Low-Complexity Approaches to Slepian–Wolf Near-Lossless Distributed Data Compression

Todd P. Coleman, *Member, IEEE*, Anna H. Lee, *Student Member, IEEE*, Muriel Médard, *Senior Member, IEEE*, and Michelle Effros, *Senior Member, IEEE*

***Abstract*—This paper discusses the Slepian–Wolf problem of distributed near-lossless compression of correlated sources. We introduce practical new tools for communicating at *all* rates in the achievable region. The technique employs a simple "source-splitting" strategy that does not require common sources of randomness at the encoders and decoders. This approach allows for pipelined encoding and decoding so that the system operates with the complexity of a single user encoder and decoder. Moreover, when this splitting approach is used in conjunction with iterative decoding methods, it produces a significant simplification of the decoding process. We demonstrate this approach for synthetically generated data. Finally, we consider the Slepian–Wolf problem when linear codes are used as syndrome-formers and consider a linear programming relaxation to maximum-likelihood (ML) sequence decoding. We note that the fractional vertices of the relaxed polytope compete with the optimal solution in a manner analogous to that observed when the "min-sum" iterative decoding algorithm is applied. This relaxation exhibits the ML-certificate property: if an integral solution is found, it is the ML solution. For symmetric binary joint distributions, we show that selecting easily constructable "expander"-style low-density parity check codes (LDPCs) as syndrome-formers admits a positive error exponent and therefore provably good performance.**

***Index Terms*—Block codes, communication systems, data compression, decoding, iterative methods.**

I. Introduction

THE Slepian–Wolf problem of distributed near-lossless compression of correlated sources (see (L) of Fig. 1) has been understood theoretically for many years [1]. It has received a lot of attention recently due to its relevance as a subcomponent of numerous distributed data dissemination systems. Practical techniques, however, have remained elusive for quite a long time. The challenges include: finding provably good codes, low-complexity decoding, and choosing source coding rates. Recently, proper application of channel coding developments to this setting has been successful at addressing some of these challenges. However, explicit practical solutions that apply to all instantiations of the problem have not yet been constructed. This paper applies channel coding developments to broaden the class of problems with low complexity solutions. Indeed, any instance of the problem can be addressed practically with our approach.

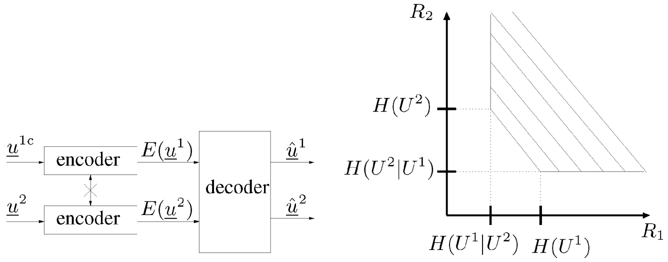

Fig. 1. The Slepian–Wolf problem: (L) model (R) achievable rate region.

The achievable rate region $\mathcal{R}\left[P\left(u^1,\dots,u^M\right)\right]$ for M memoryless sources $(U^1,\dots,U^M)$ with joint probability distribution $P\left(u^1,\dots,u^M\right)$ is given by [1]

$$\mathcal{R}=\Big\{\underline{R}\in\mathbb{R}_+^M: \sum_{i\in S}R_i\geq H\left(U(S)|U(S^c)\right)\ \forall\, S\subseteq\{1,2,\dots,M\}\Big\} \tag{1}$$

where $U(S)=\left\{U^j\right\}_{j\in S}$. (See (R) of Fig. 1.) In [2], Cover simplified the proof by proposing a code design strategy whereby each encoder randomly places all possible source sequences into bins and gives the bin index to the decoder. Linear block codes can be used to perform binning practically and with no loss in either the achievable rate region or the error exponent [3]. In code operation, the decoder receives a single bin index from each transmitter and then searches for a collection $\left(\hat{\underline{U}}^1,\dots,\hat{\underline{U}}^M\right)$ of "jointly typical" sequences [4, pp. 194–197] lying in the described bins. This can be done with high probability provided that the rates lie within the achievable region. At certain rate points, which we call "vertices" or "corner points," this joint search over all codebooks for "jointly typical" sequences can be done successively. The corner points are the rate tuples $(R_1,\dots,R_M)$ that are obtained by expanding $H(U^1,\dots,U^M)$ by M successive applications of the chain rule and assigning to each rate the unique corresponding term

Manuscript received October 5, 2004; revised March 30, 2006. This work was supported in part by an NSF Graduate Research Fellowship under NSF Grant CCR-0220039, the National Reconnaissance Office, and Caltech's Lee Center for Advanced Networking. The material presented in this paper was presented in part at the IEEE Data Compression Conference, Snowbird, UT, March 2004, the IEEE International Symposium on Information Theory, Chicago, IL, July 2004, and the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, March 2005.

T. P. Coleman is with the Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA (e-mail: colemant@uiuc.edu).

A. H. Lee and M. Médard are with the Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: annalee@mit.edu; medard@mit.edu).

M. Effros is with the Data Compression Laboratory, Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: effros@caltech.edu).

Communicated by V. A. Vaishampayan, Associate Editor At Large.

Digital Object Identifier 10.1109/TIT.2006.878215

in the expansion. For instance, if users would like to communicate at the rate $(R_1, R_2) = \left(H(U^1), H(U^2|U^1)\right)$, then we describe the source U^1 at rate $H(U^1)$ by entropy-encoding $\underline{U}^1$. (We can do this by using either a variable-rate lossless code or a fixed-rate near-lossless code.) After successful decoding, U^1 can be used as side information to help decode U^2 at rate $H(U^2|U^1)$. By exchanging the roles of U^1 and U^2, it follows that the same approach applies to encoding at rate $(R_1, R_2) = \left(H(U^1|U^2), H(U^2)\right)$. Thus, in this case, the decoding process can be decomposed into a pipelined approach that operates at the speed of a single-user decoder. Recently, a lot of attention has been paid to the construction of low-complexity decoders to achieve rates of R_2 very close to $H(U^2|U^1)$. These attempts, which include iterative techniques for turbo-code [5] constructions [6]–[9] and low-density parity check code (LDPC) [10] constructions [11]–[15], have met much success when U^1 and U^2 are binary random variables.

While these codes can be combined using time-sharing to achieve nonvertex rates, time-sharing has practical drawbacks. Rate fluctuations arise at different points of the encoding process, and the delay required to communicate near a target rate can be prohibitively long.

We consider in Section III a practical method to perform "*source-splitting*," which transforms all points in the Slepian–Wolf achievable region into vertices in a Slepian–Wolf achievable region with more sources. Once the rate point becomes a vertex, we can parallelize encoding and pipeline decoding. Inspired by rate-splitting for Gaussian [16] and discrete memoryless [17] multiple access channels, source-splitting was introduced in [18], but that approach required shared randomness at the encoders and decoder, and the outputs of the splitting operation had alphabets larger than the original source. Another approach that allows parallelized encoding and pipelined decoding is [19], but this also requires common randomness at the encoder and decoder *and* involves searching for jointly typical sequences at the *encoder*. Our splitting technique involves a simple thresholding operation followed by specifying a bin index, reduces the alphabet size of the outputs of the splitter, and does not require common randomness.

In Sections III-A and III-B we also illustrate via the "method of types" [20] and reasoning similar to [17] that performing the proposed splitting strategy at most once per user can achieve any rate in the Slepian–Wolf achievable rate region with parallelized encoding and pipelined decoding. Analogous to Section III of [21], we show in Section III-C that rate tuples on the boundary of the dominant face can be split into two sets of sources that may be decoded sequentially.

We discuss in Section IV how the splitting strategy can be combined with iterative decoding in a practical setting. Our splitting technique has an important simplification in part of the decoding process. Simulation results from synthetically generated data confirm the practicality and effectiveness of this approach.

We also consider in Section V the Slepian–Wolf problem when LDPCs are used as syndrome-formers and consider a linear programming (LP) relaxation to maximum-likelihood sequence decoding (MLSD). This decoder exhibits the maximum-likelihood (ML)-certificate property: if an integral solution is found, it is the ML solution. We note that the fractional vertices of the relaxed polytope, termed *pseudocodewords*, compete with the ML solution in a manner analogous to that observed when a "min-sum" iterative decoding algorithm is applied [22]–[25]. We show how this relaxation relates to "coset-leader" decoding across binary symmetric channels. From there, we show an equivalence between this LP formulation and one developed for channel coding [26]–[28]. This equivalence allows us to illustrate that for symmetric binary joint distributions, Slepian–Wolf vertex rates can be achieved using easily constructable 'expander'-style LDPC's [29]–[31] as syndrome-formers with a positive error exponent (i.e., exponential error probability decay in block length).

II. Model and Definitions

In this paper, we will consider a set of M discrete memoryless sources $U^1, U^2, \ldots, U^M$ drawn according to $P\left(u^1, u^2, \ldots, u^M\right)$ with alphabets $\mathcal{U}^1, \mathcal{U}^2, \ldots, \mathcal{U}^M$. We denote U^i_j as the jth symbol from process U^i. We use the following notation:

$$\begin{aligned}
[r] &\triangleq \{1, 2, \ldots, r\} \\
R(\mathcal{S}) &\triangleq \sum_{i \in \mathcal{S}} R_i \text{ for any } \mathcal{S} \subseteq \mathbb{Z}, \text{ where } R_i \in \mathbb{R}_+ \\
\underline{U}^{\mathcal{S}} &\triangleq \left(U^i\right)_{i \in \mathcal{S}} \text{ for any } \mathcal{S} \subseteq \mathbb{Z} \\
\underline{U}_{\mathcal{S}} &\triangleq \left(U_j\right)_{j \in \mathcal{S}} \text{ for any } \mathcal{S} \subseteq \mathbb{Z} \\
\mathcal{S}^c &\triangleq [M] \backslash \mathcal{S} \\
\Pi(\mathcal{U}) &= \{\pi | \pi \text{ permutes } \mathcal{U}\} \\
H(U) &\triangleq \sum_{a \in \mathcal{U}} -P_U(a) \log_2 \left(P_U(a)\right) \\
\mathcal{H}(U) &\triangleq \lim_{n \to \infty} \frac{1}{n} H\left(\underline{U}_{[n]}\right) \\
D(P \| Q) &\triangleq \sum_{a \in \mathcal{U}} P(a) \log_2 \left(\frac{P(a)}{Q(a)}\right)
\end{aligned}$$

A. Dominant Face

The *dominant face* $\mathcal{D}\left[\mathcal{R}\left[P(\underline{u}^{[M]})\right]\right]$ consists of all $R \in \mathcal{R}\left[P(\underline{u}^{[M]})\right]$ that satisfy

$$R([M]) = H(\underline{U}^{[M]}). \tag{2}$$

Note that any point in $\mathcal{R}$ is dominated (with respect to the standard partial order on $\mathbb{R}^M_+$) by a point in the dominant face.

Throughout the paper, we exploit the chain rule for entropy

$$H(\underline{U}^{\mathcal{T}}) = H(\underline{U}^{\mathcal{S}}) + H(\underline{U}_{\mathcal{T} \backslash \mathcal{S}} | \underline{U}^{\mathcal{S}}) \quad \forall \mathcal{S} \subseteq \mathcal{T} \subseteq [M]. \tag{3}$$

We may now apply the chain rule to derive an alternative description of the dominant face $\mathcal{D}$. By combining the chain rule with (1) and (2), we arrive at

$$\begin{aligned}
R(\mathcal{S}) &= R([M]) - R(\mathcal{S}^c) \\
&= H(\underline{U}^{[M]}) - R(\mathcal{S}^c) \\
&\leq H(\underline{U}^{[M]}) - H(\underline{U}^{\mathcal{S}^c} | \underline{U}^{\mathcal{S}}) \\
&= H(\underline{U}^{\mathcal{S}}).
\end{aligned}$$

So we see that achievability (1) and lying on the dominant face (2) imply that

$$H(\underline{U}^{\mathcal{S}}|\underline{U}^{\mathcal{S}^c}) \leq R(\mathcal{S}) \leq H(\underline{U}^{\mathcal{S}}) \quad \forall \mathcal{S} \subseteq [M]. \tag{4}$$

Conversely, we see that the leftmost inequality in (4) directly implies achievability (1) and setting $\mathcal{S} = [M]$ in (4) directly implies lying on the dominant face (2). Hence, we may alternatively characterize the dominant face as

$$\mathcal{D} = \Big\{ R \in \mathbb{R}_+^M : \\ H(\underline{U}^{\mathcal{S}}|\underline{U}^{\mathcal{S}^c}) \leq R(\mathcal{S}) \leq H(\underline{U}^{\mathcal{S}}) \, \forall \mathcal{S} \subseteq [M] \Big\}. \tag{5}$$

Vertices are the rate tuples $\underline{R}_{[M]} \in \mathcal{D}$ that occur at the intersection of the bounding surfaces (for instance, they are the two "corner points" of Fig. 1). They are obtained by expanding $H(\underline{U}^{[M]})$ into M terms by $M-1$ successive applications of the chain rule, and assigning to R_i the value of the unique term in the expansion having the form $H(U^i|\underline{U}^{\mathcal{S}})$ for some set $\mathcal{S} \subseteq [M]$. Each unique vertex of the dominant face corresponds to a rate-tuple that is single-user decodable given side information of the previously decoded sources. Most of the practical methods [6]–[9], [11]–[15] to achieve rates near the Slepian–Wolf achievable rate region boundary are only applicable to vertices.

III. Source-Splitting for Slepian–Wolf

Let us now consider taking each symbol of a discrete memoryless source (DMS) $U = (U_1, U_2 \ldots)$ where $U_i \in \mathcal{U} = \{0, \ldots, |\mathcal{U}|-1\}$ and splitting it into a collection of random variables of smaller cardinality. We write $U_i \leftrightarrow (U_i^a, U_i^b)$ if there is a bijection between the random variables U_i and (U_i^a, U_i^b). We consider the following way to perform source-splitting:

$$U_i \mapsto \begin{pmatrix} U_i^a = \min(\pi(U_i), T) \\ U_i^b = \max(\pi(U_i), T) - T \end{pmatrix} \tag{6a}$$

$$\mapsto U_i = \pi^{-1}\left(U_i^a + U_i^b\right) \tag{6b}$$

where $T \in \mathcal{U}$ operates as a thresholder and $\pi \in \Pi(\mathcal{U})$ is a permutation operator.

