Feedback Capacity and Coding for the $(0, k)$-RLL Input-Constrained BEC

Ori Peled, *Student Member, IEEE*, Oron Sabag, *Student Member, IEEE*, and Haim H. Permuter, *Senior Member, IEEE*

***Abstract*—The input-constrained binary erasure channel (BEC) with strictly causal feedback is studied. The channel input sequence must satisfy the $(0, k)$-runlength limited (RLL) constraint, i.e., no more than k consecutive '0's are allowed. The feedback capacity of this channel is derived for all $k \geq 1$, and is given by $C^{\text{fb}}_{(0,k)}(\varepsilon) = \max \frac{\bar{\varepsilon} H_2(\delta_0)+\sum_{i=1}^{k-1}\left(\bar{\varepsilon}^{i+1} H_2(\delta_i) \prod_{m=0}^{i-1} \delta_m\right)}{1+\sum_{i=0}^{k-1}\left(\bar{\varepsilon}^{i+1} \prod_{m=0}^{i} \delta_m\right)}$, where ε is the erasure probability, $\bar{\varepsilon} = 1 - \varepsilon$ and $H_2(\cdot)$ is the binary entropy function. The maximization is only over δ_{k-1}, while the parameters δ_i for $i \leq k-2$ are straightforward functions of δ_{k-1}. The lower bound is obtained by constructing a simple coding for all $k \geq 1$. It is shown that the feedback capacity can be achieved using zero-error, variable length coding. For the converse, an upper bound on the non-causal setting, where the erasure is available to the encoder just prior to the transmission, is derived. This upper bound coincides with the lower bound and concludes the search for both the feedback capacity and the non-causal capacity. As a result, non-causal knowledge of the erasures at the encoder does not increase the feedback capacity for the $(0, k)$-RLL input-constrained BEC. This property does not hold in general: the $(2, \infty)$-RLL input-constrained BEC, where every '1' is followed by at least two '0's, is used to show that the feedback capacity can be strictly smaller than the non-causal capacity.**

***Index Terms*—Constrained coding, feedback capacity, finite-state machine, Markov decision process, posterior matching, runlength limited (RLL) constraints.**

I. Introduction

THE physical limitations of the hardware used in recording and communication systems cause some digital sequences to be more prone to errors than others. This elicits the need to ensure that such sequences will not be recorded or transmitted. Constrained coding is a method that enables such systems to encode arbitrary data sequences into sequences that abide by the imposed restrictions [2]. In the classical constrained coding setting, it is assumed that the transmission is noiseless if the transmitted sequence satisfies the imposed constraint. In this paper, however, we consider a transmission of constrained sequences where the transmission is over a noisy channel, the binary erasure channel (BEC) (Fig. 2).

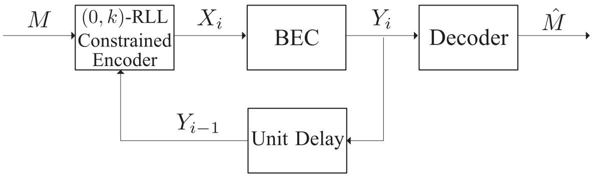

Fig. 1. Input constrained BEC with strictly causal feedback. The channel input X_i is a function of the message M and of the channel output history Y^{i-1}.

Run-length limited (RLL) constraints are common in magnetic and optical recording standards, where the run length of consecutive '0's should be limited between d and k ($d < k$). A (d, k)-RLL constrained binary sequence must satisfy two restrictions:

1) at least d '0's must follow each '1'.
2) no more than k consecutive '0's are allowed.

The first restriction ensures that the frequency of transitions, i.e., 1 → 0 or 0 → 1, will not be too high. This is necessary in systems where the sequence is conveyed over band-limited channels. Timing is commonly recovered with a phase-locked loop (PLL) that adjusts the phase of the detection instant according to the observed transition of the received waveform. The second restriction guarantees that the PLL does not fall out of synchronization with the waveform [2], [3]. RLL constraints are also present in the field of flash memory for various other reasons [4].

Two important families of the RLL constraint are the (d, ∞)-RLL and $(0, k)$-RLL. These constraints might seem symmetric in some sense, but indeed, may greatly differ in their behavior, see e.g., [5], [6]. Therefore, when dealing with RLL constraints, it is common to tackle each of these families separately before approaching the general (d, k)-RLL. In this paper, we adopt this approach and show that the $(0, k)$-RLL problem is solvable, while the same problem with a (d, ∞)-RLL constraint is a great deal more challenging.

The model studied in this paper is a BEC (Fig. 2), in which the input sequences must satisfy the $(0, k)$-RLL constraint. Two cases of this model are investigated, based on the

Manuscript received December 7, 2017; revised December 8, 2018; accepted January 26, 2019. Date of publication March 5, 2019; date of current version June 14, 2019. This work was supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 337752, in part by the ISF research grant 818/17, and in part by the German Research Foundation (DFG) via the German-Israeli Project Cooperation [DIP]. This paper was presented at the 2017 IEEE International Symposium on Information Theory (ISIT 2017) [1].

The authors are with the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 8410501, Israel (e-mail: oripe@post.bgu.ac.il; oronsa@post.bgu.ac.il; haimp@bgu.ac.il).

Communicated by J. Kliewer, Associate Editor for Coding Techniques.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2903252

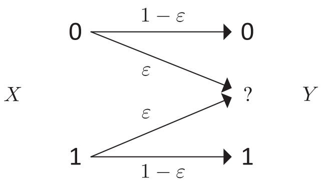

Fig. 2. Binary erasure channel with erasure parameter ε.

information that is available to the encoder. In the first case, described in Fig. 1, the encoder has access to all past outputs via a noiseless feedback link. In the second case, described in Fig. 3, the encoder has non-causal access to the erasure that is about to occur, that is, the encoder knows in advance whether the BEC behaves like a clean channel or not. From an operational point of view, the capacity of the non-causal case must be greater than the feedback case due to the additional information that the encoder has.

As RLL constraints are common in storage devices, the investigated model can be thought of as a model for writing data to flash memory with the following mechanism: when failing to write a particular cell, the cell is considered to contain an erasure. The feedback is a read operation that follows each write and indicates whether or not the write attempt was successful.

We show that the feedback capacity of the $(0, k)$-RLL input-constrained BEC is:

$$C^{\text{fb}}_{(0,k)}(\varepsilon) = \max_{0\le\delta_{k-1}\le\frac{1}{2}} \frac{\overline{\varepsilon} H_2(\delta_0) + \sum_{i=1}^{k-1}\left(\overline{\varepsilon}^{i+1} H_2(\delta_i)\prod_{m=0}^{i-1}\delta_m\right)}{1+\sum_{i=0}^{k-1}\left(\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right)}, \tag{1}$$

for all $\varepsilon \in [0, 1]$ and $k \geq 1$, where $\delta_0, \dots, \delta_{k-2}$ are simple functions of δ_{k-1}, given in Eq. (6), below. Surprisingly, we are also able to show that the non-causal knowledge of the channel erasure does not increase the feedback capacity, so that (1) is the non-causal capacity as well.

This work generalizes the results in [7], where the feedback capacity of the $(1, \infty)$-RLL input-constrained BEC was calculated.[1] In [7] and other works, [8]–[15], the capacity was derived by formulating it as a dynamic programming (DP) problem and then solving the corresponding Bellman equation. In all past works, the DP state was an element of the 1-simplex, an essential property in the solution of the Bellman equation. However, the DP state in our case is an element of the k-simplex. This makes the approach of solving the Bellman equation intractable and different methods are required.

To circumvent this difficulty, we use alternative techniques to solve the capacity of our problems. The upper bound follows from standard converse techniques for the non-causal model, where the encoder knows the erasure ahead of time. This upper bound is trivially an upper bound also for the feedback model, since non causal knowledge might increase the capacity only. Then, we construct a simple coding scheme for the feedback setting, inspired by the posterior matching principle [16]–[19]. The coding scheme enables both the encoder and the decoder to systematically reduce the size of the set of possible messages to a single message, which is then declared by the decoder as the correct message. An analysis of the achieved rate reveals an expression that is similar to the upper bound. The equivalence of these bounds is finally derived, and this concludes both the feedback capacity and the non-causal capacity for our setting.

