases}$	$\underbrace{z_1\cdots z_{n'-n''-2}\,0\overbrace{1\cdots1}^{n''\text{ bits}}1.}_{n'\text{ bits}=\lceil\ell(x^n)\rceil}$	$\overbrace{z_1\cdots z_{n'-n''-2}}^{(n'-n''-2)\text{ bits}}$
O-pattern	$\begin{cases} 11.\alpha_{n'+1}\cdots\alpha_{n'+d-2} \\ 01.\beta_{n'+1}\cdots\beta_{n'+d-2} \end{cases}$	$\underbrace{z_1\cdots z_{n'-n''-2}\,1\overbrace{0\cdots0}^{n''\text{ bits}}0.}_{n'\text{ bits}=\lceil\ell(x^n)\rceil}$	

*The encoder output of the shortest termination.

patterns: **N0**-pattern ($\alpha_{n'+1} = \beta_{n'+1} = 0$), **N1**-pattern ($\alpha_{n'+1} = \beta_{n'+1} = 1$), and **NX**-pattern ($\alpha_{n'+1} = 1$ and $\beta_{n'+1} = 0$). For both **N0**-pattern and **N1**-pattern, we have $0 \le \xi_\alpha - \xi_\beta < 0.5$; while for the **NX**-pattern, we have $0.5 \le \xi_\alpha - \xi_\beta < 1$. There are three cases:

- 2-MSB$_h$ = 2-MSB$_l$ = 0: It includes the **N0**-pattern and the **Z**-pattern. The encoder outputs two bits 01 as ending bits.
- 2-MSB$_h$ = 2-MSB$_l$ = 1: It includes the **N1**-pattern and the **O**-pattern. The encoder outputs two bits 10 as ending bits;
- 2-MSB$_h$ = 1 and 2-MSB$_l$ = 0: It corresponds to the **NX**-pattern. Either 01 or 10 can be output, but [7] recommends to output 10 as ending bits.

In Tab. I, the second column lists the final states of $high$ and low after coding x^n, while the third column lists the encoder output according to the termination rule in [7] for each pattern defined in Fig. 3. For clarity, the decimal points are shown in the second and third columns of Tab. I. It can be found that for both **Z**-pattern and **O**-pattern, the length of encoder output is $m = n'$; but for the **N**-pattern, the length of encoder output is $m = (n' + 1)$. It is easy to verify that no matter what bits z_{m+1}^{∞} follow the encoder output Z^m, the NOB $\varphi(X^n; z_{m+1}^{\infty})$ always falls

into the NMI $\mathcal{N}(X^n)$, ensuring the correctness of decoding. If $z_{m+1}^{\infty} = 0\ldots$, then

- For the **Z**-pattern, the **O**-pattern, and the **N1**-pattern, $\varphi(x^n; 0\ldots) = \lceil \ell(x^n) \rceil$.
- For the **N0**-pattern, $\varphi(x^n; 0\ldots) = \lceil \ell(x^n) \rceil \pm 0.5$ ('$-$' for $\xi_\beta > 0$ and '$+$' for $\xi_\beta = 0$).
- For the **NX**-pattern, if the ending bits are 10, then $\varphi(x^n; 0\ldots) = \lceil \ell(x^n) \rceil$ for $\xi_\beta > 0$ and $\varphi(x^n; 0\ldots) = \lceil \ell(x^n) \rceil + 1$ for $\xi_\beta = 0$; if the ending bits are 01, then $\varphi(x^n; 0\ldots) = \lceil \ell(x^n) \rceil \pm 0.5$ ('$-$' for $\xi_\beta > 0$ and '$+$' for $\xi_\beta = 0$).

Hence, the classical termination in [7] will cause complex explanations of encoder output that hinders our analysis on the final CCS. So we need a simpler termination rule.

E. Shortest Termination of AC Bitstream

The classical termination in [7] is based on such an assumption: The length of encoder output m is unknown at the decoder, so it is unpredictable what bits will be appended to encoder output at the decoder. However, most of data are nowadays packetized, so the length of encoder output can easily be inferred at the decoder from the overhead information. Based on this assumption, we propose a novel termination rule: The encoder just outputs all identical bits of binarized $h(x^n)$ and $l(x^n)$. As shown by the fourth column of Tab. I, with this rule, the length of encoder output is $m = (n' - n'' - 1)$ for the **N**-pattern and $m = (n' - n'' - 2)$ for both **Z**-pattern and **O**-pattern. If only we append $z_{m+1}^{\infty} = 10\ldots$ to Z^m,

$$\varphi(X^n; 10\ldots) \in [0 : 2^{n'}) \cap \mathcal{N}(X^n) \subset \mathbb{Z}, \tag{29}$$

i.e., the NOB is always an integer in the NMI. Actually, this method will produce the shortest encoder output, thus we refer to it as the **shortest termination**. Though the encoder output of the shortest termination is $(n'' + 2)$-bits shorter than that of the classical termination, some overhead bits are needed to convey the length of encoder output. In addition, with the classical termination, there is no special demand on the appending bits; while with the shortest termination, only $z_{m+1}^{\infty} = 10\ldots$ can be appended to the encoder output Z^m. From now on, **only the shortest termination with appending bits** $10\ldots$ **will be further considered**. For conciseness, $\varphi(X^n; 10\ldots)$ will be shortened to $\varphi(X^n)$ and $u(X^n; 10\ldots)$ will be shortened to $u(X^n)$.

Now we are ready to answer the second and third problems in sub-Sect. IV-B. The following proposition answers the second problem in sub-Sect. IV-B.

Proposition IV.3. If $c(x^n) \in \mathcal{C}_s(\tau(a, b))$, then $\varphi(x^n) = \lceil \ell(x^n) \rceil$. As for $c(x^n) \in \mathcal{C}_t(\tau(a, b))$,

- If $c(x^n) = 0$ or $\mathrm{comb}(h(x^n), l(x^n)) \in$ **Z**-pattern, then $\varphi(x^n) = \lceil \ell(x^n) \rceil + 1$;
- If $\mathrm{comb}(h(x^n), l(x^n)) \in$ **O**-pattern, then $\varphi(x^n) = \lceil \ell(x^n) \rceil$.

Proof. Please refer to (25), (26), (27), and (28), respectively. □

Hence, **the encoder output of the shortest termination is always an integer** ($\lceil \ell(x^n) \rceil + 1$ or $\lceil \ell(x^n) \rceil$), simplifying our analysis. Let p_o denote the conditional probability of $\varphi(X^n) = \lceil \ell(X^n) \rceil$ and p_z the conditional probability of $\varphi(X^n) = \lceil \ell(X^n) \rceil + 1$, given $c(X^n) \in \mathcal{C}_t(\tau(A, B))$, i.e.,

$$\begin{cases} p_z \triangleq \Pr(\varphi(X^n) = \lceil \ell(X^n) \rceil + 1 \,|\, c(X^n) \in \mathcal{C}_t(\tau(A, B))) \\ p_o \triangleq \Pr(\varphi(X^n) = \lceil \ell(X^n) \rceil \,|\, c(X^n) \in \mathcal{C}_t(\tau(A, B))) \end{cases}. \tag{30}$$

The following proposition answers the third problem in sub-Sect. IV-B.

Proposition IV.4. $\lim_{n\to\infty} \frac{p_z}{p_o} = 1$.