Definition (6) gives many possible splits, since there are many possible values of $\pi \in \Pi(\mathcal{U})$ and $T \in \mathcal{U}$. For a nontrivial splitting threshold $(T \in \mathcal{U} \setminus \{0, |\mathcal{U}|-1\})$, $U_i^a \in \{0, \ldots, T\}$, $U_i^b \in \{0, \ldots, |\mathcal{U}|-1-T\}$, and there are $\binom{|\mathcal{U}|}{T}$ distinct ways to map the $|\mathcal{U}|$ symbols to the splitting sets in (6). This provides a total of

$$\begin{aligned} \sum_{i=1}^{|\mathcal{U}|-2} \binom{|\mathcal{U}|}{i} &= 2^{|\mathcal{U}|} - \binom{|\mathcal{U}|}{0} - \binom{|\mathcal{U}|}{|\mathcal{U}|-1} - \binom{|\mathcal{U}|}{|\mathcal{U}|} \\ &= 2^{|\mathcal{U}|} - |\mathcal{U}| - 2 \\ &= O(2^{|\mathcal{U}|}) \end{aligned}$$

distinct ways to perform the splitting mechanism and form the bijection $U_i \leftrightarrow (U_i^a, U_i^b)$.

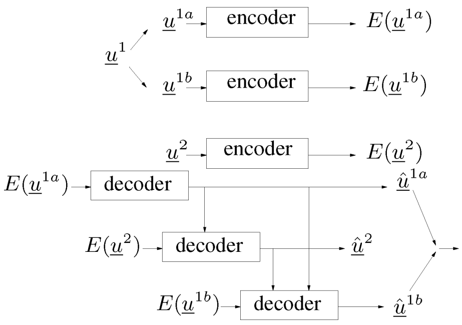

Fig. 2. Source splitting and decoding for a two-source Slepian–Wolf problem.

If we have two discrete memoryless sources (U^1, U^2) drawn according to $P\left(u^1, u^2\right)$, then we can split U^1 to form (U^{1a}, U^{1b}) as shown in (6). At this point, we have three sources, each of which can be encoded separately at rates R_{1a}, R_{1b}, R_2. We note that because $U \leftrightarrow (U^{1a}, U^{1b})$, $H\left(U^1, U^2\right) = H\left(U^{1a}, U^{1b}, U^2\right)$. Through the chain rule for entropy, we consider the rates

$$R_{1a} = H\left(U^{1a}\right) \tag{7a}$$

$$R_2 = H\left(U^2|U^{1a}\right) \tag{7b}$$

$$R_{1b} = H\left(U^{1b}|U^2, U^{1a}\right) \tag{7c}$$

$$R_1 = R_{1a} + R_{1b}. \tag{7d}$$

For any nontrivial split, (R_1, R_2) is not a vertex in $\mathcal{R}\left[P\left(u^1, u^2\right)\right]$, but (R_{1a}, R_2, R_{1b}) *is* a vertex in $\mathcal{R}\left[P\left(u^{1a}, u^2, u^{1b}\right)\right]$. This directly implies a parallelizable encoding strategy and pipelined single-user decoding strategy that operates with the complexity of a smaller-alphabet decoder. By varying across the different values of the threshold $T \in \mathcal{U}$ and $\pi \in \Pi(\mathcal{U})$, we may sweep across $O(2^{|\mathcal{U}|})$ distinct nonvertex points on the dominant face $\mathcal{D}\left[\mathcal{R}\left[P\left(u^1, u^2\right)\right]\right]$. Fig. 2 illustrates the proposed encoding and decoding strategy.

Source-splitting may be performed to transform a source U of cardinality $|\mathcal{U}|$ into $|\mathcal{U}|-1$ binary random variables

$$U_i \mapsto \begin{pmatrix} U_i^1 = 1_{\{\pi(U_i)=1\}} \\ U_i^2 = 1_{\{\pi(U_i)=2\}} \\ \vdots \\ U_i^{|\mathcal{U}|-1} = 1_{\{\pi(U_i)=|\mathcal{U}|-1\}} \end{pmatrix} \tag{8a}$$

$$\mapsto U_i = \pi^{-1}\left(\sum_{k=1}^{|\mathcal{U}|-1} k U_i^k\right) \tag{8b}$$

where $\pi \in \Pi(\mathcal{U})$ and $1_{\{A\}} = 1$ if event A occurs and 0 otherwise. Each $\pi \in \Pi(\mathcal{U})$ yields new splits and thus there are $|\mathcal{U}|!$ splits.

The motivation for binary splitting is the reduction in complexity of near-lossless block-compression of high-rate sources: the splitting approach allows for parallelized encoding and pipelined single-user decoding of low-rate binary sources.

In the next section we show that although this method generates a *finite* number of distinct splits, we may group consecutive symbols together and interpret them as a single outcome of a source of larger alphabet. Because of the exponential growth in the number of splits as a function of the source alphabet size, it follows that long super-symbols lengths are not required. We also discuss in the next section a controlled way to map super-symbols to a desired rate point. Moreover we arrive at similar details about the required number of splits per source, as in the case of multiple access [18].

A. Two Sources: At Most One Split Per Source Required

We consider a DMS U drawn according to pmf Q over alphabet $\mathcal{U} = \{0, 1, \dots, |\mathcal{U}| - 1\}$ and assume without loss of generality that $Q(a) > 0$ for each $a \in \mathcal{U}$. We treat the first n outcomes $\underline{U}_{[n]}$ of the source U as the single outcome of a DMS with alphabet $\{0, \dots, |\mathcal{U}|^n - 1\}$ through the standard integral representation

$$\mathrm{sr}\left(\underline{u}_{[n]}\right) = \sum_{j=1}^{n} u_j |\mathcal{U}|^{j-1}. \tag{9}$$

Splitting $\mathrm{sr}\left(\underline{U}_{[n]}\right)$ according to (6) on $\mathrm{sr}\left(\underline{U}_{[n]}\right)$ yields $(\underline{U}^a_{[n]}, \underline{U}^b_{[n]})$ and a total of $2^{|\mathcal{U}|^n} - |\mathcal{U}|^n - 2 = O(2^{|\mathcal{U}|^n})$ nontrivial splits. We use the "method of types" [20] to take a subset of all $\pi \in \Pi(\mathrm{sr}\,(\mathcal{U}^n))$ and $T \in \{0, \dots, |\mathcal{U}|^n - 1\}$, parametrize them according to $\epsilon \in [0, 1]$, and demonstrate that $P_{\underline{U}^a_{[n]}(\epsilon)}(\cdot)$ tends to a continuous function of ϵ and $\frac{1}{n} H\left(\underline{U}^a_{[n]}(\epsilon)\right)$ tends to $\epsilon H(U)$. Moreover, we illustrate in Theorem 3.5 that any point on the dominant face of the two-user Slepian–Wolf achievable rate region can be transformed to a vertex in a three-user problem via source-splitting. Since the number of nontrivial splits grows as $O(2^{|\mathcal{U}|^n})$, operating near any target rate does not require long super-symbol lengths. We introduce some intermediate lemmas that are useful in the proof of Theorem 3.5.

We denote the set of all probability distributions on $\mathcal{U}$ by $\mathcal{P}(\mathcal{U})$. For a length-n sequence $\underline{u} = (u_1, u_2, \dots, u_n) \in \mathcal{U}^n$, the type $P_{\underline{u}} \in \mathcal{P}(\mathcal{U})$ is the probability distribution defined by $P_{\underline{u}}(a) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{u_i = a\}}$, for all $a \in \mathcal{U}$. We denote by Q^n the pmf induced on $\mathcal{U}^n$ by n independent drawings according to Q. We denote by $\mathcal{P}_n(\mathcal{U}) = \{P^{0,n}, P^{1,n}, \dots\}$ the subset of $\mathcal{P}(\mathcal{U})$ consisting of the possible types of sequences $\underline{u} \in \mathcal{U}^n$. For any type $P^{j,n} \in \mathcal{P}_n(\mathcal{U})$, the type class $T\left(P^{j,n}\right)$ is the set of all $\underline{u} \in \mathcal{U}^n$ such that $P_{\underline{u}} = P^{j,n}$. From [20] we note that

$$\begin{aligned} |\mathcal{P}_n(\mathcal{U})| &= \binom{n + |\mathcal{U}| - 1}{|\mathcal{U}| - 1} \\ &\le (n+1)^{|\mathcal{U}|} \end{aligned} \tag{10}$$

$$Q^n(\underline{u}) = 2^{-n\left(H\left(P_{\underline{u}}\right) + D\left(P_{\underline{u}} \| Q\right)\right)} \; \forall\, \underline{u} \in \mathcal{U}^n. \tag{11}$$

Define

$$\mathcal{J}(n) = \{0, 1, \dots, |\mathcal{P}_n(\mathcal{U})| - 1\} \tag{12}$$

$$\mathcal{K}(j, n) = \{0, 1, \dots, |T\left(P^{j,n}\right)| - 1\},\; j \in \mathcal{J}(n) \tag{13}$$

$$A(j, \epsilon, n) = \left\lceil \epsilon \left| T\left(P^{j,n}\right)\right| \right\rceil,\; j \in \mathcal{J}(n). \tag{14}$$

We now construct the set of permutations $\Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n)) \subset \Pi(\mathrm{sr}\,(\mathcal{U}^n))$. For each $j \in \mathcal{J}(n)$, order the members of $T\left(P^{j,n}\right)$ lexicographically. Then any $\underline{u}_{[n]} \in \mathcal{U}^n$ can be uniquely specified by $\left(j(\underline{u}_{[n]}), k(\underline{u}_{[n]})\right)$ where $j(\underline{u}_{[n]}) \in \mathcal{J}(n)$ satisfies $\underline{u}_{[n]} \in T\left(P^{j(\underline{u}_{[n]}),n}\right)$ and $k(\underline{u}_{[n]}) \in \mathcal{K}(j(\underline{u}_{[n]}), n)$ denotes the lexicographically ordered position of $\underline{u}_{[n]}$ in $T\left(P^{j(\underline{u}_{[n]}),n}\right)$. Conversely, we define $\underline{u}^{j,k}_{[n]}$ to be the $(k+1)$st member of $T\left(P^{j,n}\right)$.

We define the type class integral representation parametrized by ϵ as

$$\tau_\epsilon\left(\underline{u}_{[n]}\right) = \left(\sum_{i=0}^{j(\underline{u}_{[n]})-1} A(i, \epsilon, n)\right) + k(\underline{u}_{[n]}). \tag{15}$$

We then construct a set $\Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ of permutations on $\mathrm{sr}\,(\mathcal{U}^n)$ so that any $\pi_{\epsilon,n} \in \Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ satisfies

$$\begin{aligned} &\forall\, \underline{u}_{[n]} \; s.t.\; k(\underline{u}_{[n]}) < A\left(j(\underline{u}_{[n]}), \epsilon, n\right) : \\ &\quad \pi_{\epsilon,n}(\mathrm{sr}\left(\underline{u}_{[n]}\right)) = \tau_\epsilon\left(\underline{u}_{[n]}\right). \end{aligned} \tag{16}$$

Finally, we define the threshold

$$T_{\epsilon,n} = \sum_{j \in \mathcal{J}(n)} A(j, \epsilon, n) = \sum_{j \in \mathcal{J}(n)} \left\lceil \epsilon \left| T\left(P^{j,n}\right)\right| \right\rceil. \tag{17}$$

Intuitively, any $\pi_{\epsilon,n} \in \Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ maps approximately a fraction ϵ of the members of each type class $P^{j,n}$ to values below the threshold $T_{\epsilon,n}$, and the remaining ones to values at or above $T_{\epsilon,n}$. As n grows, this approximation becomes more exact. The set $\Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ contains more than one permutation since the definition given by (16) does not specify the order for strings $\underline{u}_{[n]}$ that satisfy $k(\underline{u}_{[n]}) \ge A\left(j(\underline{u}_{[n]}), \epsilon, n\right)$.

We now split $\underline{U}_{[n]}$ into $\underline{U}^a_{[n]}(\epsilon)$ and $\underline{U}^b_{[n]}(\epsilon)$

$$\underline{U}_{[n]} \mapsto \begin{pmatrix} \underline{U}^a_{[n]}(\epsilon) = \min\left(\pi_{\epsilon,n}\left(\mathrm{sr}\left(\underline{U}_{[n]}\right)\right), T_{\epsilon,n}\right) \\ \underline{U}^b_{[n]}(\epsilon) = \max\left(\pi_{\epsilon,n}\left(\mathrm{sr}\left(\underline{U}_{[n]}\right)\right), T_{\epsilon,n}\right) - T_{\epsilon,n} \end{pmatrix} \tag{18}$$

where $\pi_{\epsilon,n} \in \Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ and $T_{\epsilon,n}$ is given by (17). Note that $\underline{U}^a_{[n]}(\epsilon)$ has cardinality $T_{\epsilon,n} + 1$ and all $\pi_{\epsilon,n} \in \Pi_{\epsilon,n}(\mathrm{sr}\,(\mathcal{U}^n))$ lead to the same random variable $\underline{U}^a_{[n]}(\epsilon)$.

We next demonstrate the asymptotic continuity of the distribution of $\underline{U}^a_{[n]}(\epsilon)$ with respect to ϵ. The given property is not obvious because for $0 \le \epsilon' < \epsilon \le 1$ and large enough n, $T_{\epsilon,n} > T_{\epsilon',n}$. Moreover, for the same value of $r < T_{\epsilon',n} < T_{\epsilon,n}$, the event $\{\underline{U}^a_{[n]}(\epsilon) = r\}$ does not necessarily correspond in any sense to the event $\{\underline{U}^a_{[n]}(\epsilon') = r\}$. Nonetheless, Lemma 3.1, proved in Appendix B, shows that asymptotic Lipschitz continuity of $P_{\underline{U}^a_{[n]}(\epsilon)}(\cdot)$ essentially holds. Lemma 3.2, proved in Appendix C, shows the corresponding property for the joint distribution $P_{\underline{U}^{\mathcal{S}}_{[n]}, \underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot, \cdot)$.