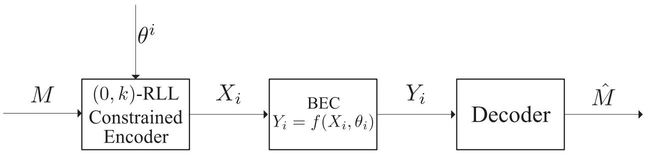

Fig. 3. Input-constrained BEC with non-causal knowledge of the erasures. The encoder has access both to the message M and to θ^i which model the erasure.

The remainder of the paper is organized as follows: Section II includes the notations we use and the problem definition. Section III contains the main results of this paper. In Section IV we present the coding scheme and its rate analysis. Section V includes an upper bound of the capacity. In Section VI we discuss the $(2, \infty)$-RLL input constraint, as an example where the non-causal capacity is strictly greater than the feedback capacity. Section VII presents the feedback capacity of the $(1, 2)$-RLL BEC, as an example for possible future avenues of research. Finally, the appendices contain proofs of several lemmas used throughout the paper.

II. Notations and Problem Definition

A. Notations

Random variables are denoted using a capital letter X. Lower-case letters x are used to denote realizations of random variables. Calligraphic letters $\mathcal{X}$ denote sets. The notation X^n means the n-tuple $(X_1, \dots, X_n)$ and x^n is a realization of such a vector of random variables. For a real number $\alpha \in [0, 1]$, we define $\overline{\alpha} := 1 - \alpha$. The binary entropy function is denoted by $H_2(\alpha) = -\alpha \log_2 \alpha - \overline{\alpha} \log_2 \overline{\alpha}$ for $\alpha \in [0, 1]$.

B. Problem Definition

The BEC (Fig. 2) is memoryless, that is $p(y_i \mid x^i, y^{i-1}) = p(y_i \mid x_i)\ \forall i$, and can be represented by:

$$y_i = \begin{cases} x_i, & \text{if } \theta_i = \checkmark \\ ?, & \text{if } \theta_i = \text{✗} \end{cases},$$

where θ^n is an i.i.d. process with $\theta_i \sim Ber(\varepsilon)$. A message M is drawn uniformly from $\{1, 2, \dots, 2^{nR}\}$ and is available to the encoder. We define two models, based on the additional information that is available to the encoder: in the first model, at time i, the encoder has knowledge of past channel outputs y^{i-1} via a noiseless feedback link (Fig. 1); in the second model, at time i, the encoder has non-causal access to θ_i (Fig. 3). In both cases, the transmission is over a BEC.

[1] The $(1, \infty)$-RLL constraint is equivalent to the $(0, 1)$-RLL constraint by swapping '0's and '1's

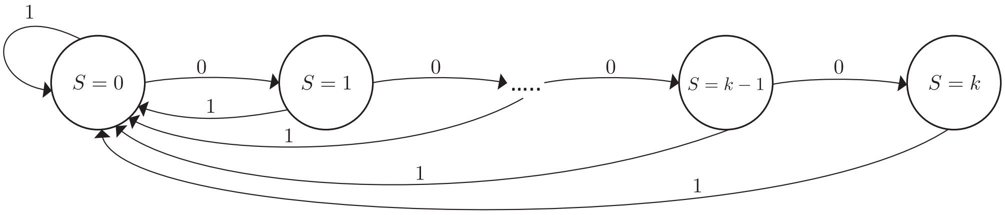

Fig. 4. State diagram describing all sequences that can be generated while satisfying the $(0,k)$-RLL constraint. Note that after k consecutive '0's the node $S=k$ is reached, which implies that the next bit is necessarily '1'.

The encoder must produce sequences that comply with the $(0,k)$-RLL input constraint. This constraint can be described graphically (Fig. 4), where all walks along the directed edges of the graph do not contain the forbidden string. Note that the node $S=k$ has only one outgoing edge, labeled '1', which implies that after k consecutive '0's, the next bit must be a '1'. The constrained encoder and the decoder operations are made precise by the following code definitions.

Definition 1. *A $(n, 2^{nR}, (0,k))$ code for an input-constrained BEC is composed of encoding and decoding functions. The encoding functions for the first model (with feedback) are:*

$$f_i : \{1,\ldots,2^{nR}\} \times \mathcal{Y}^{i-1} \to \mathcal{X}_i,\ i=1,\ldots,n, \qquad (2)$$

satisfying $f_i\left(m, y^{i-1}\right) = 1$ if $\left(f_{i-1}\left(m,y^{i-2}\right),\ldots,f_{i-k}\left(m,y^{i-k-1}\right)\right) = (0,\ldots,0)$ for all $\left(m,y^{i-1}\right)$. For the non-causal model the encoding functions are defined by:

$$g_i : \{1,\ldots,2^{nR}\} \times \{\checkmark, \text{✗}\}^i \to \mathcal{X}_i,\ i=1,\ldots,n, \qquad (3)$$

satisfying $g_i\left(m,\theta^i\right) = 1$ if $\left(g_{i-1}\left(m,\theta^{i-1}\right),\ldots,g_{i-k}\left(m,\theta^{i-k}\right)\right) = (0,\ldots,0)$ for all $\left(m,\theta^i\right)$. The decoding function for both models is defined by:

$$\Psi : \mathcal{Y}^n \to \{1,\ldots,2^{nR}\}.$$

Without loss of generality, we assume that $x_0 = 1$, so that the initial state is $s_0 = 0$.

The average probability of error for a code is defined as $P_e^{(n)} = Pr\left(M \neq \Psi(Y^n)\right)$. A rate R is said to be $(0,k)$-achievable if there exists a sequence of $(n,2^{nR},(0,k))$ codes such that $\lim_{n\to\infty} P_e^{(n)} = 0$. The capacity is defined to be the supremum over all $(0,k)$-achievable rates and is a function of k and the erasure probability ε. Denote by $C^{\text{fb}}_{(0,k)}(\varepsilon)$ the capacity of the feedback model and $C^{\text{nc}}_{(0,k)}(\varepsilon)$ that of the non-causal model. Since y^{i-1} is computable from θ^{i-1} and M, we have the relation $C^{\text{nc}}_{(0,k)}(\varepsilon) \geq C^{\text{fb}}_{(0,k)}(\varepsilon)$, for all $k \geq 1$, $\varepsilon \in [0,1]$.

III. Main Results

In this section we present the main results, including the feedback capacity and the non-causal capacity for the BEC with $(0,k)$-RLL input constraints. We then explain the methodology used to prove the results. The following theorem constitutes our main results regarding the feedback capacity and the capacity achieving coding scheme. Define the function:

$$R_\varepsilon\left(\delta_0,\ldots,\delta_{k-1}\right) = \frac{\overline{\varepsilon}H_2(\delta_0) + \sum_{i=1}^{k-1}\left(\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=0}^{i-1}\delta_m\right)}{1+\sum_{i=0}^{k-1}\left(\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right)}, \qquad (4)$$

where δ_i takes values in $[0,1]$ for $i = 0,\ldots,k-1$.

Theorem 1. *The feedback capacity of the $(0,k)$-RLL input-constrained BEC with feedback is:*

$$C^{\text{fb}}_{(0,k)}(\varepsilon) = \max_{0\leq\delta_{k-1}\leq\frac{1}{2}} R_\varepsilon\left(\delta_0,\ldots,\delta_{k-1}\right), \qquad (5)$$

where $\delta_0,\ldots,\delta_{k-2}$ are functions of δ_{k-1} and can be calculated recursively using:

$$\delta_j = \frac{\delta_{j+1}}{\delta_{j+1}+\overline{\delta}_{j+1}\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}}} \quad j=0,1,\ldots,k-2, \qquad (6)$$

with $\overline{\delta}_k := 1$. In addition, there exists a simple coding scheme that achieves the capacity given in (5).