Proof. According to Prop. IV.2, if $c(X^n) \in \mathcal{C}_t(\tau(A, B))$, there are three cases: $c(X^n) = 0$, $\mathrm{comb}(h(X^n), l(X^n)) \in$ **Z**-pattern, or $\mathrm{comb}(h(X^n), l(X^n)) \in$ **O**-pattern. Compared with other two cases, the case $c(X^n) = 0$ is negligible because it is a very strict condition, rarely fulfilled. Since the **O**-pattern is just an up-shifted version of the **Z**-pattern, by Prop. IV.3, we have

$$\lim_{n\to\infty} \frac{p_z}{p_o} = \lim_{n\to\infty} \frac{\Pr(\mathrm{comb}(h(X^n), l(X^n)) \in \textbf{Z}\text{-pattern})}{\Pr(\mathrm{comb}(h(X^n), l(X^n)) \in \textbf{O}\text{-pattern})} = 1. \tag{31}$$

□

V. Fractional-Bit Rate Loss

Let X be an SMBS with bias probability p, compressed by an (n, t, r) DAC encoder. Let $A \triangleq |X^{n-t}|$ and $B \triangleq |X_{n-t+1}^n|$. As analyzed in Sect. II-C, to uniquely indicate $\mathcal{I}(X^n)$, at least $\eta(A, B) \in \mathbb{R}$ bits are needed in theory, where $\eta(\cdot, \cdot)$ is defined by (4). However, since the output of a real-world DAC encoder must contain a whole number of bits, at least $\lceil \eta(A, B) \rceil \in \mathbb{Z}$ bits are needed in practice, which will cause the FBRL $\tau(A, B)$ as defined by (6). Since X is memoryless, A and B are mutually independent. Hence, A obeys the $(n - t, p)$ binomial distribution, while B obeys the (t, p) binomial distribution:

$$\begin{cases} p_{_A}(a) \triangleq \Pr(A = a) = \binom{n-t}{a} p^a (1-p)^{n-t-a} \\ p_{_B}(b) \triangleq \Pr(B = b) = \binom{t}{b} p^b (1-p)^{t-b} \end{cases}, \tag{32}$$

where $a \in [0 : (n - t)]$ and $b \in [0 : t]$. This section will deduce the distribution of $\tau(A, B)$.

A. Space of FBRL

Before answering how $\tau(A,B)$ is distributed, we must know its space. Let us rewrite (4) as

$$\begin{aligned}\eta(a,b) &= -((n-t)r+t)\log_2(1-p) + (ar+b)\log_2\tfrac{1-p}{p} \\ &= \eta(0,0) + ard + bd, \end{aligned} \tag{33}$$

where $d = \log_2\frac{1-p}{p}$. Clearly, $\eta(a,b)$'s form a 2D arithmetic sequence with common differences rd and d, so the space of $\tau(A,B)$ depends on the values of rd and d:

- **Birational Case** ($d \in \mathbb{Q}$ and $rd \in \mathbb{Q}$): Let $d = l/k$ and $rd = l_r/k_r$, where l, k, l_r, $k_r \in \mathbb{Z}$ and $\gcd(k,l) = \gcd(k_r,l_r) = 1$. According to Prop. III.7, we have $\tau(a,b) \in \{\dot\tau, \dot\tau + \frac{1}{\kappa}, \ldots, \dot\tau + \frac{\kappa-1}{\kappa}\}$, where $\kappa = \mathrm{lcm}(k_r,k)$ and $\dot\tau = \frac{\{\kappa\eta(0,0)\}}{\kappa} \in [0, \frac{1}{\kappa})$.
- **Irrational Case** ($d \notin \mathbb{Q}$ and/or $rd \notin \mathbb{Q}$): According to Prop. III.6, we have $\tau(a,b) \in [0,1)$.

B. Distribution of FBRL

Let $f_{_T}(\tau)$, where $\tau \in [0,1)$, denote the PDF of $\tau(A,B)$ and let $d = \log_2\frac{1-p}{p}$. Depending on the values of d and rd, $f_{_T}(\tau)$ has two different forms.

Proposition V.1. Let $d = l/k \in \mathbb{Q}$ and $rd = l_r/k_r \in \mathbb{Q}$, where l, k, l_r, $k_r \in \mathbb{Z}$ and $\gcd(l,k) = \gcd(l_r,k_r) = 1$. Let $\kappa = \mathrm{lcm}(k,k_r)$ and $\dot\tau = \frac{\{\kappa\eta(0,0)\}}{\kappa}$. Then as $(n-t) \to \infty$ and $t \to \infty$,

$$f_{_T}(\tau) = \frac{1}{\kappa}\sum_{\lambda=0}^{\kappa-1}\delta(\tau - \dot\tau - \tfrac{\lambda}{\kappa}). \tag{34}$$

Proof. Given $X^n = x^n$, the conditional PDF of $\tau(A,B)$ is $\delta(\tau - \tau(a,b))$, where $a = |x^{n-t}| \in [0:(n-t)]$ and $b = |x_{n-t+1}^n| \in [0:t]$. Note the difference between τ and $\tau(\cdot,\cdot)$, that is τ is a scalar value, while $\tau(\cdot,\cdot)$ is a function. Then

$$f_{_T}(\tau) = \sum_{x^n\in\mathbb{B}^n}\Pr(X^n = x^n)\delta(\tau - \tau(a,b)). \tag{35}$$

Since X is an SMBS, it is obvious that $\Pr(X^n = x^n) = p^{a+b}(1-p)^{n-a-b}$, where $a = |x^{n-t}|$ and $b = |x_{n-t+1}^n|$, and further

$$f_{_T}(\tau) = \sum_{a=0}^{n-t}\sum_{b=0}^{t} p_{_A}(a)p_{_B}(b)\delta(\tau - \tau(a,b)). \tag{36}$$

As proved in Sect. V-A, $\tau(a,b) \in \{\dot{\tau}, \cdots, \dot{\tau} + \frac{\kappa-1}{\kappa}\}$. By the equipartition property of binomial distribution shown by Prop. III.10 proved in Sect. III-E, as $(n-t) \to \infty$ and $t \to \infty$, $\tau(A,B)$ will be uniformly distributed over its discrete space. □

Corollary V.1. If $d \notin \mathbb{Q}$ and/or $rd \notin \mathbb{Q}$, then $f_{_T}(\tau) = \Pi_{[0,1)}(\tau)$ as $(n-t) \to \infty$ and $t \to \infty$.

Proof. Since the irrational case is the asymptotic form of the birational case as $\kappa \to \infty$, it is obvious that this corollary holds. □

VI. Analytical Form of Final CCS

We are now ready to study the CCS of biased SMBS. The proposed methodology is similar to that used for uniform SMBS [25]: We begin with the final CCS and then recursively deduce other levels of CCS backwards until the initial CCS. However, in contrast to the final CCS of uniform SMBS, which is always a uniform function after carefully tuning parameters, the final CCS of biased SMBS is very complex. We devote this whole section to the final CCS of biased SMBS. For simplicity, we shorten $u(X^n; z_{m+1}^{\infty})$, where $z_{m+1}^{\infty} = 10\ldots$, to $u(X^n)$ below.

A. Decomposition of Final CCS

Let us define

$$\phi(X^n) \triangleq 2^{-\tau(A,B)}(\varphi(X^n) - \ell(X^n)), \tag{37}$$

where $\varphi(X^n)$ is defined by (8), $\ell(X^n)$ is defined by (20), $A = |X^{n-t}|$ and $B = |X_{n-t+1}^{n}|$. According to (9), we have

$$U_n = \frac{u(X^n) - l(X^n)}{h(X^n) - l(X^n)} = \frac{2^{n'}u(X^n) - 2^{n'}l(X^n)}{2^{n'}(h(X^n) - l(X^n))} = \phi(X^n). \tag{38}$$

Therefore,

$$\begin{aligned} f_n(u) &= \sum_{x^n \in \mathbb{B}^n} \Pr(X^n = x^n)\delta(u - \phi(x^n)) \\ &= \sum_{a=0}^{n-t}\sum_{b=0}^{t} p_A(a)p_B(b)g_{a,b}(u), \end{aligned} \tag{39}$$