Lemma 3.1: For any $\epsilon, \epsilon' \in [0, 1]$, $\underline{U}^a_{[n]}(\epsilon)$ forms a bijection with another random variable $\underline{\tilde{U}}^a_{[n]}(\epsilon)$ that satisfies

$$\lim_{n \to \infty} \left| P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot) \right|_1 = 2|\epsilon - \epsilon'|.$$

Lemma 3.2: Let $(\underline{U}^a_{[n]}(\epsilon), \underline{U}^b_{[n]}(\epsilon))$ be a split of the discrete memoryless source U, and let $\underline{U}^{\mathcal{S}}$ be another set of discrete memoryless sources. Then for any $\epsilon, \epsilon' \in [0,1]$, $\underline{U}^a_{[n]}(\epsilon)$ forms a bijection with another random variable $\tilde{\underline{U}}^a_{[n]}(\epsilon)$ that satisfies

$$\lim_{n\to\infty} \left| P_{\underline{U}^{\mathcal{S}}_{[n]},\tilde{\underline{U}}^a_{[n]}(\epsilon)}(\cdot,\cdot) - P_{\underline{U}^{\mathcal{S}}_{[n]},\tilde{\underline{U}}^a_{[n]}(\epsilon')}(\cdot,\cdot) \right|_1 = 2|\epsilon - \epsilon'|.$$

Lemma 3.3, proved in Appendix D, demonstrates the relationship between the entropy rate $\mathcal{H}(U^a(\epsilon))$ and $H(U)$. Lemma 3.4, proved in Appendix E, shows the corresponding continuity for the conditional entropy.

Lemma 3.3: For $\epsilon \in [0,1]$, the random variable $\underline{U}^a_{[n]}(\epsilon)$ defined in (18) satisfies $\mathcal{H}(U^a(\epsilon)) = \epsilon H(U)$.

Lemma 3.4 (Range Lemma): Let $(\underline{U}^a_{[n]}(\epsilon), \underline{U}^b_{[n]}(\epsilon))$ be a split of the discrete memoryless source U. Then $g(\epsilon) = \mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon)\right)$ defines a continuous function from [0, 1] onto the interval $[H(\underline{U}^{\mathcal{S}}|U), H(\underline{U}^{\mathcal{S}})]$.

Together, these results prove that any point on the dominant face of the achievable rate region can be approximated to arbitrary accuracy using the given approach, as shown in Theorem 3.5.

Theorem 3.5: For two sources U^1, U^2 with joint distribution $P\left(u^1, u^2\right)$, any point on the dominant face $\mathcal{D}$ of $\mathcal{R}\left[P\left(u^1, u^2\right)\right]$ can be transformed via source-splitting U^1 according to (18) to a vertex in $\mathcal{R}\left[P\left(u^{1a}, u^2, u^{1b}\right)\right]$.

Proof: Using the chain rule for entropy and the fact that $U^1 \leftrightarrow (U^{1a}(\epsilon), U^{1b}(\epsilon))$, we have that

$$\begin{aligned} R_1 + R_2 &= H\left(U^1, U^2\right) \\ &= \mathcal{H}\left(U^1, U^2\right) \\ &= \mathcal{H}\left(U^{1a}(\epsilon), U^{1b}(\epsilon), U^2\right) \\ &= \mathcal{H}\left(U^{1a}(\epsilon)\right) + \mathcal{H}\left(U^2|U^{1a}(\epsilon)\right) \\ &\quad + \mathcal{H}\left(U^{1b}(\epsilon)|U^{1a}(\epsilon), U^2\right). \end{aligned}$$

By the Range Lemma we can set ϵ so that $R_2 = \mathcal{H}\left(U^2|U^{1a}(\epsilon)\right)$. We may then define $R_a = \mathcal{H}\left(U^{1a}(\epsilon)\right)$ and $R_b = \mathcal{H}\left(U^{1b}(\epsilon)|U^{1a}(\epsilon), U^2\right)$ where $R_a + R_b = R_1$. Then we note from the Slepian–Wolf theorem that the rate-tuple (R_a, R_2, R_b) is achievable, and furthermore, it is a vertex of the region $\mathcal{R}\left[P\left(u^{1a}, u^2, u^{1b}\right)\right]$. □

B. M Sources: At Most One Split Per Source Required

We now apply the source-splitting procedure for the Slepian–Wolf problem with $M > 2$ users and show that $2M - 1$ virtual sources are sufficient. The argument is based upon a recursive generalization of Theorem 3.5. The technique employed to show this is analogous to [17, Sec. II]. From there it follows from direct manipulation of the arguments in [17, Sec. III] that at most one split per source is required.

Theorem 3.6: Consider M correlated sources $\underline{U}^{[M]}$ with product distribution $P_{\underline{U}^{[M]}}\left(\underline{u}^{[M]}\right)$, and let $\mathcal{R}\left[P_{\underline{U}^{[M]}}\left(\underline{u}^{[M]}\right)\right]$ and $\mathcal{D}$ be the corresponding Slepian–Wolf achievable achievable rate region and dominant face. Any $R_{[M]} \in \mathcal{D}$ may be transformed to a vertex in a $2M - 1$ source Slepian–Wolf achievable rate region by splitting each source at most once using (18).

Proof: Suppose $R_{[M]} \in \mathcal{D}$. Apply the split (18) to source U^M to arrive at $\left(U^a(\epsilon), U^b(\epsilon)\right)$. For each $\mathcal{S} \subseteq [M-1]$ the inequality

$$R(\mathcal{S}) \le \mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon)\right) \tag{19}$$

is valid for all sufficiently small $\epsilon \in [0,1]$ by the following argument. For $\epsilon = 0$ it is valid, since

$$R(\mathcal{S}) \le H\left(\underline{U}^{\mathcal{S}}\right) = H\left(\underline{U}^{\mathcal{S}}|U^a(0)\right).$$

Since $\mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon)\right)$ is continuous in ϵ, there exists a largest interval $J_{\mathcal{S}} = [0, \epsilon_{\mathcal{S}}] \subset [0,1]$ such that (19) is fulfilled for all $\epsilon \in J_{\mathcal{S}}$.

Hence, for any $\mathcal{S} \subseteq [M-1]$ we have from (19) that

$$R(\mathcal{S}) \le \mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon_{\mathcal{S}})\right) \tag{20}$$

and from the definition of $\epsilon_{\mathcal{S}}$ it follows that

$$R(\mathcal{S}) = \mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon_{\mathcal{S}})\right). \tag{21}$$

Choose

$$\epsilon' = \min_{\mathcal{S}\subseteq[M-1]} \epsilon_{\mathcal{S}} \tag{22}$$

and let $\mathcal{T} \subseteq [M-1]$ be the largest subset of $[M-1]$ that satisfies $\epsilon_{\mathcal{T}} = \epsilon'$. From (21) with $\mathcal{S} = \mathcal{T}$ we have

$$R(\mathcal{T}) = \mathcal{H}\left(\underline{U}^{\mathcal{T}}|U^a(\epsilon')\right). \tag{23}$$

Define a virtual $(M+1)$-source $\left(U^1, \dots, U^{M-1}, U^b(\epsilon'), U^a(\epsilon')\right)$. Let $(R'_1, R'_2, \dots, R'_{M+1})$ be the $(M+1)$-tuple defined by $R'_i = R_i$, $i \in [M-1]$ and

$$\begin{aligned} R'_{M+1} &= \mathcal{H}\left(U^a(\epsilon')\right) \\ R'_M &= R_M - R'_{M+1}. \end{aligned} \tag{24}$$

We next show that $(R'_1, R'_2, \dots, R'_{M+1}) \in \mathcal{D}'$ where $\mathcal{D}'$ is the dominant face of the Slepian–Wolf achievable rate region corresponding to the $M+1$ sources. We first illustrate that (2) holds and then show achievability (1).

Note that by the definition of $(R'_1, R'_2, \dots, R'_{M+1})$ and since the splits form a bijection we have that

$$R'([M+1]) = R([M]) = H\left(\underline{U}^{[M]}\right) = \mathcal{H}\left(\underline{U}^{[M+1]'}\right). \tag{25}$$

It remains to be shown that the rate tuple $(R'_1, R'_2, \dots, R'_{M+1})$ is achievable, i.e.,

$$R'(\mathcal{S}) \ge \mathcal{H}\left(\underline{U'}^{\mathcal{S}}|\underline{U'}^{\mathcal{S}^c}\right), \ \forall\, \mathcal{S} \subseteq [M+1]. \tag{26}$$

We note from (25) and the chain rule for entropy that

$$\begin{aligned} R'(\mathcal{S}^c) \leq \mathcal{H}\left(\underline{U'}^{\mathcal{S}^c}\right) \Rightarrow R'(\mathcal{S}) &\geq \mathcal{H}\left(\underline{U'}^{[M+1]}\right) - \mathcal{H}\left(\underline{U'}^{\mathcal{S}^c}\right) \\ &= \mathcal{H}\left(\underline{U'}^{\mathcal{S}}|\underline{U'}^{\mathcal{S}^c}\right). \end{aligned} \tag{27}$$

Alternatively, it suffices to show the first inequality in (27) for each $\mathcal{S}^c \subseteq [M+1]$. We enumerate the following cases.

- $\{M, M+1\} \subset \mathcal{S}$ or $\{M, M+1\} \subset \mathcal{S}^c$: Equation (26) follows from (1).
- $M+1 \in \mathcal{S}^c$ and $M \in \mathcal{S}$

$$\begin{aligned} R'(\mathcal{S}^c) &= R'_{M+1} + R'(\mathcal{S}^c \backslash \{M+1\}) \\ &= R'_{M+1} + R(\mathcal{S}^c \backslash \{M+1\}) \\ &\leq \mathcal{H}(U^a(\epsilon')) + \mathcal{H}\left(\underline{U}^{\mathcal{S}^c \backslash \{M+1\}}|U^a(\epsilon')\right) \\ &= \mathcal{H}\left(U^a(\epsilon'), \underline{U}^{\mathcal{S}^c \backslash \{M+1\}}\right) \\ &= \mathcal{H}\left(\underline{U'}^{\mathcal{S}^c}\right) \end{aligned} \tag{28}$$

where (28) holds owing to (24) and (20).

- $M \in \mathcal{S}^c$ and $M+1 \in \mathcal{S}$

$$\begin{aligned} R'(\mathcal{S}^c) &= R'(\mathcal{S}^c \cup \{M+1\}) - R'_{M+1} \\ &= R(\mathcal{S}^c) - R'_{M+1} \\ &\leq H\left(\underline{U}^{\mathcal{S}^c}\right) - \mathcal{H}(U^a(\epsilon')) \\ &= H\left(\underline{U}^{\mathcal{S}^c \backslash \{M\}}, U^M\right) - \mathcal{H}(U^a(\epsilon')) \\ &= \mathcal{H}\left(\underline{U}^{\mathcal{S}^c \backslash \{M\}}, U^a(\epsilon'), U^b(\epsilon')|U^a(\epsilon')\right) \\ &\quad + \mathcal{H}(U^a(\epsilon')) - \mathcal{H}(U^a(\epsilon')) \\ &= \mathcal{H}\left(\underline{U}^{\mathcal{S}^c \backslash \{M\}}, U^a(\epsilon'), U^b(\epsilon')|U^a(\epsilon')\right) \\ &\leq \mathcal{H}\left(\underline{U}^{\mathcal{S}^c \backslash \{M\}}, U^a(\epsilon'), U^b(\epsilon')\right) \\ &= \mathcal{H}\left(\underline{U'}^{\mathcal{S}^c}\right) \end{aligned} \tag{29}$$

where (29) holds owing to (24) and (20).

Thus we have that $(R'_1, R'_2, \ldots, R'_{M+1}) \in \mathcal{D}'$. Note further that by our choice of ϵ' there exists a $\mathcal{T} \subseteq [M-1]$ such that

$$R'(\mathcal{T}) = R(\mathcal{T}) = \mathcal{H}\left(\underline{U}^{\mathcal{T}}|U^a(\epsilon')\right) = \mathcal{H}\left(\underline{U'}^{\mathcal{T}}|U^a(\epsilon')\right). \tag{30}$$

It follows that, besides (24), we also have

$$\begin{aligned} R'(\mathcal{S}) &= R'(\mathcal{T}) + R'(\{M+1\}) - R'(\{\mathcal{T} \backslash \mathcal{S}\} \cup \{M+1\}) \\ &\geq \mathcal{H}\left(\underline{U'}^{\mathcal{T}}|U^a(\epsilon')\right) + \mathcal{H}(U^a(\epsilon')) \\ &\quad - \mathcal{H}\left(\underline{U'}^{\mathcal{T} \backslash \mathcal{S}}, U^a(\epsilon')\right) \\ &= \mathcal{H}\left(\underline{U'}^{\mathcal{T}}, U^a(\epsilon')\right) - \mathcal{H}\left(\underline{U'}^{\mathcal{T} \backslash \mathcal{S}}, U^a(\epsilon')\right) \\ &= \mathcal{H}\left(\underline{U'}^{\mathcal{S}}|\underline{U'}^{\mathcal{T} \backslash \mathcal{S}}, U^a(\epsilon')\right) \quad \forall \mathcal{S} \subseteq \mathcal{T}. \end{aligned} \tag{31}$$

Finally, as part of (26) we have

$$R'(\mathcal{S}) \geq \mathcal{H}\left(\underline{U'}^{\mathcal{S}}|\underline{U}^{\mathcal{S}^c}\right) \qquad \forall \mathcal{S} \subseteq [M] \backslash \mathcal{T}. \tag{32}$$

This suggests the following parallelizable way of decoding $(R'_1, R'_2, \ldots, R'_{M+1})$. First note that from (24), we can entropy encode and decode $U^a(\epsilon')$ at rate $\mathcal{H}(U^a(\epsilon'))$. Knowledge of $U^a(\epsilon')$ can be kept at the decoder and we see that the group $\underline{U'}^{\mathcal{T}}$ can be encoded and decoded according to (30). This follows from (31) and the Slepian–Wolf coding theorem. Finally, it follows from (31) that with knowledge of $U^a(\epsilon')$ and $\underline{U'}^{\mathcal{T}}$ at the decoder, we may decode the remaining group of users. Each of these three groups has size at most $M-1$. From the $M=2$ case, we know that every rate point on the dominant face can be achieved by rate-splitting with at most $2M-1=3$ virtual sources. Let us assume by induction that for the $M-1$ user case, every rate tuple may be achieved with rate-splitting using at most $2(M-1)-1$ virtual sources. We just saw that for the M-user case, we can decompose it into a single-source encoding problem, and two Slepian–Wolf encoding problems of size m and $M-m$, respectively, where $1 \leq m < M$. By applying the induction hypothesis on these two smaller Slepian–Wolf encoding problems, we see that any rate-tuple in the M-user region can be achieved by rate-splitting with at most

$$1 + (2m-1) + (2(M-m)-1) = 2M-1$$

virtual sources.