Fig. 5 presents graphs of the feedback capacity as a function of ε for several values of k. The capacity is a decreasing function of ε, and an increasing function of k. For $\varepsilon = 0$, the channel is noiseless and so the capacity is that of the corresponding constraint. For example, $C^{\text{fb}}_{(0,1)}(0) = \log_2(\phi)$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, which is known to be the capacity of sequences that do not contain two consecutive '0's. For $\varepsilon = 1$, the output is constant so we have $C^{\text{fb}}_{(0,k)}(1) = 0$ for all k. As k increases the constraint becomes more lenient and the capacity approaches $1-\varepsilon$, which is the capacity of the unconstrained BEC. Since the constrained capacity is always upper bounded by its unconstrained counterpart, we have $C^{\text{fb}}_{(0,k)}(\varepsilon) \leq 1-\varepsilon$. When $k\to\infty$ this upper bound is achievable by choosing $\delta_j = \frac{1}{2}$, $\forall j \in \mathbb{N}$. A straightforward calculation yields $R_\varepsilon(\frac{1}{2},\frac{1}{2},\ldots) = 1-\varepsilon$. It is also pleasing to note that such a choice of δ's is compatible with the recursive relation defined in (6), $\forall j \in \mathbb{N}$. Additionally, numerical evaluations indicate that the capacity is a concave function of ε. This is a surprising observation since the capacity of a memoryless channel is known to be a convex function of the channel parameters. While it may be challenging to prove its concavity, the fact that it is non-convex can be shown by noting that the capacity is a monotone decreasing function of ε, linear and that the line between $C^{\text{fb}}_{(0,k)}(0)$ and $C^{\text{fb}}_{(0,k)}(1)$ is achievable. The

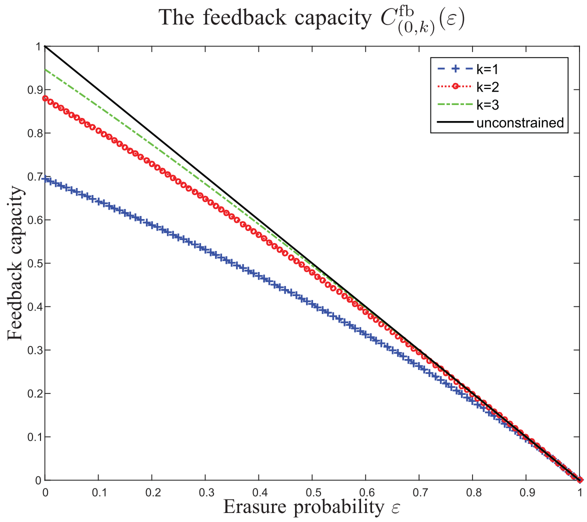

Fig. 5. Feedback capacity as a function of ε for several values of k and the unconstrained capacity. As k increases, the performance approaches that of the unconstrained channel.

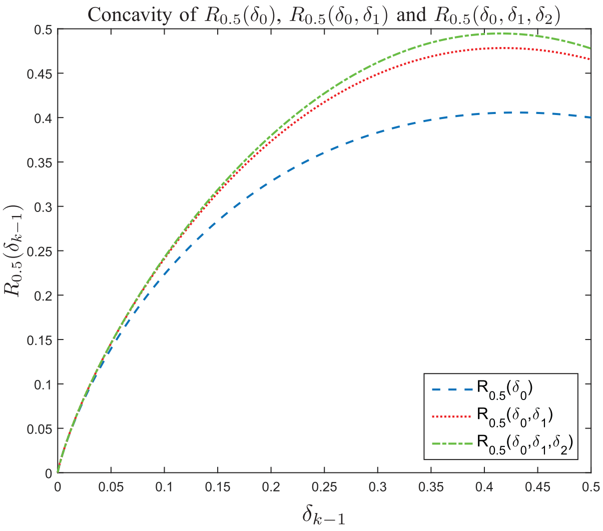

Fig. 6. Graphs of $R_{0.5}(\delta_0,\ldots,\delta_{k-1})$ as a function of δ_{k-1} for $k = 1, 2, 3$, the horizontal axis is $\delta_0, \delta_1, \delta_2$, respectively. The relations in (6) are applied to express $\delta_0,\ldots,\delta_{k-2}$ in terms of δ_{k-1}.

achievability of this linear line follows from the concatenation of a standard constrained code with a capacity-achieving code for the unconstrained erasure channel.

Fig. 6 shows numerical evaluations of $R_{0.5}(\delta_0,\ldots,\delta_{k-1})$ as a function of δ_{k-1} for $k = 1, 2, 3$. The graphs indicate that once the relations in (6) are applied, $R_{0.5}(\delta_0,\ldots,\delta_{k-1})$ is concave in δ_{k-1}. An analytical proof of concavity for any $\varepsilon \in [0, 1]$ and $k \in \mathbb{N}$ has eluded us so far.

Theorem 1 guarantees that even though the function we aim to maximize is a function of k variables, to calculate the capacity, one needs only to perform a maximization over δ_{k-1}. For any $\delta_{k-1} \in [0, 1]$, the values of all other variables can be calculated by utilizing the set of equations given in (6).

Our proposed coding scheme has k degrees of freedom, represented by $\delta_0,\ldots,\delta_{k-1}$. For this reason, it is rather surprising that the feedback capacity is a simple optimization problem of one variable for all $k \geq 1$. Indeed, the relaxation of the optimization domain shows that optimizing over the k-tuple and the optimization in (5) and (6) are equivalent. In addition we also prove the following:

Theorem 2. *Non-causal knowledge of the erasures does not increase the feedback capacity, that is $\forall k \geq 1, \varepsilon \in [0, 1]$:*

$$C^{\text{fb}}_{(0,k)}(\varepsilon) = C^{\text{nc}}_{(0,k)}(\varepsilon).$$

It is tempting to conjecture that this property holds for the general (d, k)-RLL constrained BEC, but we will provide a counterexample in Section VI. Theorems 1 and 2 both generalize parallel results shown in [7], where the special case of $k = 1$ was calculated using different techniques. The following inequalities are the main steps required to prove Theorems 1 and 2:

$$\begin{aligned}\max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq\frac{1}{2}} R_\varepsilon\left(\delta_0^{k-1}\right) &\overset{(a)}{\leq} C^{\text{fb}}_{(0,k)}(\varepsilon)\\ &\overset{(b)}{\leq} C^{\text{nc}}_{(0,k)}(\varepsilon)\\ &\overset{(c)}{\leq} \max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq 1} R_\varepsilon\left(\delta_0^{k-1}\right), \quad (7)\end{aligned}$$

where $\delta_0^{k-1} := \delta_0,\ldots,\delta_{k-1}$ and,

- Inequality (a) follows from the coding scheme that is presented in Algorithm 1. Specifically, it is shown that $R_\epsilon(\delta_0,\ldots,\delta_{k-1})$ is achievable for any choice of $\delta_i \leq 0.5,\ i = 0,\ldots,k-1$.
- Inequality (b) follows from the operational definitions of the code in Section II.
- Inequality (c) follows from standard converse techniques for the non-causal setting.

The next lemma shows that the maximal value of $R_\varepsilon(\delta_0,\ldots,\delta_{k-1})$ remains the same whether the maximization domain is $0 \leq \delta_0,\ldots,\delta_{k-1} \leq \frac{1}{2}$ or $0 \leq \delta_0,\ldots,\delta_{k-1} \leq 1$. Thus, the chain of inequalities is actually a chain of equalities.

Lemma 1. *For all $\varepsilon \in [0, 1]$ and $k \geq 1$,*

$$\begin{aligned}&\max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq\frac{1}{2}} R_\varepsilon(\delta_0,\ldots,\delta_{k-1})\\ &\qquad = \max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq 1} R_\varepsilon(\delta_0,\ldots,\delta_{k-1}).\end{aligned}$$

Moreover, the k-tuple $\arg\max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq\frac{1}{2}} R_\varepsilon(\delta_0,\ldots,\delta_{k-1})$ *satisfies Eq.* (6).

Theorems 1 and 2 are concluded from the inequalities chain (7) and Lemma 1, which is proved in Appendix A

IV. Optimal Coding Scheme for the Input-Constrained BEC with Feedback

In this section, we introduce the idea behind the optimal coding scheme, as well as the coding itself, which is presented in Algorithm 1. We then prove that the scheme is feasible, meaning that the generated input sequence does not violate the input constraint, and, finally, calculate its rate.