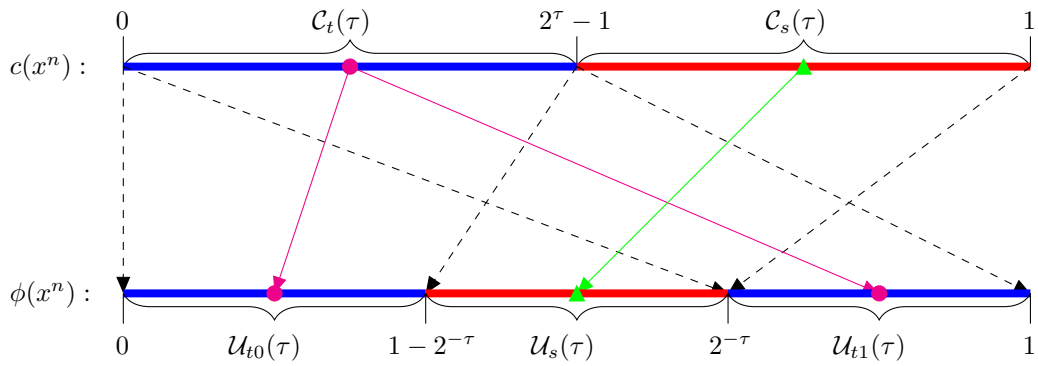


Fig. 4. Mapping from $\mathcal{C}_t(\tau)$ and $\mathcal{C}_s(\tau)$ to $\mathcal{U}_{t0}(\tau)$, $\mathcal{U}_s(\tau)$, and $\mathcal{U}_{t1}(\tau)$, where $\tau(a,b)$ is shortened to τ.

where

$$g_{a,b}(u) \triangleq \frac{1}{\binom{n-t}{a}\binom{t}{b}} \sum_{x^{n-t}:|x^{n-t}|=a} \sum_{x_{n-t+1}^n:|x_{n-t+1}^n|=b} \delta(u-\phi(x^n)). \tag{40}$$

Evidently, $\int_0^1 g_{a,b}(u)du = 1$ and $g_{a,b}(u)$ is the conditional PDF of U_n given $|X^{n-t}| = a$ and $|X_{n-t+1}^n| = b$. We call $g_{a,b}(u)$ the conditional final CCS.

B. Conditional Final CCS

Proposition VI.1. If the sequence $\omega(p,r) = (p^{-i}(1-p)^{-j}p^{-kr}(1-p)^{-lr})$, $i,j,k,l \in \mathbb{N}$, is u.d. mod 1, then $c(X^n)$ will be u.d. over $[0,1)$ as $(n-t) \to \infty$ and $t \to \infty$.

Proof. For $i \in [(n-t):n]$,

$$\frac{|\mathcal{I}(X^i)|}{|\mathcal{I}(X^n)|} = p^{-|X_{i+1}^n|} \cdot (1-p)^{-((n-i)-|X_{i+1}^n|)} \geq 1, \tag{41}$$

where $|X_{i+1}^n|$ denotes the number of non-zero elements of X_{i+1}^n. Further, for $i \in [0:(n-t)]$,

$$\frac{|\mathcal{I}(X^i)|}{|\mathcal{I}(X^n)|} = p^{-r|X_{i+1}^{n-t}|} \cdot (1-p)^{-r((n-t-i)-|X_{i+1}^{n-t}|)} \cdot \frac{|\mathcal{I}(X^{n-t})|}{|\mathcal{I}(X^n)|} \geq 1. \tag{42}$$

Obviously, $p^{-1} > 1$, $(1-p)^{-1} > 1$, $p^{-r} > 1$, and $(1-p)^{-r} > 1$. It is shown by (20) that $\ell(X^n)$ is actually the weighted sum of partial terms randomly drawn from the sequence $\omega(p,r)$. Thus, if the sequence $\omega(p,r)$ is u.d. mod 1, $\ell(X^n)$ will be u.d. mod 1 as $(n-t) \to \infty$ and $t \to \infty$. □

Proposition VI.2. If the sequence $\omega(p,r)=(p^{-i}(1-p)^{-j}p^{-kr}(1-p)^{-lr})$, $i,j,k,l\in\mathbb{N}$, is u.d. mod 1, then as $(n-t)\to\infty$ and $t\to\infty$, we have $g_{a,b}(u)=\psi_{\tau(a,b)}(u)$, where

$$\psi_\tau(u)\triangleq\begin{cases}2^{\tau-1}, & u\in[0,(1-2^{-\tau}))\cup[2^{-\tau},1)\\ 2^{\tau}, & u\in[(1-2^{-\tau}),2^{-\tau})\\ 0, & u\notin[0,1)\end{cases}. \tag{43}$$

Proof. Based on the space division of $c(X^n)$ defined by (22), we further divide the space $[0,1)$ of $U_n=\phi(X^n)$ into three non-overlapping subspaces:

$$\begin{cases}\mathcal{U}_{t0}(\tau)\triangleq 2^{-\tau}\mathcal{C}_t(\tau)=[0,(1-2^{-\tau}))\\ \mathcal{U}_{s}(\tau)\triangleq 2^{-\tau}\mathcal{C}_s(\tau)=[(1-2^{-\tau}),2^{-\tau})\\ \mathcal{U}_{t1}(\tau)\triangleq 2^{-\tau}(\mathcal{C}_t(\tau)+1)=[2^{-\tau},1)\end{cases}. \tag{44}$$

Clearly, $|\mathcal{U}_{t0}(\tau)|=|\mathcal{U}_{t1}(\tau)|=(1-2^{-\tau})\in[0,0.5)$ and $|\mathcal{U}_s(\tau)|=(2^{1-\tau}-1)\in(0,1]$. The mapping from $\mathcal{C}_t(\tau)$ and $\mathcal{C}_s(\tau)$ to $\mathcal{U}_{t0}(\tau)$, $\mathcal{U}_s(\tau)$, and $\mathcal{U}_{t1}(\tau)$ can be well illustrated by Fig. 4. Given $|x^{n-t}|=a$ and $|x_{n-t+1}^n|=b$,

- If $c(x^n)\in\mathcal{C}_t(\tau(a,b))$, then $\varphi(x^n)=\lceil\ell(x^n)\rceil$ or $\lceil\ell(x^n)\rceil+1$, and $\phi(x^n)=2^{-\tau(a,b)}c(x^n)\in\mathcal{U}_{t0}(\tau(a,b))$ or $\phi(x^n)=2^{-\tau(a,b)}(c(x^n)+1)\in\mathcal{U}_{t1}(\tau(a,b))$. This mapping can be exampled with the arrowed lines between magenta circles in Fig. 4.
- If $c(x^n)\in\mathcal{C}_s(\tau(a,b))$, then $\varphi(x^n)\equiv\lceil\ell(x^n)\rceil$ and $\phi(x^n)=2^{-\tau(a,b)}c(x^n)\in\mathcal{U}_s(\tau(a,b))$. This mapping can be exampled with the arrowed lines between green triangles in Fig. 4.

Let us pay attention to the following three points:

- According to Prop. VI.1, as $(n-t)\to\infty$ and $t\to\infty$, $c(X^n)$ will be u.d. over $[0,1)$.
- According to Prop. IV.4, given $c(x^n)\in\mathcal{C}_t(\tau(a,b))$, as $(n-t)\to\infty$ and $t\to\infty$, the probability of $\varphi(X^n)=\lceil\ell(X^n)\rceil$ is equal to that of $\varphi(X^n)=\lceil\ell(X^n)\rceil+1$.
- Both $c(x^n)\to 2^{-\tau(a,b)}c(x^n)$ and $c(x^n)\to 2^{-\tau(a,b)}(c(x^n)+1)$ are linear bijective mappings.