Finally we observe that each user needs to split at most once to achieve any rate point on the dominant face. Algebraic topology techniques used to prove the analogous result in the discrete multiple access setting ([17, Sec. III]) directly apply in this setting. □

C. M Sources: The Boundary of the Dominant Face

Now we show that rate tuples on the *boundary* of the dominant face can be divided into two sets of sources that may be decoded successively but otherwise independently.

We can express the dominant face $\mathcal{D}\left[\mathcal{R}\left[P(\underline{u}^{[M]})\right]\right]$ in three ways:

$$\begin{aligned} \mathcal{D} = \mathcal{D}_1 &= \Big\{\underline{R} \in \mathbb{R}_+^M \mid H(\underline{U}^{\mathcal{S}}|\underline{U}^{\mathcal{S}^c}) \leq R(\mathcal{S}) \; \forall \mathcal{S} \subseteq [M] \\ &\qquad \text{with equality for } \mathcal{S} = [M]\Big\} \qquad (33) \\ = \mathcal{D}_2 &= \Big\{\underline{R} \in \mathbb{R}_+^M \mid H(\underline{U}^{\mathcal{S}}|\underline{U}^{\mathcal{S}^c}) \leq R(\mathcal{S}) \leq H(\underline{U}^{\mathcal{S}}) \\ &\qquad \forall \mathcal{S} \subseteq [M]\Big\} \qquad (34) \\ = \mathcal{D}_3 &= \Big\{\underline{R} \in \mathbb{R}_+^M \mid R(\mathcal{S}) \leq H(\underline{U}^{\mathcal{S}}) \; \forall \mathcal{S} \subseteq [M] \\ &\qquad \text{with equality for } \mathcal{S} = [M]\Big\} \qquad (35) \end{aligned}$$

where (33) is a restatement of (1),(2); (34) is a restatement of (5); and (35) follows because $\mathcal{D}_3 \supseteq \mathcal{D}_2$ holds directly and $\mathcal{D}_3 \subseteq \mathcal{D}_1$ holds by exchanging $\mathcal{S}$ in $\mathcal{D}_3$ with $\mathcal{S}^c$ in $\mathcal{D}_1$ and applying the chain rule for entropy.

We say a rate tuple $R \in \mathcal{D}$ lies on the *boundary* of $\mathcal{D}$ if there exists a proper subset $\mathcal{A} \subset [M]$ such that

$$R(\mathcal{A}) = H\left(\underline{U}^{\mathcal{A}}\right). \tag{36}$$

Rates that are on the boundary of $\mathcal{D}$ have the desirable property that they allow serial, but otherwise independent, decoding of sets of sources and their complements. More specifically, if R is on the boundary of $\mathcal{D}$ and $\mathcal{A}$ satisfies (36), then we can jointly decode the subset of inputs with index in $\mathcal{A}$ and subsequently jointly decode the subset of inputs with index in $\mathcal{A}^c = [M] \backslash \mathcal{A}$. The proof is as follows.

By definition, for a point on the boundary there is at least one $\mathcal{A} \subset [M]$ such that (36) holds. Now note that for any $\mathcal{L} \subset \mathcal{A}$

$$\begin{aligned} R(\mathcal{L}) &= H\left(\underline{U}^{\mathcal{A}}\right) - R(\mathcal{A}\backslash\mathcal{L}) \\ &\geq H\left(\underline{U}^{\mathcal{A}}\right) - H\left(\underline{U}^{\mathcal{A}\backslash\mathcal{L}}\right) \qquad (37) \\ &= H\left(\underline{U}^{\mathcal{L}} | \underline{U}^{\mathcal{A}\backslash\mathcal{L}}\right) \qquad (38) \end{aligned}$$

where (37) follows from (35). From (33) and (38), (36) we now have

$$R_{\mathcal{A}} \in \mathcal{D}\left[\mathcal{R}\left[P\left(\underline{u}^{\mathcal{A}}\right)\right]\right] \tag{39}$$

where $R_{\mathcal{A}} = (R_i)_{i \in \mathcal{A}}$. Thus $\underline{U}^{\mathcal{A}}$ can be decoded *independently* of $\underline{U}^{\mathcal{A}^c}$. Finally, since $R \in \mathcal{D}\left[\mathcal{R}\left[P(\underline{u}^{[M]})\right]\right]$, (33) allows for $\underline{U}^{\mathcal{A}^c}$ to be decoded successfully by using a successive decoder with $\underline{U}^{\mathcal{A}}$ as side information.

IV. Source-Splitting and Iterative Decoding for Slepian–Wolf

We discuss in this section how we can combine iterative decoding methods with source-splitting and point out how the splitting strategies defined in (6) and (8) significantly facilitate part of the decoding process. We conclude by showing simulation results.

Using the successive decoding approach of Section III we can near-losslessly compress a pair of sources (U^1, U^2) drawn according to $P\left(u^1, u^2\right)$ at any rate (R_1, R_2) on the dominant face $\mathcal{D}$ of $\mathcal{R}\left[P\left(u^1, u^2\right)\right]$. The strategy performs the splitting operation (6) and allocates rates according to (7a)–(7d).

Good binning strategies exist to perform successing decoding at rates that are vertices of the Slepian–Wolf region. Iterative decoding using "syndrome-former" LDPC encoders [11]–[15] and punctured turbo code encoders [6]–[9] have been extremely successful.

The iterative decoding technique applied here is the sum-product algorithm [32], which operates on the graphical structure of the code. For example, Fig. 3 illustrates a normal graph representation [33] of an LDPC used as a syndrome-former encoder, where the syndrome $\underline{s}$ is the index of the bin in which input $\underline{u}$ lies. The sum-product algorithm produces symbol-wise a posteriori probabilities (*APP*s), which are approximate on graphs with cycles. We use carefully constructed graphical representations that allow for the approximate *APP*s to give credible empirical performance. In the context of our problem, the bin indices handed to the decoder for (U^{1a}, U^{1b}, U^2) are denoted as $(\underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2)$. At each level of the pipeline, the *APP* outputs of previously decoded users are used as inputs to the currently operating decoder. The outputs of the iterative decoders are the approximate *APP*s

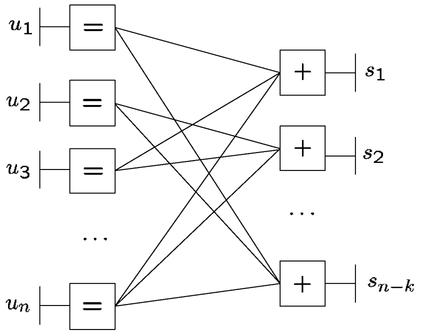

Fig. 3. Normal syndrome-former encoding graph.

$$\begin{aligned} P\left(U_i^{1a} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^{1a}(u) \\ P\left(U_i^{1b} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^{1b}(u) \\ P\left(U_i^{2} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^{2}(u). \end{aligned}$$

Let the outputs of the decoder be the estimate $(\hat{\underline{U}}^1, \hat{\underline{U}}^2)$, which may be constructed from the *APP*s of (U^1, U^2) by performing the symbol-based maximum *a posteriori* (MAP) decoding

$$\hat{U}_i^j = \arg \max_{u \in \{0,1,\ldots|\mathcal{U}|-1\}} \mathrm{app}_i^j(u).$$

While $\mathrm{app}_i^2(u)$ is the direct output of one of the iterative decoders, $(\mathrm{app}_i^{1a}(u), \mathrm{app}_i^{1b}(u))$ must be combined to yield $\mathrm{app}_i^1(u)$. The splitting strategy (6) leads to the implication

$$\begin{aligned} j \neq T : U_i^{1a} = j &\Rightarrow U^{1b} = 0 \qquad (40) \\ j \neq 0 : U_i^{1b} = j &\Rightarrow U^{1a} = T \qquad (41) \end{aligned}$$

and thus $\mathrm{app}_i^1(u)$ can be constructed with very low complexity

$$\begin{aligned} u < T : &P\left(U_i^1 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1a} = u, U_i^2 = 0 | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1b} = 0 | U_i^1 = u, \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &\quad \times P\left(U_i^{1a} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1a} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= \mathrm{app}_i^1(u) \quad \text{by (40)} \\ u > T : &P\left(U_i^1 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1a} = T, U_i^{1b} = u - T | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1a} = T | U_i^{1b} = u - T, \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &\quad \times P\left(U_i^{1b} = u - T | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= P\left(U_i^{1b} = u - T | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= \mathrm{app}_i^{1b}(u - T) \quad \text{by (41)} \\ &\quad \times P\left(U_i^1 = T | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= 1 - \sum_{u \neq T} P\left(U_i^1 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) \\ &= 1 - \left(\sum_{u=0}^{T-1} \mathrm{app}_i^{1a}(u)\right) - \left(\sum_{u=T+1}^{|\mathcal{U}|-1} \mathrm{app}_i^{1b}(u - T)\right). \end{aligned}$$

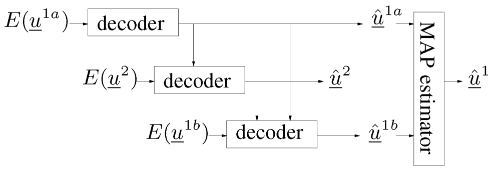

Fig. 4. Combining iterative decoding with source-splitting.

Fig. 4 gives a schematic of the decoding process. In the case of binary splitting (8), the decoder observes bin indices $\underline{s}^1, \underline{s}^2, \ldots, \underline{s}^{|\mathcal{U}|-1}$ and the iterative successive decoder outputs will be the *APP*s

$$P\left(U_i^1 = u|\underline{s}^1, \underline{s}^2, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \triangleq \text{app}_i^1(u)$$

$$\vdots \quad \vdots$$

$$P\left(U_i^{|\mathcal{U}|-1} = u|\underline{s}^1, \underline{s}^2, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \triangleq \text{app}_i^{|\mathcal{U}|-1}(u).$$

In this case the implication

$$k \in \{1, \ldots, |\mathcal{U}|-1\} \text{ and } U_i^k = 1 \Rightarrow U_i^r = 0,\ \forall r \neq k \quad (42)$$

holds and we can construct $\text{app}_i(u)$ again with very low complexity:

$$\begin{aligned} k \neq 0 : P&\left(U_i = k|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= P\left(U_i^k = 1, U_i^r = 0, r \neq k|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= P\left(U_i^r = 0, r \neq k|U_i^k = 1, \underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &\quad \times P\left(U_i^k = 1|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= P\left(U_i^k = 1|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= \text{app}_i^k(1) \text{ by } (42) \\ &\quad P\left(U_i = 0|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= 1 - \sum_{k=1}^{|\mathcal{U}|-1} P\left(U_i = k|\underline{s}^1, \ldots, \underline{s}^{|\mathcal{U}|-1}\right) \\ &= 1 - \sum_{k=1}^{|\mathcal{U}|-1} \text{app}_i^k(1). \end{aligned}$$

A. *Simulation Results*

Synthetic Data: We now discuss simulation results that illustrate the promise of this splitting technique. The experiments begin with the random selection of a joint probability distribution for sources over $\mathcal{U}^1 = \mathcal{U}^2 = \mathcal{U} = GF(2^m)$ for some m. We then draw n independent samples and encode using an irregular LDPC with degree distribution drawn according to the density evolution results provided in [34]. Once the nonzero components of the parity matrix are constructed, their values are selected randomly from $\{1, \ldots, 2^m - 1\}$. We perform the sum-product update rule in its dual form ([33, Sec. IX]), which operates on the Fourier Transform of *APP*s. Also we note that in the case of $GF(2^m)$, the transormed *APP*s lie in $\mathbb{R}$ rather than $\mathbb{C}$. Thus the same gain in decoding complexity reduction is attained here as is in the binary case.

Fig. 5 illustrates the achievability of nonvertices in the two source Slepian–Wolf problem using splitting and iterative decoding for $m = 2$ and $n = 5000$. The leftmost plot shows four nonvertex rate pairs on the boundary of the achievable region. We perform iterative decoding in their neighborhoods for a collection of points. The rightmost plot shows the symbol error rate as a function of the difference between the sum rate and the joint entropy. The given results show error probabilities of 10^{-4} at sum rate penalties between 0.1 and 0.25.