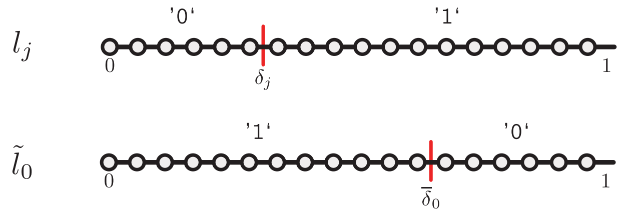

Fig. 7. Labelings used in the coding scheme, with $j = 1, \dots, k$. Each subsection of [0, 1] is labeled with '0' or '1'.

A. Coding Scheme

Before presenting the coding scheme itself, we discuss the basic ideas in accordance with which the scheme operates. The coding scheme is a mechanism that allows the encoder to transmit a message $m \in \mathcal{M} = \{1, 2, \dots, 2^{nR}\}$ to the decoder without violating the channel's input constraint. The main feature of the scheme is a dynamic set of possible messages that is known both to the encoder and to the decoder at all times. Both parties will systematically reduce the size of the set of possible messages from 2^{nR} in the beginning of the transmission process to a single message that will then be announced as the correct message.

Initially, the messages are mapped uniformly to message points in the unit interval by applying $m \mapsto \frac{m-1}{2^{nR}}$. As long as transmission proceeds, the set of possible messages is represented by uniform points on the unit interval with proper scaling.

Channel inputs are determined by $k+2$ *labeling* functions, which map the unit interval into $\mathcal{X}$. Given a labelling, l_j, with a corresponding parameter δ_j the encoder assigns the label '0' to a subsection of [0, 1] of length δ_j and the label '1' to the rest of [0, 1]. Recall that $\delta_k \coloneqq 0$, so the label l_k assigns the label $'1'$ to the entire unit interval. Fig. 7 depicts the various labelings. Define the following function:

$$X(m, L) = \begin{cases} 0, & (L = \tilde{l}_0 \text{ and } m > \bar{\delta}_0) \text{ or} \\ & (L = l_j \text{ and } m < \delta_j,\ j = 0, \dots, k) \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

The channel input is $X_i = X(m, L_i)$, where m is the correct message point and L_i is the labeling being used at time i.

The labelling at each time is a function of all channel outputs and can be computed recursively from the previous channel output and the previous labelling. Therefore, the instantaneous labelling is available both to the encoder and the decoder. Transition between the various labelings is controlled by a finite-state machine (FSM), which is illustrated in Fig. 8. Define the following function:

$$G(L, Y) = \begin{cases} l_0, & (L = l_k) \text{ or } (Y = 1) \text{ or} \\ & (Y = ? \text{ and } L = \tilde{l}_0) \\ \tilde{l}_0, & Y = ? \text{ and } L \neq \tilde{l}_0 \\ l_{j+1}, & Y = 0 \text{ and } (L = \tilde{l}_0 \text{ or} \\ & L = l_j, j = 0, \dots, k-1) \end{cases}, \tag{9}$$

and thus, $L_{i+1} = G(L_i, Y_i)$. The chronological order of a single channel use is as follows: $L_i = l_j \rightarrow X_i = X(m, l_j) = x_i \rightarrow Y_i = BEC(\varepsilon, x_i) = y_i \rightarrow L_{i+1} = G(l_j, y_i)$.

A transmission at time i is said to be successful if $Y_i \neq ?$. Due to the nature of the BEC, whenever $Y_i \neq ?$ the decoder can know with certainty the value of X_i. Denote by $\hat{M}_i^{(0)}$ and $\hat{M}_i^{(1)}$ the subsets of messages which are labeled at time i with '0' and '1', respectively. Define $\hat{M}_0 = \mathcal{M}$ and for $i \geq 1$:

$$\hat{M}_i = \begin{cases} \hat{M}_{i-1}, & Y_i = ? \\ \hat{M}_{i-1}^{(0)}, & Y_i = 0 \\ \hat{M}_{i-1}^{(1)}, & Y_i = 1. \end{cases} \tag{10}$$

Thus, a successful transmission reduces the size of the set of possible messages. Following a successful transmission, the remaining messages in the set of possible messages are uniformly mapped again to $[0, 1)$. Fig. 9 depicts a successful transmission and the subsequent reduction of the number of possible messages. The process continues until such a time that the set of possible messages contains only one message, at which point the decoder declares it to be $\hat{m}$. The proposed scheme relies on the posterior matching principle which essentially imitates the optimal inputs distribution [17] in each step. This principle has been shown to result in capacity-achieving coding schemes for all memoryless channels [17], [18], [20]. Capacity-achieving matching schemes for channels with memory appeared in several instances. For example, in channels where the state is computed at the encoder and the decoder [21] and in input constrained memoryless channels [7], [15].

Algorithm 1 presents the coding scheme in pseudo code form. The functions $X(\cdot, \cdot)$ and $G(\cdot, \cdot)$ mentioned in the algorithm are defined in Eq. (8) and Eq. (9), respectively.

Algorithm 1 Coding Scheme

```
Inputs: $m$ - correct message
$\hat{\mathcal{M}} = \mathcal{M}$
Label $= l_0$
while $|\hat{M}| > 1$ do
    Transmit $X(m, \text{Label})$    %% Encoder operation
    if $Y = 0$ then
        $\hat{M} = \hat{M}^{(0)}$
    else if $Y = 1$ then
        $\hat{M} = \hat{M}^{(1)}$
    end if
    Label $= G(\text{Label}, Y)$
end while
$\hat{m} = \hat{\mathcal{M}}$    %% Decoder operation
```

B. Feasibility of the Proposed Scheme

First, we show that the coding scheme satisfies the $(0, k)$-RLL constraint, that is, no message is mapped by the scheme into a sequence with more than k consecutive '0's. The following lemma shows that the constraint is satisfied when restricting the scheme parameters δ_j.

Lemma 2. *If $\delta_j \leq \frac{1}{2}$ for $j = 0, \dots, k-1$, then any channel input sequence generated by the proposed coding scheme satisfies the $(0, k)$-RLL constraint.*

Proof. We show that if $\delta_j \leq \frac{1}{2}$ for $j = 0, \dots, k-1$, then no message is labeled '0' more than k times in a row.

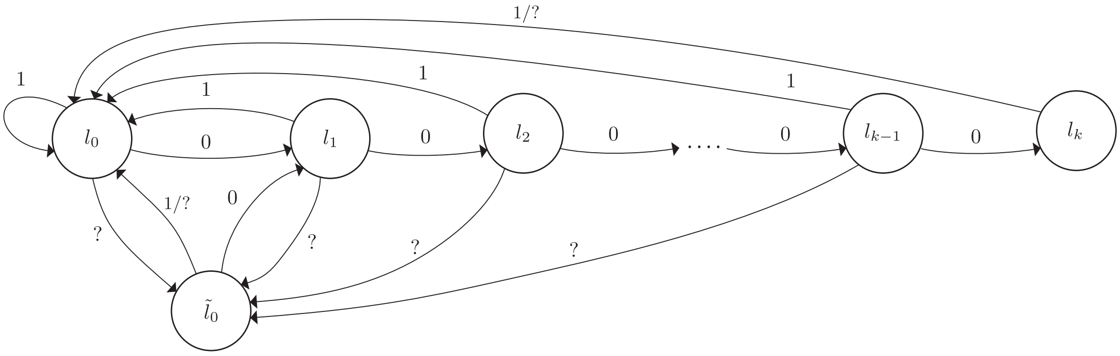

Fig. 8. Finite state machine for the labelings transition. The nodes describe the instantaneous labelling that is used by the encoder. Edges correspond to channel outputs. Each node in the diagram corresponds to a labelling that can be calculated both at the encoder and at the decoder, since edges are a function of the outputs.