Based on the above analyses, as $(n-t)\to\infty$ and $t\to\infty$,

- $g_{a,b}(u)$ is piecewise uniform over $\mathcal{U}_{t0}(\tau(a,b))$, $\mathcal{U}_s(\tau(a,b))$, and $\mathcal{U}_{t1}(\tau(a,b))$, respectively;
- $g_{a,b}(u)\equiv\mu$ for all $u\in\mathcal{U}_{t0}(\tau(a,b))\cup\mathcal{U}_{t1}(\tau(a,b))$ and $g_{a,b}(u)\equiv 2\mu$ for all $u\in\mathcal{U}_s(\tau(a,b))$, where μ is a constant that can be obtained by making use of $\int_0^1 g_{a,b}(u)du=1$, i.e.,

$$\mu=\frac{1}{|\mathcal{U}_{t0}(\tau(a,b))|+2|\mathcal{U}_s(\tau(a,b))|+|\mathcal{U}_{t1}(\tau(a,b))|}=2^{\tau(a,b)-1}. \tag{45}$$

Hence, as $(n-t)\to\infty$ and $t\to\infty$, $g_{a,b}(u)=\psi_{\tau(a,b)}(u)$, where $u\in[0,1)$ and $\tau\in[0,1)$. □

C. Final CCS in Birational Case

Theorem VI.1. Let $d = \log_2 \frac{1-p}{p}$. Assume $d = l/k$ and $rd = l_r/k_r$, where l, k, l_r, $k_r \in \mathbb{Z}$ and $\gcd(l,k) = \gcd(l_r,k_r) = 1$. Let $\tau(a,b) \triangleq \lceil \eta(a,b) \rceil - \eta(a,b)$, $\kappa \triangleq \mathrm{lcm}(k,k_r)$, and $\dot{\tau} = \frac{\{\kappa\eta(0,0)\}}{\kappa}$. If the sequence $\omega(p,r)$ is u.d. mod 1, then as $(n-t) \to \infty$ and $t \to \infty$,

$$f_n(u) = \frac{1}{\kappa} \sum_{\lambda=0}^{\kappa-1} \psi_{\dot{\tau}+\frac{\lambda}{\kappa}}(u). \tag{46}$$

Proof. According to Prop. VI.2 and (39), we can obtain

$$f_n(u) = \sum_{a=0}^{n-t} \sum_{b=0}^{t} p_A(a) p_B(b) \psi_{\tau(a,b)}(u). \tag{47}$$

According to Prop. V.1, as $(n-t) \to \infty$ and $t \to \infty$, $\tau(a,b)$ will be u.d. over $\{\dot{\tau}, \ldots, \dot{\tau} + \frac{\kappa-1}{\kappa}\}$. Hence, we can obtain (46). □

To get an intuitive impression of $f_n(u)$ in the birational case, the reader may refer to Fig. 7 in Sect. VIII-B. To facilitate the analysis, it is necessary to further simplify (46). As shown by (43), $\int_0^1 \psi_\tau(u)du = 1$ and $\psi_\tau(u)$ is symmetric around $u = 0.5$ (excluding $u = 0$ and $u = 1$). Thus $f_n(u)$ is also symmetric around $u = 0.5$ (excluding $u = 0$ and $u = 1$). Hence, we consider $f_n(u)$ only over $u \in [0.5, 1)$ below. In the birational case, according to (46), for $u \in [0.5, 1)$, $f_n(u)$ has κ jump points at $u \in \{2^{-\dot{\tau}}, 2^{-\dot{\tau}+1/\kappa}, \cdots, 2^{-\dot{\tau}+(\kappa-1)/\kappa}\}$, which divide the interval $[0.5, 1)$ into $(\kappa+1)$ segments given by $\mathcal{R}_j \triangleq [2^{\max(-1,-\dot{\tau}+(j-1)/\kappa)}, 2^{\min(0,-\dot{\tau}+j/\kappa)})$, where $0 \le j \le \kappa$. It is easy to show that for $2 \le j \le (\kappa-1)$, $|\mathcal{R}_j| = 2^{1/\kappa}|\mathcal{R}_{j-1}|$. According to (46), we have $f_n(u) = 2^{\dot{\tau}}\theta_\kappa(j)$ for $u \in \mathcal{R}_j$, where

$$\begin{aligned} \theta_\kappa(j) &\triangleq \frac{1}{\kappa}\left(2^{-1}\sum_{\alpha=0}^{j-1} 2^{-\alpha/\kappa} + \sum_{\alpha=j}^{\kappa-1} 2^{-\alpha/\kappa}\right) \\ &= \frac{2^{-j/\kappa}}{\kappa(2-2^{(\kappa-1)/\kappa})} = \frac{2^{-j/\kappa}}{2\kappa(1-2^{-1/\kappa})}. \end{aligned} \tag{48}$$

The following properties of $\theta_\kappa(j)$ can be easily proved:

- $\theta_\kappa(j)$ is strictly decreasing w.r.t. j, i.e., $\theta_\kappa(0) > \cdots > \theta_\kappa(\kappa)$;
- $\theta_\kappa(0) = 2\theta_\kappa(\kappa) = \frac{1}{2\kappa(1-2^{-1/\kappa})}$;
- $\theta_\kappa(j+1) = 2^{-1/\kappa}\theta_\kappa(j)$ and $\theta_\kappa(j+1) - \theta_\kappa(j) = -\frac{2^{-j/\kappa}}{2\kappa}$.

Uniform SMBS When $p = 0.5$, we have $\eta(a,b) \equiv nR$ and $\tau(a,b) \equiv \bar{\tau} \triangleq (\lceil nR \rceil - nR)$ for

all $(a,b) \in [0:(n-t)] \times [0:t]$, where $R = \frac{(n-t)r+t}{n}$. We have $\dot{\tau} = \bar{\tau}$ and $\kappa = 1$. Hence, $f_n(u) = \psi_{\bar{\tau}}(u)$. In particular, if $\bar{\tau} = 0$, then $f_n(u) = \Pi_{[0,1)}(u)$.

For uniform SMBS, in our prior work [21], [22], [23], [24], [25], only the special case $\bar{\tau} = 0$ is considered to simplify the analysis. This is a reasonable assumption in practice because the output of an (n,t,r) DAC encoder is almost fixed-length for any uniform SMBS (if we ignore the effect of underflow). This point can be explained in more detail as below.

If the uniform SMBS X is compressed by an (n,t,r) DAC encoder, then the length of output bitstream is $\lceil nR \rceil$ and the FBRL is $\bar{\tau} = \lceil nR \rceil - nR$, where $R = \frac{(n-t)r+t}{n}$. If $\bar{\tau} > 0$, then we can replace the (n,t,r) encoder with an (n,t,r') encoder, where $r' = \frac{\lceil nR \rceil - t}{n-t} > r$. With this new DAC encoder, the length of output bitstream is $\lceil nR' \rceil = nR' = \lceil nR \rceil$ and the FBRL is $\bar{\tau}' = (\lceil nR' \rceil - nR') = 0$, where $R' = \frac{(n-t)r'+t}{n} = \lceil nR \rceil / n > R$. It can be seen that, for the uniform SMBS, by using a different DAC encoder, the FBRL can always be removed, while the length of output bitstream remains unchanged.