V. Linear Programming Methods for Slepian–Wolf

Low-density parity check codes are linear codes based on bipartite graphs whose nodes have bounded degrees regardless of block length. Let us consider a binary linear (n, k) code $\mathcal{C}$ that consists of 2^k binary codewords of length n. We associate $\mathcal{C}$ with its parity check matrix $H \in \{0,1\}^{(n-k)\times n}$

$$H = \begin{bmatrix} -H_1- \\ -H_2- \\ \vdots \\ -H_{n-k}- \end{bmatrix}$$

using the condition that any $\underline{c} \in \{0,1\}^n$ satisfies $\underline{c} \in \mathcal{C}$ if and only if

$$H\underline{c} = \underline{0}. \quad (43)$$

Graphical representations denote the dependencies between codewords based upon the constraints they must satisfy. For a linear code, each local constraint is a smaller linear code. Fig. 3 illustrates a normal graph representation [33], where bits are associated with edges and constraint codes are associated with nodes. A node with a "+" sign and degree d is a $(d, d-1, 2)$ single parity check code that imposes the constraint that the bits lying on the d edges adjacent to that node must sum (modulo 2) to 0. A node of degree d' with an "=" sign is a $(d', 1, 2)$ repetition code and imposes the constraint that the bits lying on the d' adjacent edges must be equal. There are n nodes with "=" labels, each of which corresponds to a single repetition code and has one "half-edge" connection to an external bit. There are $n - k$ nodes with "+" labels, each of which corresponds to a single parity-check code and has one "half-edge" connection to an external syndrome bit. The ith parity-check node is connected to the jth repetition node if and only if the i, j entry of H is 1, i.e., if the jth variable node is involved in the computation of the ith syndrome symbol. The set of all valid $(\underline{u}, \underline{s})$ input-output sequences is the set of $(\underline{u}, \underline{s})$ pairs that satisfy all local constraints.

For a given $\underline{s} \in \{0,1\}^{n-k}$, the set of all input $\underline{u}$ sequences consistent with the output $\underline{s}$ is given by

$$\text{Co}(H, \underline{s}) = \{\underline{u} : H\underline{u} = \underline{s}\}. \quad (44)$$

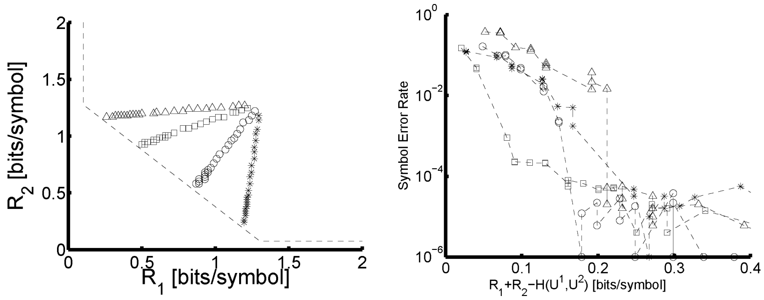

Fig. 5. Symbol error rate for source-splitting to achieve nonvertex rate pairs.

A. Decoding of Linear Block Codes on the Binary Symmetric Channel

Over a memoryless binary symmetric channel (BSC), we have the following channel model:

$$Y_i = X_i \oplus W_i$$

where each transmitted symbol $X_i \in \{0,1\}$, each noise symbol $W_i \in \{0,1\}$ is $\mathrm{Bernoulli}(p)$, each received symbol $Y_i \in \{0,1\}$, and $\oplus$ is binary addition modulo 2. If a binary linear code $\mathcal{C}$ is used to transmit, then we have from (43),(44) that $\underline{c} \in \mathcal{C}$ if and only

$$\underline{c} \in \mathrm{Co}\left(H, \underline{0}\right). \tag{45}$$

If we define $\gamma \triangleq \ln\left(\frac{P(W_i=0)}{P(W_i=1)}\right) > 0$, a sufficient statistic for decoding is

$$\gamma_i = \ln\left(\frac{P(Y_i = y_i | X_i = 0)}{P(Y_i = y_i | X_i = 1)}\right) = (-1)^{y_i}\gamma$$

for each i. We describe maximum likelihood sequence decoding (MLSD) as follows:

$$\begin{aligned}
\underline{c}^*(\underline{y}) &= \arg \max_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(\prod_{i=1}^{n} P(Y_i = y_i | X_i = c_i)\right) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(-\sum_{i=1}^{n} \ln P(Y_i = y_i | X_i = c_i)\right) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \Bigg(\sum_{i=1}^{n} \Big[\ln P(Y_i = y_i | X_i = 0) \\
&\qquad - \ln P(Y_i = y_i | X_i = c_i)\Big]\Bigg) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(\sum_{i:c_i=1} \ln\left(\frac{P(Y_i = y_i | X_i = 0)}{P(Y_i = y_i | X_i = 1)}\right)\right) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(\sum_{i=1}^{n} \gamma_i c_i\right) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(\sum_{i=1}^{n} \gamma (-1)^{y_i} c_i\right) \\
&= \arg \min_{\underline{c} \in \mathrm{Co}(H,\underline{0})} \left(\sum_{i=1}^{n} (-1)^{y_i} c_i\right).
\end{aligned}$$

By noting that all linear programs over polytopes have an optimal extreme point [35], and by defining

$$\mathcal{B}(\underline{s}) = CH\left(\mathrm{Co}\left(H, \underline{s}\right)\right) \tag{46}$$

where CH denotes the convex hull, the above problem can be cast as a linear program:

$$\underline{c}^*(\underline{y}) = \arg \min_{u \in \mathcal{B}(\underline{0})} \sum_{i=1}^{n} (-1)^{y_i} c_i. \tag{47}$$

However, the MLSD decoding problem is NP-complete [36] and thus explicitly representing the polytope (46) is prohibitively complex.

1) Coset Leaders: We next discuss an alternative decoding approach for the BSC that also directly applies to ML-decoding scenario for block-encoding for lossless source compression, where the syndrome-former matrix is H, and the noise $\underline{w}$ is identified with the source. Since every $\underline{c} \in \mathcal{C}$ satisfies (45), we have that

$$H\underline{y} = H\underline{c} + H\underline{w} = H\underline{w} = \underline{s} \tag{48}$$

and thus $\underline{w} \in \mathrm{Co}\left(H, \underline{s}\right)$. So we may first find the most likely noiseword in the corresponding coset

$$\begin{aligned}
\underline{w}^*(\underline{s}) &= \arg \max_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(\prod_{i=1}^{n} P(W_i = w_i)\right) \\
&= \arg \min_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(-\sum_{i=1}^{n} \ln P(W_i = w_i)\right) \\
&= \arg \min_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(\sum_{i=1}^{n} \left[\ln P(W_i = 0) - \ln P(W_i = w_i)\right]\right) \\
&= \arg \min_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(\sum_{i:w_i=1} \ln\left(\frac{P(W_i = 0)}{P(W_i = 1)}\right)\right) \\
&= \arg \min_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(\sum_{i=1}^{n} \gamma w_i\right) \\
&= \arg \min_{\underline{w} \in \mathrm{Co}(H,\underline{s})} \left(\sum_{i=1}^{n} w_i\right)
\end{aligned}$$

which corresponds to being the minimal-weight member of the coset, or *coset leader*. We note that $\underline{w}^*(\underline{s})$ may also be characterized in LP form as

$$\underline{w}^*(\underline{s}) = \arg \min_{\underline{w} \in \mathcal{B}(\underline{s})} \left(\sum_{i=1}^{n} w_i \right) \tag{49}$$

and that the complexity in expressing the polytope $\mathcal{B}(\underline{s})$ arises here as well. Nonetheless, we see the codeword estimate is

$$\underline{c}^*(\underline{y}) = \underline{y} \oplus \underline{w}^*(\underline{s})$$

and that for the same $\underline{w}$ sequence, (47) is correct if and only if (49) is correct. Thus, these two optimization problems are equally powerful.

We now discuss relaxed polytopes for (47) and (49) that can be efficiently represented. A relaxed polytope $\tilde{\mathcal{B}}(\underline{s})$ can be defined as the intersection of $n-k$ polyhedra $\tilde{\mathcal{B}}_j(s_j)$, where each $\tilde{\mathcal{B}}_j(s_j)$ is the convex hull of all code symbols consistent with the local parity check j and syndrome s_j. Note that (as in Fig. 3), parity-check j is connected to one syndrome symbol s_j and a set $\bar{N}(j)$ of $\delta_j = |\bar{N}(j)|$ adjacent variable nodes. If $s_j = 0$, then $\mathrm{Co}\,(H_j, 0)$ is the set of valid configurations for the symbols with indices in $\bar{N}(j)$. We define $[\mathrm{Co}\,(H_j, 0)]$ to be the matrix with each element of $\mathrm{Co}\,(H_j, 0)$ as a column vector. In the event that $s_j = 1$, the valid configurations form the coset $\mathrm{Co}\,(H_j, 1)$ which is $\{0,1\}^{\delta_j} \setminus \mathrm{Co}\,(H_j, 0)$ and $[\mathrm{Co}\,(H_j, 1)]$ is defined similarly. With this notation we construct $\tilde{\mathcal{B}}(\underline{s})$ as follows:

$$\begin{aligned}\tilde{\mathcal{B}}_j(s_j) = \Big\{ &\underline{u} \in [0,1]^n : \ \underline{u}_{|\bar{N}(j)} = [\mathrm{Co}\,(H_j, s_j)]\,\underline{x}, \\ &\underline{x} \in \mathbb{R}^{2^{\delta_j - 1}}, 0 \le x_i \le 1, \sum_i x_i = 1 \Big\} \\ \tilde{\mathcal{B}}(\underline{s}) = &\bigcap_{j=1}^{n-k} \tilde{\mathcal{B}}_j(s_j)\end{aligned}$$

where $\underline{u}_{|V}$ is defined to be the restriction of $\underline{u}$ to the coordinates in V.

We now state the following lemma.

Lemma 5.1: Any $\underline{u} \in \mathbb{R}^n$ is an extreme point of $\tilde{\mathcal{B}}_j(s_j)$ if and only if:

1) $\forall\ i \in \{1, \ldots, n\} \setminus \bar{N}(j), u_i \in \{0,1\}$

2) $\underline{u}_{|\bar{N}(j)} \in \mathrm{Co}\,(H_j, s_j)$.

Proof: For any $i \notin \bar{N}(j)$, the only constraint involving u_i in the polytope is the constraint that $u_i \in [0,1]$. It thus follows that 1) holds for any extreme point $\underline{u}$ of $\tilde{\mathcal{B}}_j(s_j)$. The only constraints involving $\underline{u}_{|\bar{N}(j)}$ are that $\underline{u}_{|\bar{N}(j)} \in [0,1]^{\delta_j}$ and that $\underline{u}_{|\bar{N}(j)} \in CH(\mathrm{Co}\,(H_j, s_j))$. Since $\mathrm{Co}\,(H_j, s_j) \subset \{0,1\}^{\delta_j} \subset [0,1]^{\delta_j}$, and since the convex hull of any set $\mathcal{S}$ has as its extreme points the set $\mathcal{S}$, 2) holds if $\underline{u}$ is an extreme point of $\tilde{\mathcal{B}}_j(s_j)$.

For the converse, suppose that 1) or 2) does not hold. If 1) does not hold, then either $\underline{u}$ is infeasible and thus is not an extreme point, or $\underline{u}$ is feasible and $u_i = \alpha$ for some $\alpha \in (0,1)$. Note that $\alpha = \alpha(1) + (1-\alpha)(0)$ and thus $\underline{u}$ is the strict convex combination of the two feasible vectors formed by replacing u_i with 1 and 0 respectively. If 2) does not hold, then either $\underline{u}$ is infeasible, and thus not an extreme point, or $\underline{u}$ is feasible and $\underline{u}_{|\bar{N}(j)} \notin \mathrm{Co}\,(H_j, s_j)$. Again, since the convex hull of any set $\mathcal{S}$ has as its extreme points the set $\mathcal{S}$, $\underline{u}_{|\bar{N}(j)}$ is a strict convex combination of vectors lying in $\mathrm{Co}\,(H_j, s_j)$. We extend each of those vectors to $[0,1]^n$ by letting their ith position (where $i \notin \bar{N}(j)$) be u_i and note they are all still feasible. It thus follows that $\underline{u}$ is a strict convex combination of feasible vectors in $\tilde{\mathcal{B}}_j(s_j)$. □

Feldman *et al.* [27], [28], [26] have recently considered polynomial-time LP relaxations for LDPC's that exhibit the ML-certificate property: if an integral LP solution is found, it is the ML-codeword. These relaxations correspond to replacing $\mathcal{B}(\underline{0})$ with $\tilde{\mathcal{B}}(\underline{0})$ in (47):

$$\underline{c}^*(\underline{y}) = \arg \min_{\underline{c} \in \tilde{\mathcal{B}}(\underline{0})} \sum_{i=1}^{n} (-1)^{y_i} c_i. \tag{50}$$

All valid codewords are vertices of this polytope, but nonintegral vertices, termed "pseudocodewords", also arise and thus compete in the optimization. Recent work in [26] shows that on the BSC, using easily constructable "expander" LDPC's [29]–[31] with this LP decoder yields a positive error exponent (exponential error probability decay in block length). By [22]–[24], pseudocodewords also compete with true codewords when the "min-sum" algorithm is applied to the graphical representation of the same code. Furthermore, [24] shows that the region over which the "pseudocodewords" compete with true codewords is in fact $\tilde{\mathcal{B}}(\underline{0})$. Discussions in [24], [28], [25] suggest that the two decoders have essentially the same performance. This gives another motivation for considering the LP decoding paradigm – it is more amenable to concrete analysis and is intimately connected to iterative decoding algorithms.

We now consider from [28] a pair $\underline{x} \in [0,1]^n$ and $\underline{y} \in \{0,1\}^n$ and note the transformation

$$\underline{x}^{[\underline{y}]} \triangleq (|y_1 - x_1|, \ldots, |y_n - x_n|). \tag{51}$$

The following lemma captures useful properties of $\underline{x}^{[\underline{y}]}$.

Lemma 5.2: Given any $\underline{x} \in [0,1]^n$ and $\underline{y} \in \{0,1\}^n$.

1) $\left(\underline{x}^{[\underline{y}]}\right)^{[\underline{y}]} = \underline{x}$.