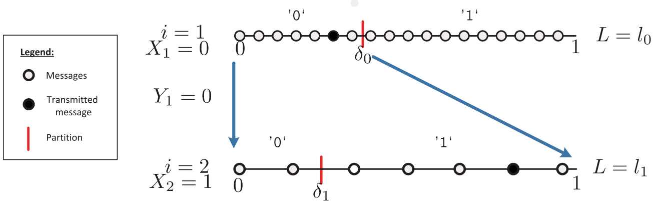

Fig. 9. Example of a successful transmission. The black dot is the correct message point. At time instance $i = 1$, the labeling is $L = l_0$ and the message point is labeled with ‘0’ since it lies within $[0, \delta_0)$; thus, the encoder transmits $X_1 = 0$. Assume that $Y_1 = 0$. Consequently, the messages that were labeled with ‘1’ are discarded, and the remaining messages are repositioned uniformly across [0, 1]. These messages are $\hat{\mathcal{M}}_1$. In accordance with the FSM in Fig. 8, the next label is $L = l_1$. At $i = 2$, the message point is labeled with ‘1’ since it now lies within $[\delta_1, 1)$; thus, the encoder will transmit $X_2 = 1$.

From Eqs. (8) and (9), the channel input, X_i is a function of m and y_{i-k-1}^{i-1}. Therefore, we divide the proof into three disjoint cases based on the k last outputs y_{i-k-1}^{i-1}. For each case, we show that the subsequent sequence of channel inputs cannot contain more than k consecutive ‘0’s. Assume that transmission begins at the node associated with labeling l_0.

1) Any output sequence (of length k) that contains a ‘1’ cannot cause a violation.
2) An output sequence of k consecutive ‘0’s ends at l_k (Fig. 8). Thus, the next channel input is $X = 1$ (Fig. 7).
3) Lastly, consider a sequence of k outputs that contains both ‘0’s and erasures. Assume the first erasure occurred at time instance i, meaning that the erasure took place while the encoder was using labeling l_i. This means that all messages between 0 and δ_i in $[0, 1)$ were labeled with ‘0’ $i+1$ times in a row. The next labeling that will be used is $\tilde{l}_0$. In this labeling, all messages between 0 and $\overline{\delta}_0$ are labeled with ‘1’. Since $\delta_0 \leq \frac{1}{2}$, we have that $\overline{\delta}_0 \geq \frac{1}{2}$, so all messages that were labeled ‘0’ $i+1$ times in a row will be labeled ‘1’ in $\tilde{l}_0$. This analysis holds for any $i = 0, \dots, k-1$.

In summary, setting $\delta_i \leq \frac{1}{2}$, $i = 0, \dots, k-1$ ensures that the scheme does not violate the $(0, k)$-RLL constraint. □

C. Rate Analysis

The achieved rate R is measured by $\frac{\text{expected number of information bits}}{\text{expected number of channel uses}}$. Define Q as the number of information bits gained in a single channel use, i.e., the quotient of the size of the set of possible messages before and after the transmission. Additionally, let L be the random variable which corresponds to the labeling and takes values in $\mathcal{L} = \{\tilde{l}_0, l_0, \dots, l_k\}$.

In the following lemma we calculate the expectation of $Q \mid L = l$.

Lemma 3. *For all $l \in \mathcal{L}$, we have that $\mathbb{E}[Q \mid L = l] = \overline{\varepsilon} H_2(\delta_l)$, where δ_l is the δ relevant to the labeling l.*

Proof. Consider:

$$\mathbb{E}[Q \mid L = l] = \varepsilon \mathbb{E}[Q \mid L = l, \theta = \text{✗}] + \overline{\varepsilon}\mathbb{E}[Q \mid L = l, \theta = \checkmark]$$
$$\overset{(a)}{=} \overline{\varepsilon}\mathbb{E}[Q \mid L = l, \theta = \checkmark], \quad (11)$$

where (a) holds since if $\theta = $ ✗, then the transmitted symbol is erased by the channel and the set of possible messages is unchanged.

In the proposed coding scheme, labeling l_j assigns a portion of [0, 1] of size δ_j to the label ‘0’ and the rest to label ‘1’ for all $j = 0, \dots, k$. The labeling $\tilde{l}_0$ also assigns δ_0 of the unit interval to ‘0’. If the labeling l_j is employed, then the channel input is distributed according to $Ber(\overline{\delta}_j)$[2]

Assume that $|\hat{\mathcal{M}}| = a$. If the successfully received bit was ‘1’, then the new set of possible messages has size $\overline{\delta}_l a$, and if it was ‘0’, then the new set of possible messages has size $\delta_l a$. The expected number of bits required to describe the new set of possible messages is $\overline{\delta}_l \log_2(\overline{\delta}_l a) + \delta_l \log_2(\delta_l a) = \log_2(a) - H_2(\delta_l)$. Thus, given that $L = l$, following a successful transmission the decoder gains $H_2(\delta_l)$ bits of information. Substituting into (11) we get:

$$\mathbb{E}[Q \mid L = l] = \overline{\varepsilon} H_2(\delta_l).$$

□

[2] For labelings $\tilde{l}_0, l_0, l_1, \dots, l_{k-1}$, the encoder transmits $X = 1$ if the correct message falls within a sub-interval of [0, 1) that has length $\overline{\delta}_0, \overline{\delta}_0, \overline{\delta}_1, \dots, \overline{\delta}_{k-1}$, respectively. Note that the messages are discrete points in [0, 1) and it is possible for the partition to occur between two messages. This implies that the transmitted bit is distributed $Ber(\overline{\delta}_i + e_i)$, where e_i is a correction factor. From the continuity of the entropy function, the contribution of this correction factor can be bounded with arbitrary constant by taking the block length n be large enough. The precise details are omitted and follow parallel argument to [7, Appendix C].

The next lemma calculates the rate achieved by the proposed coding scheme.

Lemma 4. *For any $\varepsilon \in [0, 1]$, $k \geq 1$ and $0 \leq \delta_0, \ldots, \delta_{k-1} \leq \frac{1}{2}$, the proposed coding scheme achieves the following rate R:*

$$R = \frac{\overline{\varepsilon} H_2(\delta_0) + \sum_{i=1}^{k-1} \left(\overline{\varepsilon}^{i+1} H_2(\delta_i) \prod_{m=0}^{i-1} \delta_m \right)}{1 + \sum_{i=0}^{k-1} \left(\overline{\varepsilon}^{i+1} \prod_{m=0}^{i} \delta_m \right)}. \tag{12}$$

Proof. Consider the averaged gain of information divided by the amount of time:

$$\begin{aligned} R &= \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}[Q_j] \\ &\overset{(a)}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \sum_{l\in\mathcal{L}} P(L_j = l) \mathbb{E}[Q_j \mid L_j = l] \\ &\overset{(b)}{=} \sum_{l\in\mathcal{L}} \overline{\varepsilon} H_2(\delta_l) \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} P(L_j = l) \\ &\overset{(c)}{=} \sum_{l\in\mathcal{L}} \overline{\varepsilon} H_2(\delta_l) \pi(l) \\ &= \overline{\varepsilon} H_2(\delta_0) \left(\pi(\tilde{l}_0) + \pi(l_0) \right) + \sum_{i=1}^{k-1} \overline{\varepsilon} H_2(\delta_i) \pi(l_i) \\ &\overset{(d)}{=} \frac{\overline{\varepsilon} H_2(\delta_0) + \sum_{i=1}^{k-1} \left(\overline{\varepsilon}^{i+1} H_2(\delta_i) \prod_{m=0}^{i-1} \delta_m \right)}{1 + \sum_{i=0}^{k-1} \left(\overline{\varepsilon}^{i+1} \prod_{m=0}^{i} \delta_m \right)}, \end{aligned}$$

where

(a) Follows from the law of total expectation.

(b) Follows from Lemma 3 and exchanging the finite sums' order.

(c) Follows from the definition of stationary probability and Cesaro mean. $\pi(l_i)$ is the stationary probability of labeling l_i. There exists a stationary probability because the random process $\{L_j\}$ is a positive recurrent, irreducible and aperiodic Markov chain, as can be seen from Fig. 8.

(d) Follows from calculation of the stationary probability of the Markov chain described in Fig. 8 and is parameterized with δ_j, $j = 0, \ldots, k-1$. Calculating the conditional probability of each edge is simple, using the law of total probability. For example, the conditional distribution of the edge beginning node l_0 and culminating in node l_1 is $\overline{\varepsilon}\delta_0$.