D. Final CCS in Irrational Case

Corollary VI.1. In the case that $d \notin \mathbb{Q}$ and/or $rd \notin \mathbb{Q}$, if the sequence $\omega(p,r)$ is u.d. mod 1, then as $(n-t) \to \infty$ and $t \to \infty$, we have

$$f_n(u) = \frac{\log_2 e}{1 + |1 - 2u|}. \tag{49}$$

Proof. We take $f_n(u)$ in the irrational case as the asymptotic form of $f_n(u)$ in the birational case as $\kappa \to \infty$. According to Theorem VI.1, for any $u = 2^{-\dot{\tau}+j/\kappa} \in [0.5, 1)$,

$$\begin{aligned}
\lim_{\kappa\to\infty} f_n'(2^{-\dot{\tau}+j/\kappa}) &= \lim_{\kappa\to\infty} \frac{f_n(2^{-\dot{\tau}+(j+1)/\kappa}) - f_n(2^{-\dot{\tau}+j/\kappa})}{2^{-\dot{\tau}+(j+1)/\kappa} - 2^{-\dot{\tau}+j/\kappa}} \\
&= \lim_{\kappa\to\infty} \frac{2^{\dot{\tau}}(\theta_\kappa(j+1) - \theta_\kappa(j))}{2^{-\dot{\tau}+(j+1)/\kappa} - 2^{-\dot{\tau}+j/\kappa}} \\
&= -\lim_{\kappa\to\infty} \frac{1}{2\kappa(2^{1/\kappa}-1)} \cdot \lim_{\kappa\to\infty} \frac{1}{(2^{-\dot{\tau}+j/\kappa})^2} \\
&= -\frac{1}{2\ln 2} \cdot \lim_{\kappa\to\infty} \frac{1}{(2^{-\dot{\tau}+j/\kappa})^2}.
\end{aligned} \tag{50}$$

Hence, $\lim_{\kappa\to\infty} f_n'(u) = -\frac{1}{(2\ln 2)u^2}$ and further $\lim_{\kappa\to\infty} f_n(u) = \frac{1}{(2\ln 2)u}$. Finally, by symmetry, we can get the final CCS $f_n(u)$ in the irrational case as given by (49). □

$$\begin{array}{ccccccccccccc} U_0 & \leftarrow \cdots \leftarrow & U_{i-1} & \leftarrow & U_i & \leftarrow & U_{i+1} & \leftarrow \cdots \leftarrow & U_{n-1} & \leftarrow & U_n \\ \uparrow & \cdots & \uparrow & & \uparrow & & \uparrow & \cdots & \uparrow & & \\ X_1 & \cdots & X_i & & X_{i+1} & & X_{i+2} & \cdots & X_n & & \end{array}$$

Fig. 5. Backward-recursion deduction of U_i.

E. Discussion on General Cases

We have deduced the analytical form of the final CCS if the sequence $\omega(p,r)$ is u.d. mod 1. As shown by Prop. III.3, the sequence $\omega(p,r)$ is almost always u.d. mod 1, and the exceptional set has Lebesgue measure zero. Now we wonder what will happen if the sequence $\omega(p,r)$ is not u.d. mod 1? We guess that (46) and (49) still hold even if the sequence $\omega(p,r)$ is not u.d. mod 1. We cannot prove this point, but it is verified by a lot of experiments. For example, if we set p^{-1} to golden ratio $\frac{1+\sqrt{5}}{2}$ or silver ratio $(1+\sqrt{2})$, which are two exceptions of Prop. III.3, the final CCS obtained by experiments still coincides well with (46) or (49).

VII. Calculation of CCS

After knowing the final CCS $f_n(u)$, the remaining problem is how to get $f_i(u)$ for $i<n$. This section will give the recursive formula for deducing $f_i(u)$ backwards. Based on the recursive formula, a numerical algorithm is also proposed to calculate $f_i(u)$ in practice.

A. Backward Recursive Formula of CCS

Theorem VII.1. Let $u' \triangleq (u - x(1-p^{\gamma(i)}))(x\circ p)^{-\gamma(i)}$ and

$$f_{i-1}(u|x) \triangleq (x\circ p)^{-\gamma(i)} f_i(u'). \tag{51}$$

Once $f_n(u)$ is known, $f_i(u)$ for $i\in[0:n)$ can be recursively deduced backwards by

$$f_i(u) = (1-p)f_i(u|0) + pf_i(u|1). \tag{52}$$

Proof. By substituting (15) and (17) into (9), we can obtain

$$U_i = \frac{u(X^n) - \big(l(X^{i-1}) + X_i(1-p^{\gamma(i)})|\mathcal{I}(X^{i-1})|\big)}{(X_i\circ p)^{\gamma(i)}|\mathcal{I}(X^{i-1})|}, \tag{53}$$

which is followed by

$$\begin{aligned}(X_i \circ p)^{\gamma(i)} U_i &= \frac{u(X^n) - l(X^{i-1}) - X_i(1-p^{\gamma(i)})|\mathcal{I}(X^{i-1})|}{|\mathcal{I}(X^{i-1})|} \\ &= \frac{u(X^n) - l(X^{i-1})}{|\mathcal{I}(X^{i-1})|} - X_i(1-p^{\gamma(i)}) \\ &= U_{i-1} - X_i(1-p^{\gamma(i)}) \\ \Rightarrow U_{i-1} &= (X_i \circ p)^{\gamma(i)} \cdot U_i + X_i \cdot (1-p^{\gamma(i)}), \end{aligned} \tag{54}$$

showing that U_{i-1} depends on both X_i and U_i. Fig. 5 well illustrates how U_i can be calculated by backward recursion using (54). According to (54),

- If $X_i = 0$, then $U_{i-1} = (1-p)^{\gamma(i)} U_i$. Since $f_{i-1}(u|0)$ is the conditional PDF of U_{i-1} given $X_i = 0$, according to the property of PDF, we have

$$f_{i-1}(u|0) = (1-p)^{-\gamma(i)} f_i(u(1-p)^{-\gamma(i)}). \tag{55}$$

- If $X_i = 1$, then $U_{i-1} = p^{\gamma(i)} U_i + (1-p^{\gamma(i)})$. Since $f_{i-1}(u|1)$ is the conditional PDF of U_{i-1} given $X_i = 1$, according to the property of PDF, we have

$$f_{i-1}(u|1) = p^{-\gamma(i)} f_i((u-(1-p^{\gamma(i)}))p^{-\gamma(i)}). \tag{56}$$

The general form for (55) and (56) is just (51). Finally, we have

$$f_i(u) = \Pr(X_i = 0) f_i(u|0) + \Pr(X_i = 1) f_i(u|1). \tag{57}$$

By substituting $\Pr(X_i = 1) = p$ into (57), we get (52). □

B. Final Body CCS

We call $f_{n-t}(u)$ the *final body CCS*. It is interesting to know whether the asymptotic form of $f_{n-t}(u)$ as $t \to \infty$ exists. If it does exist, then we have the following corollary.

Corollary VII.1. As $t \to \infty$, we have $f_{n-t}(u) = \Pi_{[0,1)}(u)$.

Proof. According to (52), as $t \to \infty$,

$$f_{n-t}(u) = f_{n-t}(u(1-p)^{-1}) + f_{n-t}((u-(1-p))p^{-1}). \tag{58}$$

Now let $v = \frac{u}{1-p}$. Then (58) becomes

$$f_{n-t}(v(1-p)) = f_{n-t}(v) + f_{n-t}((1-p)(v-1)p^{-1}). \tag{59}$$

For all $v \in [0,1)$, $(v-1) < 0$ and thus $f_{n-t}((1-p)(v-1)p^{-1}) \equiv 0$. Hence, for all $v \in [0,1)$,

$$f_{n-t}(v) = f_{n-t}(v(1-p)) = f_{n-t}(v(1-p)^k), \tag{60}$$

where $k \in \mathbb{N}$. Since $\lim_{k\to\infty}(1-p)^k = 0$, we have $f_{n-t}(u) \equiv f_{n-t}(0)$ for all $u \in [0,1)$. Finally, by exploiting $\int_0^1 f_{n-t}(u)du = 1$, it is clear that $f_{n-t}(u) = \Pi_{[0,1)}(u)$ as $t \to \infty$. □

Immediately following Corol. VII.1, the initial CCS of the classic AC is $f_0(u) = \Pi_{[0,1)}(u)$ as $n \to \infty$ [21], [22], because the DAC degenerates into the classic AC when $t = n$.