2) For any $\alpha \in [0,1]$ and any $\tilde{\underline{x}} \in [0,1]^n$, $(\alpha\underline{x} + (1-\alpha)\tilde{\underline{x}})^{[\underline{y}]} = \alpha\underline{x}^{[\underline{y}]} + (1-\alpha)\tilde{\underline{x}}^{[\underline{y}]}$.

3) $\underline{x} \in \{0,1\}^n \Rightarrow \underline{x}^{[\underline{y}]} = \underline{x} \oplus \underline{y}$.

Proof:

1)

$$\begin{aligned} |y_i - |y_i - x_i|| &= \begin{cases} 1 - (1 - x_i), & \text{if } y_i = 1 \\ |0 - |0 - x_i||, & \text{if } y_i = 0 \end{cases} \\ &= x_i. \end{aligned}$$

2)

$$\begin{aligned} &|y_i - (\alpha x_i + (1-\alpha)\tilde{x}_i)| \\ &= |\alpha y_i + (1-\alpha) y_i - (\alpha x_i + (1-\alpha)\tilde{x}_i)| \\ &= |\alpha\,(y_i - x_i) + (1-\alpha)\,(y_i - \tilde{x}_i)| \\ &= \begin{cases} \alpha\,(1 - x_i) + (1-\alpha)\,(1 - \tilde{x}_i), & \text{if } y_i = 1 \\ \alpha x_i + (1-\alpha)\tilde{x}_i, & \text{if } y_i = 0 \end{cases} \\ &= \alpha\,|y_i - x_i| + (1-\alpha)\,|y_i - \tilde{x}_i|. \end{aligned}$$

3) Follows by inspection. □

It was proposed in [37] to replace the polytope $\mathcal{B}(\underline{s})$ in (49) with $\tilde{\mathcal{B}}(\underline{s})$

$$\underline{w}^*(\underline{s}) = \arg \min_{\underline{w} \in \tilde{\mathcal{B}}(\underline{s})} \left(\sum_{i=1}^{n} w_i \right) \tag{52}$$

in the context of source block compression with LDPC's to arrive at a polynomial-time algorithm that also exhibits the ML-certificate property. In light of the equivalence of (47) and (49) in Section V-A, it is natural to ask the question whether the two relaxations (50) and (52) are equally as powerful. The answer is yes, as shown by the following theorem.

Theorem 5.3: Given any $\underline{y} \in \{0,1\}^n$, any $\underline{s}^c \in \{0,1\}^{n-k}$, and any $H \in \{0,1\}^{n-k \times n}$, the following two optimization problems are isomorphic:

$$\min_{\underline{c} \in \tilde{\mathcal{B}}(\underline{s}^c)} \left(\sum_{i=1}^{n} (-1)^{y_i} c_i \right) \tag{53}$$

$$\min_{\underline{w} \in \tilde{\mathcal{B}}(\underline{s}^c \oplus \underline{s}^y)} \left(\sum_{i=1}^{n} w_i \right) \tag{54}$$

where

$$\underline{s}^y \triangleq H\underline{y}. \tag{55}$$

Proof:

a) Consider the objective function in (53) and note that for all $\underline{c} \in [0,1]^n$

$$\begin{aligned} \sum_{i=1}^{n} (-1)^{y_i} c_i &= \left(\sum_{i=1}^{n} |y_i - c_i| \right) - |\{i : y_i = 1\}| \\ &= \left(\sum_{i=1}^{n} w_i \right) - |\{i : y_i = 1\}| \end{aligned}$$

where we have defined $\underline{w} = \underline{c}^{[\underline{y}]}$.

b) We consider the transformation

$$\underline{w} = \underline{c}^{[\underline{y}]}$$

that maps points in $\tilde{\mathcal{B}}(\underline{s}^c)$ to points in $\tilde{\mathcal{B}}(\underline{s}^y \oplus \underline{s}^c)$. Consider any $\underline{x}$ that is an extreme point of $\tilde{\mathcal{B}}_j(s_j^c)$ and any $i \notin \bar{N}(j)$. By 1) of Lemma 5.1 and 3) of Lemma 5.2

$$x_i^{[y_i]} \in \{0,1\}. \tag{56}$$

By 2) of Lemma 5.1, $\underline{x}_{|\bar{N}(j)} \in \mathrm{Co}\left(H_j, s_j^c\right)$. By 3) of Lemma 5.2

$$\underline{x}_{|\bar{N}(j)}{}^{[\underline{y}_{|\bar{N}(j)}]} = \underline{x}_{|\bar{N}(j)} \oplus \underline{y}_{|\bar{N}(j)}.$$

It thus follows that

$$\begin{aligned} H_j \underline{x}_{|\bar{N}(j)}{}^{[\underline{y}_{|\bar{N}(j)}]} &= H_j \left(\underline{x}_{|\bar{N}(j)} \oplus \underline{y}_{|\bar{N}(j)} \right) \\ &= H_j \underline{x}_{|\bar{N}(j)} \oplus H_j \underline{y}_{|\bar{N}(j)} \\ &= s_j^c \oplus s_j^y \\ \Leftrightarrow \underline{x}_{|\bar{N}(j)}{}^{[\underline{y}_{|\bar{N}(j)}]} &\in \mathrm{Co}\left(H_j, s_j^c \oplus s_j^y\right). \end{aligned} \tag{57}$$

From (56), (57), and Lemma 5.1, $\underline{x}^{[\underline{y}]}$ is an extreme point of $\tilde{\mathcal{B}}_j(s_j^c \oplus s_j^y)$. Moreover, since $\underline{x}^{[\underline{y}]}$ satisfies 2) of Lemma 5.2, the polytope $\tilde{\mathcal{B}}(\underline{s}^c)$ is mapped to the polytope $\tilde{\mathcal{B}}(\underline{s}^y \oplus \underline{s}^c)$.

From a), the $\underline{x}^{[y]}$ transformation maps the objective function of (50) to the objective function of (52) plus a constant that is invariant to the polytope. From b), the $\underline{x}^{[y]}$ transformation maps the constraint polytope of (50) to the constraint polytope of (52). By exchanging the roles of $\underline{w}$ and $\underline{c}$, noting 1) of Lemma 5.2 and applying the exact same arguments, we conclude that the two problems are one-to-one transformations of one another. □

As a consequence, Theorem 5.3 guarantees that the same class of expander codes [29]–[31] discussed in [26] yields the same positive error exponent when applied as syndrome-formers for source block encoding. Moreover, it follows from a direct manipulation of the arguments in [24] that application of the LP decoder is also intimately related to applying the 'min-sum' algorithm to the syndrome-former graphical representation of the code.

B. LP Decoding at Vertices for Slepian–Wolf

We next show how Theorem 5.3 applies to decoding at vertex points of the Slepian–Wolf problem. Suppose (U^1, U^2) drawn according to $P\left(u^1, u^2\right)$ have been encoded at rate $(R_1, R_2) = \left(H(U^1), H(U^2|U^1)\right)$. Assume U^1 has been decoded correctly and the objective is to decode U^2 given U^1 as side information. Suppose $U^1 \in \mathcal{U} = \{0, 1, \ldots, |\mathcal{U}| - 1\}$ and $U^2 \in \mathcal{V} = \{0, 1\}$. Then we define the following likelihood ratios

$$\gamma_u \triangleq \ln \left(\frac{P(U_i^2 = 0 | U_i^1 = u)}{P(U_i^2 = 1 | U_i^1 = u)} \right), \quad \forall\, u \in \mathcal{U}.$$

By performing an analysis similar to the above derivations, we arrive at the following Slepian–Wolf ML relaxation

$$\underline{u}^* = \arg \min_{\underline{u}^2 \in \tilde{\mathcal{B}}(\underline{s}^2)} \sum_{k=0}^{|\mathcal{U}|-1} \sum_{i : u_i^1 = k} \gamma_k u_i^2$$

and note that it also exhibits the ML-certificate property.

If we consider the special case where $\mathcal{U} = \{0, 1\}$ and the correlation structure is symmetric, (i.e., $P(U_i^2 = 0 | U_i^1 = 0) = P(U_i^2 = 1 | U_i^1 = 1)$), then MLSD of U^2 from its syndrome $\underline{s}^2$ given U^1 as side information corresponds to

$$\min_{\underline{u}^2 \in \mathcal{B}(\underline{s}^2)} \left(\sum_{i=1}^{n} (-1)^{u_i^1} u_i^2 \right).$$

Consider the following relaxation with the ML-certificate property

$$\min_{\underline{u}^2 \in \tilde{\mathcal{B}}(\underline{s}^2)} \left(\sum_{i=1}^{n} (-1)^{u_i^1} u_i^2 \right)$$

and note that it is of the form of (53). It thus follows from Theorem 5.3 that using an expander-style LDPC as mentioned previously along with this LP decoder results in a positive error exponent.

VI. Conclusion

In this paper we continue to apply the theme of transforming successful low-complexity channel coding strategies to ones that are applicable to source coding. We introduce low-complexity approaches for Slepian–Wolf distributed data compression. These techniques include a source-splitting approach to facilitate successive decoding at nonvertices. This method does not require common sources of randomness at the encoders and decoder, reduces the alphabet sizes at the outcome of the splitter, and has nice simplifications when used with iterative decoding. We demonstrate the effectiveness of this approach with synthetically generated data. Using this technique, any arbitrary rate can be transformed into a vertex in a higher-dimensional problem by splitting each source at most once. We also discuss rate tuples on the boundary of the dominant face and show how they can be split into two sets decodable sequentially but otherwise independently.

We also introduce linear programming methodologies for the Slepian–Wolf problem. These methodologies are deeply connected to an LP methodology designed for the channel coding domain, as well as the "min-sum" iterative decoding algorithm. By showing the equivalence between two linear programs, we show that for the two-source Slepian–Wolf problem with a symmetric binary joint distribution, using an easily constructable encoder and the proposed LP decoder results in a positive error exponent.

It is our hope that these methodologies will further strengthen the quest to design practical coding schemes for the general Slepian–Wolf problem with provably good performance.

Appendix A
Definitions

The following definitions and lemma are useful for proving Lemmas 3.1, 3.3, and 3.4. Define

$$e_{\min}(Q,\mathcal{U}) = \inf_{P\in\mathcal{P}(\mathcal{U})} \{H(P) + D(P\|Q)\} \tag{58}$$

$$e_{\max}(Q,\mathcal{U}) = \sup_{P\in\mathcal{P}(\mathcal{U})} \{H(P) + D(P\|Q)\}. \tag{59}$$

Lemma 1.1: Consider any $Q \in \mathcal{P}(\mathcal{U})$ such that $|\mathcal{U}| > 1$ and $Q(a) > 0$ for each $a \in \mathcal{U}$. Then $e_{\min}(Q,\mathcal{U})$ as defined in (58) satisfies $e_{\min}(Q,\mathcal{U}) > 0$, and $e_{\max}(Q,\mathcal{U})$ as defined in (59) satisfies $e_{\max}(Q,\mathcal{U}) < \infty$.

Proof: For any $P \in \mathcal{P}(\mathcal{U})$,

$$H(P) + D(P\|Q) = \sum_{a\in\mathcal{U}} -P(a)\log_2(Q(a)). \tag{60}$$

Since $Q(a) > 0$ for each $a \in \mathcal{U}$, $Q(a) < 1$ for each $a \in \mathcal{U}$. Thus $-\log_2(Q(a)) > 0$ for each $a \in \mathcal{U}$. Since Q is fixed in the optimization (58), and since the log function is monotonic, there exists a P^* that minimizes (60) and satisfies $P^*(a_{\max}) = 1$ where $a_{\max} \in \arg\max_{a\in\mathcal{U}} Q(a)$. Thus

$$e_{\min}(Q,\mathcal{U}) = -\log_2(Q(a_{\max})) > 0.$$

Since $Q(a) > 0$ for each $a \in \mathcal{U}$, $-\log_2(Q(a)) < \infty$ for all $a \in \mathcal{U}$. Again, since Q is fixed in the optimization (59), and since the log function is monotonic, there exists a P^* that maximizes (60) that satisfies $P^*(a_{\min}) = 1$ where $a_{\min} \in \arg\min_{a\in\mathcal{U}} Q(a)$. Thus,

$$e_{\max}(Q,\mathcal{U}) = -\log_2(Q(a_{\min})) < \infty. \qquad \square$$

To aid in proving Lemmas 3.1, 3.2, and 3.4, map each $\underline{u}_{[n]} \in \mathcal{U}^n$ to $\tau_1\left(\underline{u}_{[n]}\right) \in \{0,1,\ldots,|\mathcal{U}|^n - 1\}$ using the type class integral representation given in (15) with $\epsilon = 1$:

$$\tau_1\left(\underline{u}_{[n]}\right) = \left(\sum_{i=0}^{j(\underline{u}_{[n]})-1} |T(P^{i,n})|\right) + k(\underline{u}_{[n]}). \tag{61}$$

Define $\xi = |\mathcal{U}|^n$ and the random variable $\tilde{\underline{U}}^a_{[n]}(\epsilon)$ with alphabet $\{0,1,\ldots,|\mathcal{U}|^n - 1\} \cup \{\xi\}$ in terms of $\underline{U}_{[n]}$ as

$$\tilde{\underline{U}}^a_{[n]}(\epsilon) = \begin{cases} \tau_1\left(\underline{U}_{[n]}\right), & \text{if } k(\underline{U}_{[n]}) < A\left(j(\underline{U}_{[n]}),\epsilon,n\right) \\ \xi, & \text{if } k(\underline{U}_{[n]}) \geq A\left(j(\underline{U}_{[n]}),\epsilon,n\right). \end{cases} \tag{62}$$