From Lemma 2, we conclude that $\max_{0 \leq \delta_0, \ldots, \delta_{k-1} \leq \frac{1}{2}} R_\varepsilon(\delta_0, \ldots, \delta_{k-1})$ is an achievable rate. □

Remark 1. The presented scheme appears as variable-length coding with a slightly loose rate analysis. A precise analysis for the current scheme in the fixed-block regime can be directly deduced from [15]. Specifically, it can be converted into a fixed block coding that has two transmission stages. The first stage is based on the current scheme and is used to transmit *most* of the message bits with rate that is arbitrarily close to the capacity. The second stage serves as a zero-rate refinement of a "small" set of suspected messages. We refer the reader to an exhaustive study of such schemes done in [15], where an input-constrained binary symmetric channel is investigated in the presence of feedback. Indeed, it covers a larger family of channels with memory including the setting discussed in the current paper.

V. Non-Causal Upper Bound

In this section, we present an upper bound of the non-causal capacity for the $(0, k)$-RLL input-constrained BEC (given in Eq. (7)). We begin with an observation: it is sufficient to look at a smaller family of codes, called *restricted codes*. We then proceed to calculate an upper bound on the achievable rate of such codes, using standard converse arguments, as well as the method of types and Markov theory results.

A code is said to be *restricted* if

$$g_i(m, \theta^{i-1}, \theta_i = \text{✗}) = 1, \ \forall m, \theta^{i-1}, i = 1, \ldots, n. \tag{13}$$

Condition (13) states that if an erasure is about to occur, the encoder transmits $X = 1$. The following lemma formalizes the fact that restricted codes can achieve the capacity.

Lemma 5. *For the $(0, k)$-RLL constrained non-causal BEC, if a rate R is achievable, then R can be achieved using a sequence of restricted codes. Proof. Assume the rate R is achieved using a sequence of codes: C_n. Define for each n a new code C'_n that is exactly the same as C_n except that in C'_n, whenever $\theta_i = \text{✗}$ the encoder transmits $x_i = 1$.*

The code C'_n does not violate the input constraint since the original C_n did not violate the constraint and transmitting $x_i = 1$ is always permitted by the $(0, k)$-RLL input constraint. In addition, the channel outputs remain the same whether the code is C_n or C'_n. This means that $P_e^{(n)}(C'_n) = P_e^{(n)}(C_n)$, and so the rate R is also achieved by the sequence of C'_n. □

A. Upper Bound Calculation

The following technical lemma is needed for the converse proof:

Lemma 6. *For any n-tuple constrained distribution $\tilde{P}_{Y^n}(y^n) = \mathbb{1}_{\{y_1=1\}} \prod_{i=2}^{n} \tilde{P}^{(i)}_{Y_i|Y_{i-1}}(y_i \mid y_{i-1})$, where $\tilde{P}^{(i)}_{Y_i|Y_{i-1}}(y_i = 0 \mid y_{i-1} = 0) = 0 \ \forall i = 2, \ldots, n$, there exists a time invariant constrained Markov distribution $\tilde{Q}_{Y^n}(y^n) = \prod_{i=1}^{n} \tilde{P}_{Y_i|Y_{i-1}}(y_i \mid y_{i-1})$ such that:*

$$H_{\tilde{P}}(Y^n) \leq H_{\tilde{Q}}(Y^n) + \zeta_n, \tag{14}$$

where $\lim_{n\to\infty} \zeta_n = 0$.

The proof is available in Appendix B. This result can readily be generalized to any (d, k)-RLL constraint imposed on the n-tuple distribution.

In the constraint graph, Fig. 4, a node S_i can be calculated from an any-length tuple X^{i-1} by walking along the edges labelled with X^{i-1}, since we assume that the initial state is $s_0 = 0$. The notation $\tilde{P}$ will be used in various forms for distributions on $\mathcal{Y}$ to signify that the probability for ? is ε and that the probability for a constrained word is 0.

Proof of the Upper Bound Let R be an achievable rate using a restricted code, and consider the following chain of inequalities:

$$
\begin{aligned}
nR &= H(M) \\
&\overset{(a)}{\leq} I(Y^n; M) + \varepsilon_n \\
&\overset{(b)}{=} H(Y^n) - \sum_{i=1}^{n} H(Y_i \mid M, Y^{i-1}) + \varepsilon_n \\
&\overset{(c)}{\leq} H(Y^n) - nH_2(\varepsilon) + \varepsilon_n \\
&\overset{(d)}{\leq} \sum_{i=1}^{n} H(Y_i \mid Y_{i-k}^{i-1}) - nH_2(\varepsilon) + \varepsilon_n \\
&\overset{(e)}{\leq} \max_{\{\tilde{P}_i(y_i|y^{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid Y_{i-k}^{i-1}) - nH_2(\varepsilon) + \varepsilon_n \\
&\overset{(f)}{=} \max_{\{\tilde{P}_i(y_i|y_{i-k}^{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid Y_{i-k}^{i-1}) - nH_2(\varepsilon) + \varepsilon_n \\
&\overset{(g)}{\leq} \max_{\{\tilde{P}(y_i|y_{i-k}^{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid Y_{i-k}^{i-1}) - nH_2(\varepsilon) + \varepsilon'_n \\
&\overset{(h)}{=} \max_{\{\tilde{P}(y_i|y_{i-k}^{i-1}, s_{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid Y_{i-k}^{i-1}, S_{i-1}) - nH_2(\varepsilon) \\
&\qquad + \varepsilon'_n \\
&\overset{(i)}{\leq} \max_{\{\tilde{P}(y_i|y_{i-k}^{i-1}, s_{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid S_{i-1}) - nH_2(\varepsilon) + \varepsilon'_n \\
&\overset{(j)}{=} \max_{\{\tilde{P}(y_i|s_{i-1})\}_{i=1}^n} \sum_{i=1}^{n} H(Y_i \mid S_{i-1}) - nH_2(\varepsilon) + \varepsilon'_n \\
&\overset{(k)}{=} \max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq 1} \sum_{i=1}^{n} \sum_{j=0}^{k-1} \Pr(S_{i-1} = j) H_3(\overline{\varepsilon}\delta_j, \varepsilon, \overline{\varepsilon}\overline{\delta}_j) \\
&\qquad - nH_2(\varepsilon) + \varepsilon'_n \\
&\overset{(l)}{=} \max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq 1} \sum_{i=1}^{n} \sum_{j=0}^{k-1} \Pr(S_{i-1} = j) \left[H_2(\varepsilon) + \overline{\varepsilon} H_2(\delta_j)\right] \\
&\qquad - nH_2(\varepsilon) + \varepsilon'_n \\
&= \max_{0\leq\delta_0,\ldots,\delta_{k-1}\leq 1} \overline{\varepsilon} \sum_{j=0}^{k-1} H_2(\delta_j) \sum_{i=1}^{n} \Pr(S_{i-1} = j) + \varepsilon'_n.
\end{aligned}
\tag{15}
$$

where

(a) Follows from Fano's inequality.

(b) Follows from the chain rule.

(c) Follows from the fact that conditioning reduces entropy, so: $H(Y_i \mid M, Y^{i-1}) \geq H(Y_i \mid X_i, M, Y^{i-1}) = H_2(\varepsilon)$.

(d) Follows from the fact that conditioning reduces entropy.

(e) The maximization domain is the set of all n-tuple distributions $\tilde{P}(y^n)$ which induce $\tilde{P}(y_i = ? \mid y^{i-1}) = \varepsilon$ and $\tilde{P}(y_i = 0 \mid y_{i-k}^{i-1} = 0^k) = 0$, for all $i = 1, \ldots, n$ and $i = k+1, \ldots, n$, respectively.