C. Numerical Algorithm for Calculating CCS

In practice, it is more convenient to calculate $f_i(u)$ numerically using an algorithm similar to that proposed for uniform SMBS in [22]. To begin with, the interval $[0,1)$ is discretized into N equal-length cells. If N is sufficiently large, $f_i(u)$ for $u \in [0,1)$ can be approximated by $f_i(\frac{k}{N})$ for $k \in [0:N)$. For simplicity, $f_i(\frac{k}{N})$ will be abbreviated to $f_i(k)$ below. In a similar way, $f_i(k|0)$ and $f_i(k|1)$ are defined as the discretized $f_i(u|0)$ and $f_i(u|1)$, respectively. Then, the proposed numerical algorithm for calculating CCS has the following steps:

1) Initialization: Depending on the values of p and r, $f_n(k)$ is set in different way. In the birational case, $f_n(k)$ is set by (46); in the irrational case, $f_n(k)$ is set by (49).

2) Calculation of $f_i(k|0)$: We discretize (55) to calculate $f_i(k|0)$:

$$\tilde{f}_{i-1}(k|0) = (1-p)^{-\gamma(i)} f_i(\lfloor k(1-p)^{-\gamma(i)} \rceil). \tag{61}$$

where $\lfloor\cdot\rceil$ denotes the rounding operation. Note that a constraint must be imposed on k to guarantee $\lfloor k(1-p)^{-\gamma(i)} \rceil \le (N-1)$ in (61). For simplicity, we force $k(1-p)^{-\gamma(i)} \le (N-1)$, which results in $k \le (N-1)(1-p)^{\gamma(i)}$. Hence, the domain of k in (61) is $[0:H_0]$, where $H_0 = \lfloor (N-1)(1-p)^{\gamma(i)} \rfloor$. It is easy to verify

$$\lfloor k(1-p)^{-\gamma(i)} \rceil \le \lfloor H_0(1-p)^{-\gamma(i)} \rceil \le (N-1). \tag{62}$$

As for $k \in (H_0 : N)$, we have $\tilde{f}_i(k|0) = 0$. Since $f_i(k|0)$ is a discretized version of $f_i(u|0)$, $\sum_{k=0}^{N-1} f_i(k|0) = N$ must hold. Hence, $\tilde{f}_i(k|0)$ must be normalized as below

$$f_i(k|0) = \frac{N\tilde{f}_i(k|0)}{\sum_{k'=0}^{N-1} \tilde{f}_i(k'|0)}. \tag{63}$$

3) Calculation of $f_i(k|1)$: We discretize (56) to calculate $f_i(k|1)$:

$$\tilde{f}_{i-1}(k|1) = p^{-\gamma(i)} f_i(\lfloor (k - N(1-p^{\gamma(i)}))p^{-\gamma(i)} \rceil). \tag{64}$$

A constraint must be imposed on k to guarantee $\lfloor (k - N(1-p^{\gamma(i)}))p^{-\gamma(i)} \rceil \geq 0$ in (64). To simplify the analysis, we force $(k - N(1-p^{\gamma(i)})) \geq 0$, which is followed by $k \geq N(1-p^{\gamma(i)})$. Hence, the domain of k in (64) is $[L_1 : N)$, where $L_1 = \lceil N(1-p^{\gamma(i)}) \rceil$. It is easy to verify

$$\lfloor (k - N(1-p^{\gamma(i)}))p^{-\gamma(i)} \rceil \geq \lfloor (L_1 - N(1-p^{\gamma(i)}))p^{-\gamma(i)} \rceil \geq 0. \tag{65}$$

As for $k \in [0 : L_1)$, we have $\tilde{f}_i(k|1) = 0$. Again, $\sum_{k=0}^{N-1} f_i(k|1) = N$ must hold because $f_i(k|1)$ is a discretized version of $f_i(u|1)$. Thus, $\tilde{f}_i(k|1)$ must be normalized as below

$$f_i(k|1) = \frac{N\tilde{f}_i(k|1)}{\sum_{k'=0}^{N-1} \tilde{f}_i(k'|1)}. \tag{66}$$

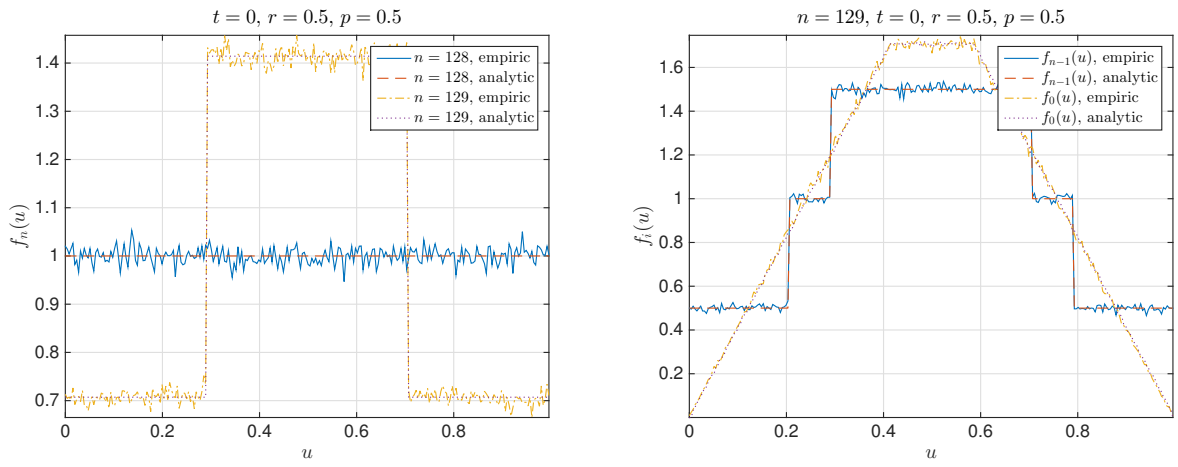
4) Calculation of $f_i(k)$: On knowing $f_i(k|0)$ and $f_i(k|1)$, we discretize (52) to get $f_i(k)$:

$$f_i(k) = (1-p)f_i(k|0) + pf_i(k|1). \tag{67}$$

Since $f_i(k|0)$ and $f_i(k|1)$ have been normalized in (63) and (66), $\sum_{k=0}^{N-1} f_i(k) = N$ holds always and thus there is no need to normalize $f_i(k)$.

VIII. Simulation Results

This section presents some simulation results to verify the above analyses. The analytical CCS is obtained by implementing the above-given equations and algorithms in MATLAB. The empirical CCS is obtained by the same method as used in [22]. First, a number of source blocks are generated according to a specific probability distribution, each of which is encoded and then decoded along the proper path by a DAC codec. During the decoding, the bitstream self-projection is calculated and discretized. Finally, the discretized bitstream self-projection of each specific level is counted to obtain the corresponding CCS. A 32-bit DAC codec is used in the



(a) (b)

Fig. 6. Examples to illustrate the impact of FBRL $\bar{\tau}$ on the CCS of uniform SMBS, where *empiric* stands for the CCS obtained by a practical DAC codec. (a) The final CCS, where *analytic* stands for the CCS obtained by (46). (b) Evolution of CCS, where *analytic* stands for the CCS obtained by (52).

following simulations and 2^{20} blocks are generated for each CCS. For both analytical CCS and empirical CCS, the interval $[0, 1)$ is discretized into $2^8 = 256$ cells.