For every $\epsilon \in [0,1]$,

$$\underline{U}^a_{[n]}(\epsilon) = \tau_\epsilon\left(\underline{u}^{j,k}_{[n]}\right) \text{ iff } \underline{U}_{[n]} = \underline{u}^{j,k}_{[n]} \text{ and } k < A(j,\epsilon,n) \tag{63}$$

$$\tilde{\underline{U}}^a_{[n]}(\epsilon) = \tau_1\left(\underline{u}^{j,k}_{[n]}\right) \text{ iff } \underline{U}_{[n]} = \underline{u}^{j,k}_{[n]} \text{ and } k < A(j,\epsilon,n) \tag{64}$$

$$\underline{U}^a_{[n]}(\epsilon) = T_{\epsilon,n} \text{ iff } k(\underline{U}_{[n]}) \geq A\left(j(\underline{U}_{[n]}),\epsilon,n\right) \tag{65}$$

$$\tilde{\underline{U}}^a_{[n]}(\epsilon) = \xi \text{ iff } k(\underline{U}_{[n]}) \geq A\left(j(\underline{U}_{[n]}),\epsilon,n\right). \tag{66}$$

Thus $\tilde{\underline{U}}^a_{[n]}(\epsilon)$ and $\underline{U}^a_{[n]}(\epsilon)$ form a bijection. Note the following properties of $P_{\tilde{\underline{U}}^a_{[n]}(\epsilon)}(\cdot)$ for all $j \in \mathcal{J}(n)$, $k \in \mathcal{K}(j,n)$:

$$P_{\tilde{\underline{U}}^a_{[n]}(\epsilon)}(\xi) = P_{\underline{U}^a_{[n]}(\epsilon)}(T_{\epsilon,n}) \tag{67}$$

$$P_{\tilde{\underline{U}}^a_{[n]}(\epsilon)}\left(\tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) = 1_{\{k<A(j,\epsilon,n)\}} P_{\underline{U}_{[n]}}\left(\underline{u}^{j,k}_{[n]}\right). \tag{68}$$

Since $\underline{U}^a_{[n]}(\epsilon)$ is a function of $\underline{U}_{[n]}$, it follows that $\underline{U}^{\mathcal{S}}_{[n]} \to \underline{U}_{[n]} \to \underline{U}^a_{[n]}(\epsilon)$ forms a Markov chain. Since $\underline{U}^a_{[n]}(\epsilon) \leftrightarrow \tilde{\underline{U}}^a_{[n]}(\epsilon)$, $\underline{U}^{\mathcal{S}}_{[n]} \to \underline{U}_{[n]} \to \tilde{\underline{U}}^a_{[n]}(\epsilon)$ also forms a Markov chain. Thus for any $j \in \mathcal{J}(n)$, $k \in \mathcal{K}(j,n)$,

$$\begin{aligned}
&P_{\underline{U}^{\mathcal{S}}_{[n]},\tilde{\underline{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) \\
&= \sum_{\underline{u}_{[n]}} P_{\tilde{\underline{U}}^a_{[n]}(\epsilon),\underline{U}_{[n]}}\left(\tau_1\left(\underline{u}^{j,k}_{[n]}\right), \underline{u}_{[n]}\right) P_{\underline{U}^{\mathcal{S}}_{[n]}|\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}|\underline{u}_{[n]}\right) \\
&= P_{\tilde{\underline{U}}^a_{[n]}(\epsilon),\underline{U}_{[n]}}\left(\tau_1\left(\underline{u}^{j,k}_{[n]}\right), \underline{u}^{j,k}_{[n]}\right) P_{\underline{U}^{\mathcal{S}}_{[n]}|\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}|\underline{u}^{j,k}_{[n]}\right) \\
&= P_{\tilde{\underline{U}}^a_{[n]}(\epsilon)|\underline{U}_{[n]}}\left(\tau_1\left(\underline{u}^{j,k}_{[n]}\right)|\underline{u}^{j,k}_{[n]}\right) P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}, \underline{u}^{j,k}_{[n]}\right) \\
&= 1_{\{k<A(j,\epsilon,n)\}} P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}, \underline{u}^{j,k}_{[n]}\right)
\end{aligned}$$

$$\begin{aligned}
&P_{\underline{U}^{\mathcal{S}}_{[n]},\tilde{\underline{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]}, \xi\right) \\
&= P_{\underline{U}^{\mathcal{S}}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}\right) \\
&\quad - \sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} P_{\underline{U}^{\mathcal{S}}_{[n]},\tilde{\underline{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right).
\end{aligned} \tag{69}$$

Define

$$DP^{n,\epsilon,\epsilon'}(\cdot,\cdot) \triangleq P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot,\cdot) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot,\cdot) \quad (70)$$

and note that for any $1 \geq \epsilon > \epsilon' \geq 0$, $A(j,\epsilon,n) \geq A(j,\epsilon',n)$ implies

$$\begin{aligned} DP^{n,\epsilon,\epsilon'}&\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) \\ &= \left(1_{\{k<A(j,\epsilon,n)\}} - 1_{\{k<A(j,\epsilon',n)\}}\right) \\ &\quad \times P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}, \underline{u}^{j,k}_{[n]}\right) \\ &\geq 0 & (71) \\ DP^{n,\epsilon,\epsilon'}&\left(\underline{u}^{\mathcal{S}}_{[n]}, \xi\right) \\ &= -\sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) \\ &\leq 0. & (72) \end{aligned}$$

Appendix B Proof of Lemma 3.1

Proof: Assume without loss of generality that $0 \leq \epsilon' \leq \epsilon \leq 1$. Since $\epsilon \geq \epsilon'$ implies $A(j,\epsilon,n) \geq A(j,\epsilon',n)$

$$\underline{\tilde{U}}^a_{[n]}(\epsilon') = 1_{\left\{\underline{\tilde{U}}^a_{[n]}(\epsilon) < T_{\epsilon',n}\right\}} \underline{\tilde{U}}^a_{[n]}(\epsilon) + 1_{\left\{\underline{\tilde{U}}^a_{[n]}(\epsilon) \geq T_{\epsilon',n}\right\}} \xi \quad (73)$$

$$P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\xi) \leq P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\xi) \quad (74)$$

$$P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(r) \geq P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(r) \ \forall\, r \in \{0,\ldots,|\mathcal{U}|^n - 1\}. \quad (75)$$

As a result

$$\begin{aligned} &\left|P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot)\right|_1 \\ &= P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\xi) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\xi) \\ &\quad + \sum_{r=0}^{|\mathcal{U}|^n-1} \left[P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(r) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(r)\right] \\ &= \left(1 - \sum_{r=0}^{|\mathcal{U}|^n-1} P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(r)\right) - \left(1 - \sum_{r=0}^{|\mathcal{U}|^n-1} P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(r)\right) \\ &\quad + \sum_{r=0}^{|\mathcal{U}|^n-1} \left[P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(r) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(r)\right] \\ &= 2\sum_{r=0}^{|\mathcal{U}|^n-1} \left[P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(r) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(r)\right] \\ &= 2\sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} \left(1_{\{k<A(j,\epsilon,n)\}} - 1_{\{k<A(j,\epsilon',n)\}}\right) P_{\underline{U}_{[n]}}\left(\underline{u}^{j,k}\right) \\ &= 2\sum_{j\in\mathcal{J}(n)} \left(A(j,\epsilon,n) - A(j,\epsilon',n)\right) 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]} \\ &= 2\sum_{j\in\mathcal{J}(n)} \left(\lceil \epsilon |T\left(P^{j,n}\right)| \rceil - \lceil \epsilon' |T\left(P^{j,n}\right)| \rceil\right) \\ &\quad \times 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]}. & (76) \end{aligned}$$

Note that

$$\begin{aligned} &\left|\lceil \epsilon |T\left(P^{j,n}\right)| \rceil - \lceil \epsilon' |T\left(P^{j,n}\right)| \rceil - (\epsilon - \epsilon')|T\left(P^{j,n}\right)|\right| \leq 1, \\ &\sum_{j\in\mathcal{J}(n)} (\epsilon - \epsilon')|T\left(P^{j,n}\right)| 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]} = \epsilon - \epsilon' \end{aligned} \quad (77)$$

where (77) follows from

$$\begin{aligned} 1 &= \sum_{\underline{u}_{[n]}\in\mathcal{U}^n} P_{\underline{U}_{[n]}}\left(\underline{u}_{[n]}\right) \\ &= \sum_{j\in\mathcal{J}(n)} \left|T\left(P^{j,n}\right)\right| 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\left|\left|P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot)\right|_1 - 2(\epsilon - \epsilon')\right| \\ &\leq 2\sum_{j\in\mathcal{J}(n)} 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]} \\ &\leq 2\left|\mathcal{P}_n(\mathcal{U})\right| 2^{-ne_{\min}(Q,\mathcal{U})} & (78) \\ &\leq 2(n+1)^{|\mathcal{U}|} 2^{-ne_{\min}(Q,\mathcal{U})} & (79) \end{aligned}$$

where (78) is due to (58) and (12) and (79) is due to (10). Thus,

$$\lim_{n\to\infty} \left|\left|P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot)\right|_1 - 2(\epsilon - \epsilon')\right| = 0$$

by Lemma 1.1. □

Appendix C Proof of Lemma 3.2

Proof: Assume without loss of generality that $0 \leq \epsilon' < \epsilon \leq 1$. Thus,

$$\begin{aligned} &\left|DP^{n,\epsilon,\epsilon'}(\cdot,\cdot)\right|_1 \\ &= \sum_{\underline{u}^{\mathcal{S}}_{[n]}} \left\{ \left|DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]}, \xi\right)\right| \right. \\ &\quad \left. + \sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} \left|DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right)\right| \right\} \\ &= 2\sum_{\underline{u}^{\mathcal{S}}_{[n]}} \sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) & (80) \\ &= 2\sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} \left(1_{\{k<A(j,\epsilon,n)\}} - 1_{\{k<A(j,\epsilon',n)\}}\right) \\ &\quad \times \sum_{\underline{u}^{\mathcal{S}}_{[n]}} P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}, \underline{u}^{j,k}_{[n]}\right) & (81) \\ &= 2\sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} \left(1_{\{k<A(j,\epsilon,n)\}} - 1_{\{k<A(j,\epsilon',n)\}}\right) \\ &\quad \times P_{\underline{U}_{[n]}}\left(\underline{u}^{j,k}_{[n]}\right) \\ &= \left|P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\cdot) - P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\cdot)\right|_1 & (82) \end{aligned}$$

where (80) is due to (72), (81) is due to (69), and (82) is due to (76). From here we finish the proof by applying Lemma 3.1. □

APPENDIX D
PROOF OF LEMMA 3.3

Proof: For an arbitrary $\epsilon \in [0,1]$

$$\begin{aligned}
&\frac{1}{n} H\left(\underline{U}^a_{[n]}(\epsilon)\right)\\
&= \frac{1}{n} \sum_{k=0}^{T_{\epsilon,n}} -\Pr\left(\underline{U}^a_{[n]}(\epsilon) = k\right) \log_2\left(\Pr\left(\underline{U}^a_{[n]}(\epsilon) = k\right)\right)\\
&= -\frac{1}{n} \sum_{j\in\mathcal{J}(n)} A(j,\epsilon,n)\, 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]}\\
&\quad \times \log_2\left(2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]}\right)\\
&\quad - \frac{1}{n}\Pr\left(\underline{U}^a_{[n]}(\epsilon)= T_{\epsilon,n}\right) \log_2\left(\Pr\left(\underline{U}^a_{[n]}(\epsilon)= T_{\epsilon,n}\right)\right)\\
&= \sum_{j\in\mathcal{J}(n)} \left[H\left(P^{j,n}\right) + D(P^{j,n}\|Q)\right] A(j,\epsilon,n)\\
&\quad \times 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]}\\
&\quad - \frac{1}{n}\Pr\left(\underline{U}^a_{[n]}(\epsilon= T_{\epsilon,n}\right) \log_2\left(\Pr\left(\underline{U}^a_{[n]}(\epsilon) = T_{\epsilon,n}\right)\right). \qquad (83)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathcal{H}(U^a(\epsilon)) \qquad (84)\\
&\quad = \lim_{n\to\infty} \frac{1}{n} H\left(\underline{U}^a_{[n]}(\epsilon)\right)\\
&\quad = \lim_{n\to\infty} \sum_{j\in\mathcal{J}(n)} \left[H\left(P^{j,n}\right) + D(P^{j,n}\|Q)\right] A(j,\epsilon,n)\\
&\qquad \times 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]}\\
&\quad = \lim_{n\to\infty} \sum_{j\in\mathcal{J}(n)} \left\lceil \epsilon\left|T\left(P^{j,n}\right)\right|\right\rceil \left[H\left(P^{j,n}\right) + D(P^{j,n}\|Q)\right]\\
&\qquad \times 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]} \qquad (85)
\end{aligned}$$

where in (85), (83) vanishes because for any $p \in [0,1]$, $0 \leq -p\log_2 p \leq \frac{1}{2}$.