(f) We want to show that it is possible to maximize over a smaller domain and maintain an equality. It suffices to prove by induction that if we have two distributions $\{\tilde{P}^{(1)}(y_i \mid y^{i-1})\}_{i\geq 1}$ and $\{\tilde{P}^{(2)}(y_i \mid y^{i-1})\}_{i\geq 1}$, which induce the same marginal distributions $\{\tilde{P}(y_i \mid y_{i-k}^{i-1})\}_{i\geq 1}$, then $\{\tilde{P}^{(1)}(y_{i-k}^{i})\}_{i\geq 1}$ and $\{\tilde{P}^{(2)}(y_{i-k}^{i})\}_{i\geq 1}$ coincide. For $i = 1$ the proof is trivial. Assume by induction that $\tilde{P}^{(1)}(y_{i-1-k}^{i-1}) = \tilde{P}^{(2)}(y_{i-1-k}^{i-1})$ and we need to prove that $\tilde{P}^{(1)}(y_{i-k}^{i}) = \tilde{P}^{(2)}(y_{i-k}^{i})$. Indeed we have:
$$
\begin{aligned}
\tilde{P}^{(1)}(y_{i-k}^{i}) &= \tilde{P}^{(1)}(y_{i-k}^{i-1})\tilde{P}(y_i \mid y_{i-k}^{i-1}) \\
&= \tilde{P}^{(2)}(y_{i-k}^{i-1})\tilde{P}(y_i \mid y_{i-k}^{i-1}) = \tilde{P}^{(2)}(y_{i-k}^{i})'
\end{aligned}
$$
since $\tilde{P}(y_i \mid y_{i-k}^{i-1})$ is the same for both distributions by assumption, and the induction assumption tells us that $\tilde{P}^{(1)}(y_{i-1-k}^{i-1}) = \tilde{P}^{(2)}(y_{i-1-k}^{i-1})$, and thus we have $\tilde{P}^{(1)}(y_{i-k}^{i-1}) = \tilde{P}^{(2)}(y_{i-k}^{i-1})$ as well. Additionally, it can easily be shown that $\tilde{P}(y_i = ? \mid y_{i-k}^{i-1}) = \varepsilon$ and $\tilde{P}(y_i = 0 \mid y_{i-k}^{i-1} = 0^k) = 0$, for all $i = 1, \ldots, n$ and $i = k+1, \ldots, n$, respectively.

(g) Follows from Lemma 6. Notice that the distributions in the maximization domain are now time-invariant.

(h) Follows from the fact that S_{i-1} is a function of Y_{i-k}^{i-1}: since the code is restricted, X_{i-k}^{i-1} is a function of Y_{i-k}^{i-1} and, by its definition, S_{i-1} is a function of X_{i-k}^{i-1}.

(i) Follows from the fact that conditioning reduces entropy.

(j) Similarly to step (f), it suffices to prove by induction that if we have two distributions $\{\tilde{P}^{(1)}(y_i \mid y_{i-k}^{i-1}, s_{i-1})\}_{i=1}^n$ and $\{\tilde{P}^{(2)}(y_i \mid y_{i-k}^{i-1}, s_{i-1})\}_{i=1}^n$ which induce the same marginal distributions $\{\tilde{P}(y_i \mid s_{i-1})\}_{i=1}^n$, then $\{\tilde{P}^{(1)}(y_{i-k}^{i}, s_{i-1}^{i})\}_{i=1}^n$ and $\{\tilde{P}^{(2)}(y_{i-k}^{i}, s_{i-1}^{i})\}_{i=1}^n$ coincide. Recall that since the code is restricted, s_i is a function of (s_{i-1}, y_i). denote this function by $s_i = h(s_{i-1}, y_i)$. For $i = 1$ we have:
$$
\begin{aligned}
\tilde{P}^{(1)}(y_1, s_0^1) &= \mathbb{1}_{\{s_0=0\}}\tilde{P}(y_1 \mid s_0)\tilde{P}^{(1)}(s_1 \mid y_1, s_0) \\
&= \mathbb{1}_{\{s_0=0\}}\tilde{P}(y_1 \mid s_0)\mathbb{1}_{s_1=h(s_0,y_1)} \\
&= \tilde{P}^{(2)}(y_1, s_0^1).
\end{aligned}
$$
Now, assume by induction that $\tilde{P}^{(1)}(y_{i-1-k}^{i-1}, s_{i-2}^{i-1}) = \tilde{P}^{(2)}(y_{i-1-k}^{i-1}, s_{i-2}^{i-1})$ and we need to prove that $\tilde{P}^{(1)}(y_{i-k}^{i}, s_{i-1}^{i}) = \tilde{P}^{(2)}(y_{i-k}^{i}, s_{i-1}^{i})$:
$$
\begin{aligned}
\tilde{P}^{(1)}(y_{i-k}^{i}, s_{i-1}^{i}) &= \tilde{P}^{(1)}(y_{i-k}^{i-1}, s_{i-1})\tilde{P}^{(1)}(y_i \mid y_{i-k}^{i-1}, s_{i-1}) \\
&\quad \tilde{P}^{(1)}(s_i \mid y_{i-k}^{i}, s_{i-1}) \\
&\overset{(1)}{=} \tilde{P}^{(2)}(y_{i-k}^{i-1}, s_{i-1})P(y_i \mid s_{i-1}) \\
&\quad \mathbb{1}_{\{s_i=h(s_{i-1},y_i)\}} \\
&= \tilde{P}^{(2)}(y_{i-k}^{i}, s_{i-1}^{i}),
\end{aligned}
$$
where (1) follows from the induction assumption, the Markov chain $y_i - s_{i-1} - y_{i-k}^{i-1}$ and the notation defined above.

(k) Follows by defining a conditional distribution, $\delta_j \triangleq p(X = 0 \mid S = j, \theta = \checkmark)$.

(l) Follows from a simple identity.

For each instance of the tuple $(\delta_0, \ldots, \delta_{k-1})$, the random process $\{S_i\}_{i=1}^n$ is first-order Markov. Additionally, for all tuples, there is a single closed communicating class for this process, so there exists a stationary distribution and the value of $\sum_{i=1}^n \Pr(S_i = j)$ can be made arbitrarily close to $n\pi_S(j)$, where $\pi_S(j)$ denotes the stationary distribution that is induced by the Markov chain in Fig. 4. Using the transitions matrix of the Markov process $\{S_i\}_{i=1}^n$:

S_{i-1} \ S_i	$S=0$	$S=1$	$S=2$	$S=3$	$\ldots$	$S=k$
$S=0$	$\varepsilon+\bar{\varepsilon}\delta_0$	$\bar{\varepsilon}\bar{\delta}_0$	0	0	$\ldots$	0
$S=1$	$\varepsilon+\bar{\varepsilon}\delta_1$	0	$\bar{\varepsilon}\bar{\delta}_1$	0	$\ldots$	0
$S=2$	$\varepsilon+\bar{\varepsilon}\delta_2$	0	0	$\bar{\varepsilon}\bar{\delta}_2$	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$S=k-1$	$\varepsilon+\bar{\varepsilon}\delta_{k-1}$	0	0	0	$\ldots$	$\bar{\varepsilon}\bar{\delta}_{k-1}$
$S=k$	1	0	0	0	$\ldots$	0

we can show that:

$$\pi_S(j) = \frac{\overline{\varepsilon}^j \prod_{m=0}^{j-1} \delta_m}{1 + \sum_{j=0}^{k-1} \left(\overline{\varepsilon}^{j+1} \prod_{m=0}^{j} \delta_m\right)}, \; j = 0, \ldots, k-1,$$

where $\prod_{m=0}^{-1} \delta_m \triangleq 1$. Therefore, we have that

$$R \le (1-\varepsilon) \max_{\{p_i(x|s,\theta=\checkmark)\}} \sum_{j=0}^{k-1} H_2(\delta_j)[\pi_S(j) + \varepsilon_n''] + \frac{\epsilon_n'}{n},$$

where ε_n'' is the correcting factor from the stationary distribution and satisfies $\varepsilon_n'' \to 0$. By taking the limit $n \to \infty$, and substituting the stationary distribution, we conclude that an achievable rate is upper bounded by

$$C_{(0,k)}^{\text{nc}} \le \max_{0 \le \delta_0, \ldots, \delta_{k-1} \le 1} R_\varepsilon(\delta_0, \ldots, \delta_{k-1}). \tag{16}$$

□

VI. Does $C^{\text{fb}} = C^{\text{nc}}$ for Any Input Constraint?

In previous sections it was shown that $C_{(0,k)}^{\text{fb}}(\varepsilon) = C_{(0,k)}^{\text{nc}}(\varepsilon)$. This section concerns the ensuing question: is it true that non-causal knowledge of the upcoming erasure does not increase the feedback capacity for any input constraint?