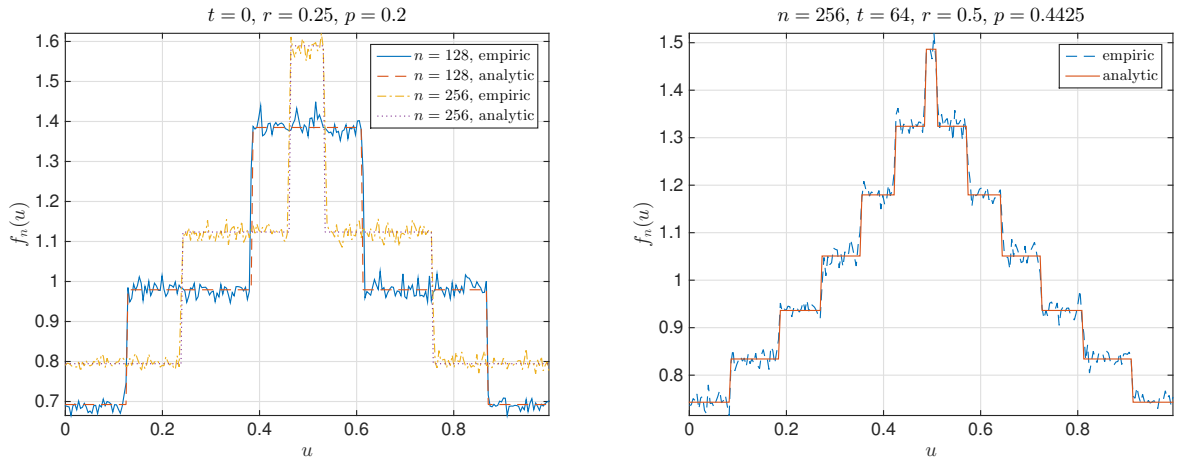
A. CCS of Uniform SMBS

Before studying the CCS of biased SMBS, let us first show the impact of FBRL $\bar{\tau}$ on the CCS of uniform SMBS. Fig. 6(a) verifies the correctness of (46). We fix $t = 0$ and $r = 0.5$. Two code lengths 128 and 129 are tested. For $n = 128$, $\bar{\tau} = 0$ and thus the final CCS $f_n(u) = \Pi_{[0,1)}(u)$; for $n = 129$, $\bar{\tau} = 0.5$ and hence there are two jump points at $u = 1 - 2^{-\bar{\tau}} \approx 0.3$ and $u = 2^{-\bar{\tau}} \approx 0.7$. Fig. 6(b) verifies the correctness of (52) for $n = 129$. We select two typical levels of CCS, $f_{n-1}(u)$ and $f_0(u)$. It can be seen that the analytical results coincide well with the empirical results. By comparing Fig. 6(b) with the results in [22], we can find that as $n \to \infty$, the initial CCS $f_0(u)$ when $\bar{\tau} > 0$ is the same as that when $\bar{\tau} = 0$.

B. Final CCS of Biased SMBS—Birational Case

Fig. 7 gives some examples to verify the analyses on the final CCS of biased SMBS in the birational case, i.e., $d = \log_2 \frac{1-p}{p} \in \mathbb{Q}$ and $rd \in \mathbb{Q}$.

Fig. 7(a) verifies the correctness of (46) for tailless DAC. Since $b = |x_{n-t+1}^n| \equiv 0$ for tailless DAC, we shorten $\eta(a, b)$ to $\eta(a)$ and $\tau(a, b)$ to $\tau(a)$. We set $r = 0.25$ and $p = 0.2$, so $d = 2$ and



(a) (b)

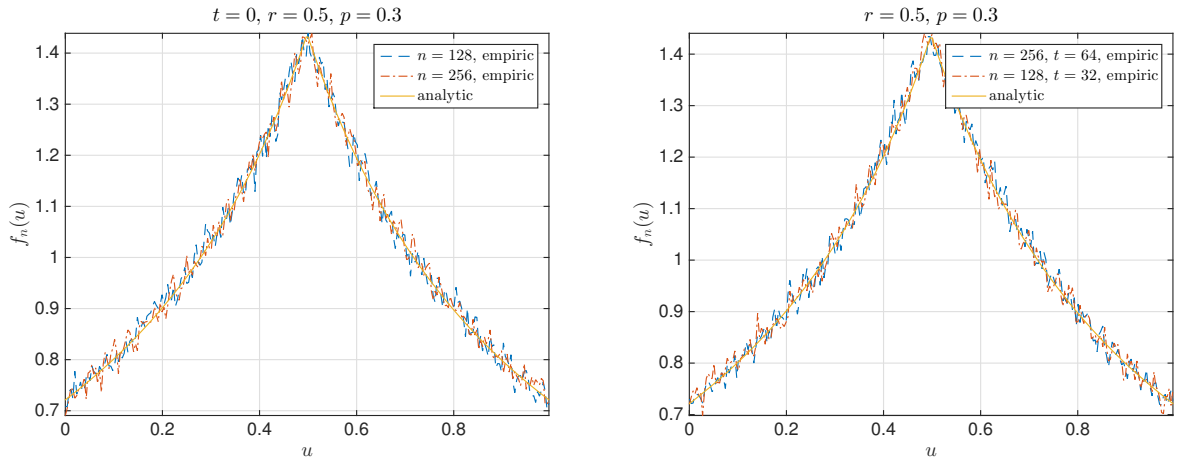
Fig. 7. Examples of the final CCS of biased SMBS when $d = \log_2 \frac{1-p}{p} \in \mathbb{Q}$ and $rd \in \mathbb{Q}$, where *empiric* stands for the CCS obtained by a practical DAC codec and *analytic* stands for the CCS obtained by (46). (a) Tailless DAC. (b) Tailed DAC.

$rd = \frac{1}{2}$, resulting in $\kappa = 2$ and $\tau(a) \in \{\dot{\tau}, \dot{\tau} + \frac{1}{2}\}$. Two code lengths are tested. For $n = 128$, $\eta(0) \approx 10.30$, so $\dot{\tau} + \frac{1}{2} = \tau(0) \approx 0.7$ and $\dot{\tau} \approx 0.2$. Hence in $[0, 0.5)$, there are two jump points at $u \approx (1-2^{-0.2}) \approx 0.1294$ and $(1-2^{-0.7}) \approx 0.3844$. For $n = 256$, $\eta(0) \approx 20.6$, so $\dot{\tau} = \tau(0) \approx 0.4$ and $\dot{\tau} + \frac{1}{2} \approx 0.9$. Hence in $[0, 0.5)$, there are two jump points at $u \approx (1 - 2^{-0.4}) \approx 0.2421$ and $(1-2^{-0.9}) \approx 0.4641$. By symmetry, we can get two other jump points in $[0.5, 1)$. These deductions are confirmed by Fig. 7(a).

Fig. 7(b) verifies the correctness of (46) for tailed DAC. We set $r = 0.5$ and $p \approx 0.4425$, so $d = \frac{1}{3}$ and $rd = \frac{1}{6}$, resulting in $\kappa = 6$ and $\tau(a, b) \in \{\dot{\tau}, \dot{\tau} + \frac{1}{6}, \cdots, \dot{\tau} + \frac{5}{6}\}$. In addition, we set $n = 256$ and $t = 64$, so $\eta(0, 0) \approx 134.873$ and $\dot{\tau} = \tau(0, 0) \approx 0.127$. Thus in $[0, 0.5)$, there are six jump points at $u \approx 0.0843$, 0.1842, 0.2732, 0.3525, 0.4231, and 0.4861. By symmetry, we can get six other jump points in $[0.5, 1)$. These deductions are confirmed by Fig. 7(b).