Note that

$$\begin{aligned}
0 &\leq \lceil \epsilon|T\left(P^{j,n}\right)|\rceil - \epsilon|T\left(P^{j,n}\right)| \leq 1,\\
\epsilon H(U) &= \sum_{j\in\mathcal{J}(n)} \epsilon\left|T\left(P^{j,n}\right)\right| \left[H\left(P^{j,n}\right) + D(P^{j,n}\|Q)\right]\\
&\quad \times 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]} \qquad (86)
\end{aligned}$$

where (86) follows from (11) since

$$\begin{aligned}
H(U) &= \frac{1}{n} H\left(\underline{U}_{[n]}\right)\\
&= \frac{1}{n} \sum_{\underline{u}_{[n]}\in\mathcal{U}^n} -P_{\underline{U}_{[n]}}\left(\underline{u}_{[n]}\right) \log_2 P_{\underline{U}_{[n]}}\left(\underline{u}_{[n]}\right)\\
&= -\frac{1}{n} \sum_{j\in\mathcal{J}(n)} \left|T\left(P^{j,n}\right)\right| 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]}\\
&\quad \times \log_2\left(2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]}\right)\\
&= \sum_{j\in\mathcal{J}(n)} \left|T\left(P^{j,n}\right)\right| \left[H\left(P^{j,n}\right) + D\left(P^{j,n}\|Q\right)\right]\\
&\quad \times 2^{-n\left[H\left(P^{j,n}\right)+D\left(P^{j,n}\|Q\right)\right]}.
\end{aligned}$$

Thus,

$$\begin{aligned}
0 &\leq \mathcal{H}(U^a(\epsilon)) - \epsilon H(U)\\
&\leq \lim_{n\to\infty} \sum_{j\in\mathcal{J}(n)} \left[H\left(P^{j,n}\right) + D(P^{j,n}\|Q)\right]\\
&\qquad \times 2^{-n\left[H\left(P^{j,n}\right)+D(P^{j,n}\|Q)\right]}\\
&\leq \lim_{n\to\infty} |\mathcal{P}_n(\mathcal{U})|\, e_{\max}(Q,\mathcal{U})\, 2^{-ne_{\min}(Q,\mathcal{U})} \qquad (87)\\
&\leq e_{\max}(Q,\mathcal{U}) \lim_{n\to\infty} (n+1)^{|\mathcal{U}|} 2^{-ne_{\min}(Q,\mathcal{U})} \qquad (88)\\
&= 0 \qquad (89)
\end{aligned}$$

where (87) is due to (58), (59), (12); (88) is due to (10); and (89) is due to Lemma 1.1. □

APPENDIX E
PROOF OF LEMMA 3.4

Proof: We first show that $\mathcal{H}\left(\underline{U}^{\mathcal{S}}, U^a(\epsilon)\right)$ is continuous in ϵ. Assume $0 < \epsilon' < \epsilon < 1$. Note from Lemma 1.1 that

$$\begin{aligned}
e_{\min}\left(Q_{U^{\mathcal{S}}}, \mathcal{U}^{\mathcal{S}}\right) &> 0\\
e_{\max}\left(Q_{U^{\mathcal{S}},U}, \mathcal{U}^{\mathcal{S}} \times \mathcal{U}\right) &< \infty. \qquad (90)
\end{aligned}$$

Define

$$\begin{aligned}
L(\epsilon,\epsilon',n) &\triangleq -\frac{1}{n} \sum_{\underline{u}^{\mathcal{S}}_{[n]}} \Bigg[P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\qquad \times \log_2\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\qquad - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\qquad \times \log_2\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\Bigg]\\
DH_n(\epsilon,\epsilon') &\triangleq \frac{1}{n}\left[H\left(\underline{U}^{\mathcal{S}}_{[n]},\underline{U}^a_{[n]}(\epsilon)\right) - H\left(\underline{U}^{\mathcal{S}}_{[n]},\underline{U}^a_{[n]}(\epsilon')\right)\right].
\end{aligned}$$

Then

$$\begin{aligned}
DH_n(\epsilon,\epsilon') &= \frac{1}{n}\left[H\left(\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)\right) - H\left(\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')\right)\right]\\
&= L(\epsilon,\epsilon',n)\\
&\quad - \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}} \sum_{j\in\mathcal{J}(n)} \sum_{k\in\mathcal{K}(j,n)} DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]}, \tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{U}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]},\underline{u}^{j,k}_{[n]}\right).
\end{aligned}$$

We now bound $L(\epsilon,\epsilon',n)$. Note that, for all $n > n_0 = \left\lceil \frac{1}{e_{\min}\left(Q_{U^{\mathcal{S}}},\mathcal{U}^{\mathcal{S}}\right)} \right\rceil$ and all $\underline{u}^{\mathcal{S}}_{[n]}$

$$\begin{aligned}
P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) &\leq P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\leq P_{\underline{U}^{\mathcal{S}}_{[n]}}\left(\underline{u}^{\mathcal{S}}_{[n]}\right) \leq 2^{-ne_{\min}\left(Q_{U^{\mathcal{S}}},\mathcal{U}^{\mathcal{S}}\right)}\\
&< \frac{1}{2}
\end{aligned}$$

where the first inequality follows from (72). Since the function $g(x) = -x\log_2 x$ is monotonically increasing on $[0, \frac{1}{2})$, for all $n > n_0$ and all $\underline{u}^{\mathcal{S}}_{[n]}$

$$-\Bigg[P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\Bigg] \le 0$$

which implies $L(\epsilon,\epsilon',n) \le 0$. We can also lower bound $L(\epsilon,\epsilon',n)$ as follows:

$$\begin{aligned}
&L(\epsilon,\epsilon',n)\\
&= \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\Bigg[P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\quad - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\Bigg]\\
&\quad + \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&= \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\log_2\left(\frac{P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)}{P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)}\right)\\
&\quad + \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\ge \frac{1}{n}\left(\sum_{\underline{u}^{\mathcal{S}}_{[n]}}P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2\frac{\left(\sum_{\underline{u}^{\mathcal{S}}_{[n]}}P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)}{\left(\sum_{\underline{u}^{\mathcal{S}}_{[n]}}P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)} &&(91)\\
&\quad + \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&= \frac{1}{n}\left(P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\xi)\right)\\
&\quad \times \log_2\frac{\left(P_{\underline{\tilde{U}}^a_{[n]}(\epsilon')}(\xi)\right)}{\left(P_{\underline{\tilde{U}}^a_{[n]}(\epsilon)}(\xi)\right)}\\
&\quad + \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\ge 0 &&(92)\\
&\quad + \frac{1}{n}\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\left(P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon')}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right) - P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\right)\\
&\quad \times \log_2 P_{\underline{U}^{\mathcal{S}}_{[n]},\underline{\tilde{U}}^a_{[n]}(\epsilon)}\left(\underline{u}^{\mathcal{S}}_{[n]},\xi\right)\\
&\ge -\frac{1}{2}e_{\max}\left(Q_{U^{\mathcal{S}},U},\mathcal{U}^{\mathcal{S}}\times\mathcal{U}\right)\left|DP^{n,\epsilon,\epsilon'}(\cdot,\cdot)\right|_1 &&(93)
\end{aligned}$$

where (91) follows from the log-sum inequality [4, p. 29], (92) is due to (74), and (93) is due to (59), Lemma 1.1, and (80). Thus

$$\begin{aligned}
&-\frac{1}{2}e_{\max}\left(Q_{U^{\mathcal{S}},U},\mathcal{U}^{\mathcal{S}}\times\mathcal{U}\right)\left|DP^{n,\epsilon,\epsilon'}(\cdot,\cdot)\right|_1\\
&\le DH_n(\epsilon,\epsilon')\\
&\le e_{\max}\left(Q_{U^{\mathcal{S}},U},\mathcal{U}^{\mathcal{S}}\times\mathcal{U}\right)\\
&\quad \times\sum_{\underline{u}^{\mathcal{S}}_{[n]}}\sum_{j\in\mathcal{J}(n)}\sum_{k\in\mathcal{K}(j,n)}DP^{n,\epsilon,\epsilon'}\left(\underline{u}^{\mathcal{S}}_{[n]},\tau_1\left(\underline{u}^{j,k}_{[n]}\right)\right) &&(94)\\
&= \frac{1}{2}e_{\max}\left(Q_{U^{\mathcal{S}},U},\mathcal{U}^{\mathcal{S}}\times\mathcal{U}\right)\left|DP^{n,\epsilon,\epsilon'}(\cdot,\cdot)\right|_1 &&(95)
\end{aligned}$$

where (94) is due to (90), and (95) is due to (80). Thus

$$\begin{aligned}
\left|\lim_{n\to\infty}DH_n(\epsilon,\epsilon')\right| &= \left|\mathcal{H}\left(\underline{U}^{\mathcal{S}},U^a(\epsilon)\right) - \mathcal{H}\left(\underline{U}^{\mathcal{S}},U^a(\epsilon')\right)\right|\\
&\le e_{\max}\left(Q_{U^{\mathcal{S}},U},\mathcal{U}^{\mathcal{S}}\times\mathcal{U}\right)(\epsilon-\epsilon')
\end{aligned}$$

by Lemma 3.2. Thus $\mathcal{H}\left(\underline{U}^{\mathcal{S}},U^a(\epsilon)\right)$ is continuous in ϵ.

Finally, $\mathcal{H}\left(\underline{U}^{\mathcal{S}}|U^a(\epsilon)\right)$ is continuous in ϵ due to the continuity of $\mathcal{H}\left(U^a(\epsilon)\right)$ and $\mathcal{H}\left(\underline{U}^{\mathcal{S}},U^a(\epsilon)\right)$ along with the chain rule for entropy. The endpoints are contained because $\underline{U}^a_{[n]}(0)$ is a point mass and thus $\frac{1}{n}H\left(\underline{U}^a_{[n]}(0)\right) = 0$ and $\underline{U}^a_{[n]}(1)$ is bijective with U, and thus $\frac{1}{n}H\left(\underline{U}^a_{[n]}(1)\right) = H(U)$. □

ACKNOWLEDGMENT

The authors thank Jon Feldman, Ralf Koetter, and Karen Haigh for their helpful discussions and comments.

REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, pp. 471–480, 1973.

[2] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inf. Theory*, vol. 21, pp. 226–228, 1975.

[3] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inf. Theory*, vol. 28, pp. 585–592, 1982.

[4] T. M. Cover and J. Thomas, *Elements of Information Theory*. New York, NY: Wiley, 1991.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting codes and decoding: Turbo codes," in *Proc. IEEE Int. Commun. Conf.*, 1993.

[6] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, pp. 417–419, Oct. 2001.

[7] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compress. Conf.*, Apr. 2002, pp. 252–261.

[8] J. Bajcsy and P. Mitran, "Coding for the Slepian–Wolf problem with turbo codes," in *IEEE Proc. GLOBECOM*, Nov. 2001, pp. 1400–1404.

[9] A. Liveris, Z. Xiong, and C. Georghiades, “Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes,” in *Proc. IEEE DCC*, Brest, France, Mar. 2003, pp. 193–202.

[10] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[11] T. Tian, J. Garcia-Frias, and W. Zhong, “Compression of correlated sources using LDPC codes,” in *Proc. IEEE Data Compress. Conf.*, 2003.

[12] D. Schonberg, S. S. Pradhan, and K. Ramchandran, “LDPC codes can approach the Slepian–Wolf bound for general binary sources,” in *Proc. 40th Allerton Conf. Commun., Contr. Comput.*, Oct. 2002.

[13] A. D. Liveris, Z. Xiong, and C. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2003.

[14] J. Garcia-Frias and W. Zhong, “LDPC codes for compression of multi-terminal sources with hidden Markov correlation,” *IEEE Commun. Lett.*, vol. 7, pp. 115–117, Mar. 2003.

[15] A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, “Slepian–Wolf coding of three binary sources using LDPC codes,” in *Proc. Int. Symp. Turbo Codes Rel. Topics*, Brest, France, Sep. 2003.

[16] B. Rimoldi and R. Urbanke, “A rate-splitting approach to the Gaussian multiple access channel,” *IEEE Trans. Inf. Theory*, vol. 42, pp. 364–375, 1996.

[17] A. Grant, B. Rimoldi, R. Urbanke, and P. A. Whiting, “Rate-splitting multiple access for discrete memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 873–890, 2001.

[18] B. Rimoldi and R. Urbanke, “Asynchronous Slepian–Wolf coding via source-splitting,” in *Proc. IEEE Int. Symp. Inf. Theory*, Ulm, Germany, Jun.–Jul. 29–4, 1997, p. 271.

[19] F. M. J. Willems, “Totally asynchronous Slepian–Wolf data compression,” *IEEE Trans. Inf. Theory*, vol. 34, pp. 35–44, 1988.

[20] I. Csiszár, “The method of types,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 2205–2523, 1998.

[21] B. Rimoldi, “Generalized time sharing: A low-complexity capacity-achieving multiple-access technique,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 2432–2442, 2001.

[22] G. Forney, R. Koetter, J. Kschischang, and A. Reznik, “On the effective weights of pseudocodewords for codes defined on graphs with cycles,” *Codes, Syst. Graph. Models*, pp. 101–112, 2001.

[23] B. J. Frey, R. Koetter, and A. Vardy, “Signal space characterization of iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 766–781, 2001.

[24] R. Koetter and P. O. Vontobel, “Graph-covers and iterative decoding of finite length codes,” in *Proc. Turbo Codes Conf.*, Brest, 2003.

[25] P. O. Vontobel and R. Koetter, “On the relationship between linear programming decoding and min-sum algorithm decoding,” in *Proc. Int. Symp. Inf. Theory Appl.*, Parma, Italy, Oct. 2004.

[26] J. Feldman, T. Malkin, C. Stein, R. A. Servedio, and M. J. Wainwright, “LP decoding corrects a constant fraction of errors,” in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, Ill, Jun.–Jul. 27–2, 2004.

[27] J. Feldman, M. Wainwright, and D. R. Karger, “Using linear programming to decode linear codes,” in *Proc. Conf. Inf. Sci. Syst., The John Hopkins University*, Baltimore, MD, Mar. 2003.

[28] J. Feldman, “Decoding Error-Correcting Codes via Linear Programming,” Ph.D., Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[29] M. Sipser and D. Spielman, “Expander codes,” *IEEE Trans. Inf. Theory*, vol. 42, pp. 1710–1722, 1996.

[30] A. Barg and G. Zémor, “Error exponents of expander codes,” *IEEE Trans. Inf. Theory*, vol. 48, pp. 1725–1729, 2002.

[31] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, “Randomness conductors and constant-degree expansion beyond the degree/2 barrier,” in *Proc. 34th ACM Symp. Theory Comput.*, 2002, pp. 659–668.

[32] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, 2001.

[33] G. D. Forney, “Codes on graphs: Normal realizations,” *IEEE Trans. Inf. Theory*, pp. 101–112, 2001.

[34] A. Amarou and R. Urbanke, Ldpcopt [Online]. Available: http://lthcwww.epfl.ch/research/ldpcopt/

[35] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[36] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, “On the intractability of certain coding problems,” *IEEE Trans. Inf. Theory*, vol. 24, pp. 384–386, 1978.

[37] T. P. Coleman, A. H. Lee, M. Médard, and M. Effros, “On some new approaches to practical Slepian–Wolf compression inspired by channel coding,” in *Proc. IEEE Data Compress. Conf.*, Snowbird, Utah, Mar. 23–25, 2004.