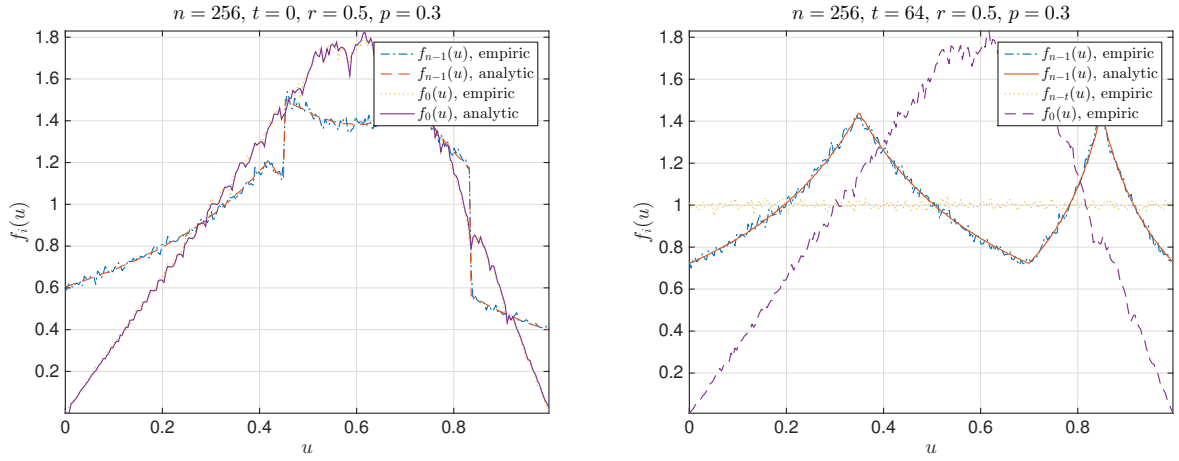
C. Final CCS of Biased SMBS—Irrational Case

Fig. 8 gives some examples to verify the analyses on the final CCS of biased SMBS in the irrational case, i.e., $d = \log_2 \frac{1-p}{p} \notin \mathbb{Q}$ and/or $rd \notin \mathbb{Q}$. We set $r = 0.5$ and $p = 0.3$. Figs. 8(a) and 8(b) present the results for tailless and tailed DAC, respectively. In Fig. 8(a), two code lengths 256 and 128 are tested, while in Fig. 8(b), two combinations of n and t are tested: $(n, t) = (256, 64)$ and $(128, 32)$. It can be seen that the correctness of (49) is well confirmed for both tailless and tailed DAC.



(a) (b)

Fig. 8. Examples of the final CCS of biased SMBS when $d = \log_2 \frac{1-p}{p} \notin \mathbb{Q}$ or $rd \notin \mathbb{Q}$, where *empiric* stands for the CCS obtained by a practical DAC codec and *analytic* stands for the CCS obtained by (49). (a) Tailless DAC. (b) Tailed DAC.



(a) (b)

Fig. 9. Examples to demonstrate the backward evolution of CCS for biased SMBS, where *empiric* stands for the CCS obtained by a practical DAC codec and *analytic* stands for the CCS obtained by (52). (a) Tailless DAC. (b) Tailed DAC.

D. CCS Evolution of Biased SMBS

Fig. 9 gives some examples to verify the analyses on the backward evolution of CCS, where $n = 256$, $r = 0.5$, and $p = 0.3$. Fig. 9(a) confirms the correctness of (52) for tailless DAC, where two typical levels of CCS, i.e., $f_{n-1}(u)$ and $f_0(u)$, are presented. Fig. 9(b) confirms the correctness of (52) for tailed DAC with $t = 64$, where three typical levels of CCS, i.e., $f_{n-1}(u)$, $f_{n-t}(u)$, and $f_0(u)$, are presented. As shown by Fig. 9(b), the final body CCS $f_{n-t}(u)$ does tend

to be uniform over $[0,1)$ as $t \to \infty$. By comparing Fig. 9(a) with Fig. 9(b), we conclude that: Though $f_{n-1}(u)$ of tailed DAC is quite different from that of tailless DAC, the initial CCS $f_0(u)$ of tailed DAC as $(n-t) \to \infty$ is the same as that of tailless DAC as $n \to \infty$.

IX. Conclusion

This paper extends our prior work on CCS from uniform SMBS to biased SMBS. Both tailless DAC and tailed DAC are investigated in this paper. An important finding is that in contrast to uniform SMBS, for biased SMBS, the final CCS cannot be converted into the uniform function over $[0,1)$ by carefully tuning coding parameters. We give the analytical form of the final CCS for both birational and irrational cases, and deduce the recursive formula for the backward evolution of CCS. A numerical algorithm is also proposed for calculating CCS effectively in practice. All analyses are verified by simulation results. How to apply the CCS of biased SMBS to decoder design will be our next work in the future.

The results in this paper can be reproduced by running the source codes released in [28].

References

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[2] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 9, pp. 80–94, Sep. 2004.

[3] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[4] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[5] E. A. Bilkent, "Polar coding for the Slepian-Wolf problem based on monotone chain rules," in: *Proc. International Symposium on Information Theory*, IEEE, pp. 566–570, 2012.

[6] J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM J. Research & Development*, vol. 20, no. 3, pp. 198–203, May 1976.

[7] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Commun. of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[8] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding," *IEEE Commun. Lett.*, vol. 11, no. 11, pp. 883–885, Nov. 2007.

[9] X. Artigas, S. Malinowski, C. Guillemot, and L. Torres, "Overlapped quasi-arithmetic codes for distributed video coding," in: *Proc. IEEE ICIP*, 2007, vol. II, pp. 9–12.

[10] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding for the Slepian-Wolf problem," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2245–2257, Jun. 2009.

[11] S. Malinowski, X. Artigas, C. Guillemot, and L. Torres, "Distributed coding using punctured quasi-arithmetic codes for memory and memoryless sources," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4154–4158, Oct. 2009.

[12] X. Chen and D. Taubman, "Coupled distributed arithmetic coding," in: *Proc. IEEE ICIP*, pp. 341–344, Sep. 2011.

[13] J. Zhou, K. Wong, and J. Chen, "Distributed block arithmetic coding for equiprobable sources," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2750–2756, Jul. 2013.

[14] M. Grangetto, E. Magli, R. Tron, and G. Olmo, "Rate-compatible distributed arithmetic coding," *IEEE Commun. Lett.*, vol. 12, no. 8, pp. 575–577, Aug. 2008.

[15] M. Grangetto, E. Magli, and G. Olmo, "Decoder-driven adaptive distributed arithmetic coding," in: *Proc. IEEE ICIP*, 2008, pp. 1128–1131.

[16] X. Chen and D. Taubman, "Distributed source coding based on punctured conditional arithmetic codes," in: *Proc. IEEE ICIP*, pp. 3713–3716, Sep. 2010.

[17] M. Grangetto, E. Magli, and G. Olmo, "Distributed joint source-channel arithmetic coding," in: *Proc. IEEE ICIP*, 2010, pp. 3717–3720.

[18] Y. Keshtkarjahromi, M. Valipour, and F. Lahouti, "Multi-level distributed arithmetic coding with nested lattice quantization," in: *Proc. Data Compression Conference*, pp. 382–391, Mar. 26–28, 2014.

[19] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[20] Z. Wang, Y. Mao, and I. Kiringa, "Non-binary distributed arithmetic coding," in: *Proc. IEEE 14th Canadian Workshop Inform. Theory* (CWIT), pp. 5–8, Jul. 2015.

[21] Y. Fang, "Distribution of distributed arithmetic codewords for equiprobable binary sources," *IEEE Signal Process. Lett.*, vol. 16, no. 12, pp. 1079–1082, Dec. 2009.

[22] Y. Fang, "DAC spectrum of binary sources with equally-likely symbols," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1584–1594, Apr. 2013.

[23] Y. Fang and L. Chen, "Improved binary DAC codec with spectrum for equiprobable sources," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 256–268, Jan. 2014.

[24] Y. Fang, V. Stankovic, S. Cheng, and E.-H. Yang, "Hamming distance spectrum of DAC codes for equiprobable binary sources," *IEEE Trans. Commun.*, vol. 64, no. 3, pp. 1232–1245, Mar. 2016.

[25] Y. Fang, V. Stankovic, S. Cheng, and E.-H. Yang, "Analysis on tailed distributed arithmetic codes for uniform binary sources," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4305–4319, Oct. 2016.

[26] Y. Liu and Y. Fang, "Codebook cardinality spectrum of distributed arithmetic codes for nonuniform binary sources," in: *Proc. Chinese Conference on Computer Vision* (CCCV), Xi'an China, Sep. 2015, Part II, pp. 458–467.

[27] L. Kuipers and H. Niederreiter, "Uniform Distribution of Sequences," *John Wiley & Sons*, 1974.

[28] http://js.chd.edu.cn/xxgcxy/fy_en/.