It turns out that the non-causal capacity of the (d, ∞)-RLL case can be easily solved using the same arguments as in previous sections. Therefore, we investigated this family with a hope to prove its feedback capacity as well. For $d = 1$, it has been proven in [7] that $C_{(1,\infty)}^{\text{fb}}(\varepsilon) = C_{(1,\infty)}^{\text{nc}}(\varepsilon)$. This result coincides with Theorem 2 since the $(1, \infty)$ constraint is equivalent to the $(0, 1)$ constraint by swapping '1's and '0's. However, for d=2, we are able to show that $C_{(2,\infty)}^{\text{fb}}(\varepsilon) < C_{(2,\infty)}^{\text{nc}}(\varepsilon)$. Thus, the answer to the aforementioned question is no.

The first result is the non-causal capacity of the BEC with a (d, ∞)-RLL input constraint, for any $d \ge 1$.

Lemma 7. *For any $d \in \mathbb{N}$, the non-causal capacity of the (d, ∞)-RLL input constrained BEC is given by:*

$$C_{(d,\infty)}^{\text{nc}}(\varepsilon) = \max_{0 \le \delta \le \frac{1}{2}} \frac{H_2(\delta)}{\frac{1}{1-\varepsilon} + d\delta}. \tag{17}$$

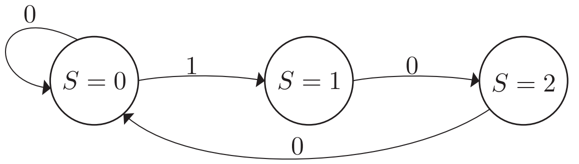

Fig. 10. State diagram describing all sequences that can be generated while satisfying the $(2, \infty)$-RLL constraint: every '1' is followed by at least two '0's.

Proof. The upper bound of $C_{(d,\infty)}^{\text{nc}}(\varepsilon)$ is derived following the same steps presented in Section V. In this case, a restricted encoder that transmits $X = 0$ whenever $\theta = \text{✗}$. The rest of the proof mirrors that of Section V, and we are able to show that $C_{(d,\infty)}^{\text{nc}}(\varepsilon) \le \max_{0 \le \delta \le \frac{1}{2}} \frac{H_2(\delta)}{\frac{1}{1-\varepsilon} + d\delta}$. This expression is also a lower bound. It is achieved by applying a restricted encoder which transmits $X \sim Ber(\delta)$ if an erasure does not occur. The expected number of information bits gained in a successful transmission is $H_2(\delta)$ and the expected number of channel uses to transmit successfully is $\frac{1}{1-\varepsilon}$, plus another d channel uses if the transmitted bit is a '1'. □

Next we prove that $C_{(2,\infty)}^{\text{fb}}(\varepsilon)$ is upper bounded by an expression which is strictly smaller than the RHS of Eq. (17) for $d = 2$. To discuss an upper bound for $C_{(2,\infty)}^{\text{fb}}(\varepsilon)$, we must first introduce the concepts of the S-graph and the Q-graph. Fig. 10 contains an S-graph, which is simply a graphical representation of the $(2, \infty)$-RLL constraint. A Q-graph is an irreducible directed graph in which each node has $|\mathcal{Y}|$ distinct outgoing edges. The upper bound is derived using the method introduced in [22]. This method involves a combined representation of both the S-graph and the Q-graph in a coupled (S, Q)-graph, which has a stationary distribution denoted $\pi(s, q)$. The main result in [22] states the following:

Theorem 3 (Theorem 2, [22])**.** *For every Q-graph, the feedback capacity is bounded by*

$$C^{\text{fb}} \le \sup_{p(x|s,q)} I(X; Y \mid Q),$$

where S represents the input constraint state. The joint distribution is $\pi(s, q)p(x \mid s, q)p(y \mid x, s)$.

We apply Theorem 3 with the Q-graph in Fig. 11. This graph was estimated from numerical evaluations of the associated DP problem. Calculating $\sup_{p(x|s,q)} I(X; Y \mid Q)$ we get:

$$C_{(2,\infty)}^{\text{fb}}(\varepsilon) \le \max_{\substack{0 \le \delta_0, \delta_1, \delta_2 \le 1 \\ \delta_0 + \delta_1 + \delta_2 \le 1}} \frac{\overline{\varepsilon}\left(H_2(\delta_0) + \varepsilon H_2(\delta_1) + \varepsilon^2 H_2(\delta_2)\right)}{1 + \varepsilon + \varepsilon^2 + 2\overline{\varepsilon}(\delta_0 + \varepsilon\delta_1 + \varepsilon^2\delta_2)}. \tag{18}$$

Fig. 12 contains graphs of the non-causal capacity and the feedback upper bound for $0 \le \varepsilon \le 1$. It is clear that the non-causal capacity is strictly greater than the feedback upper bound in the case of the $(2, \infty)$-RLL input constrained BEC. The following lemma states the strong inequality for a specific ε:

Lemma 8. *For $\varepsilon = \frac{1}{2}$, non-causal knowledge of the erasure does increase the feedback capacity, that is:*

$$C_{(2,\infty)}^{\text{fb}}(\tfrac{1}{2}) < C_{(2,\infty)}^{\text{nc}}(\tfrac{1}{2}). \tag{19}$$

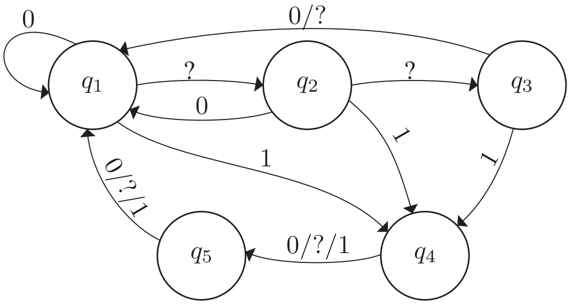

Fig. 11. Q-graph for the $(2,\infty)$-RLL BEC

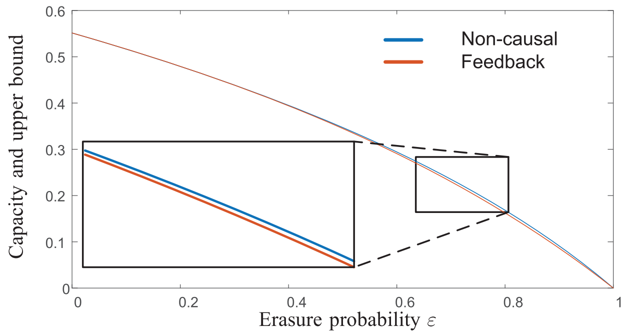

Fig. 12. Non-causal capacity and feedback upper bound for the $(2,\infty)$-RLL input constrained BEC, as a function of ε. The non-causal capacity is greater than the upper bound of the feedback capacity. Note that $C^{\text{nc}}_{(2,\infty)}(0) = C^{\text{nc}}_{(2,\infty)}(0) \sim 0.551$, which is the $(2,\infty)$-RLL constraint capacity.

Proof. By partially deriving the RHS of (18), the only critical point in the compact domain $\{(\delta_0,\delta_1,\delta_2)\in\mathbb{R}^3|0\le \delta_0,\delta_1,\delta_2\le 1\}$ is $\delta\triangleq\delta_0=\delta_1=\delta_2$. Substituting δ into (18) gives the objective of (17), so all that is left to show is that the argument which achieves the maximum in (17) is greater than $\frac{1}{3}$. For $\varepsilon=\frac{1}{2}$, one can show that the maximum of (17) is obtained at $\frac{1}{3}<\delta<\frac{1}{2}$. This means that the local maximum of (17) is located outside the maximization domain of (18). Additional tedious calculations also reveal that (17) on its boundaries is strictly smaller than its local maximum. □

VII. Feedback Capacity of (1,2)-RLL BEC and Future Research

In this section we present the feedback capacity of a BEC with a $(1,2)$-RLL input constraint, $C^{\text{fb}}_{(1,2)}(\varepsilon)$. This is the first example we see in which both d and k constraints are active. Additionally, we discuss possible avenues for future research on this topic.

A. Feedback Capacity of (1,2)-RLL BEC

A binary sequence satisfies the $(1,2)$-RLL constraint if every '1' is followed by at least one '0', but no more than two consecutive '0's are allowed. Graphical representation of the constraint is provided in Fig. 13. We present a capacity achieving coding scheme and an upper bound based on the Q-graph approach.

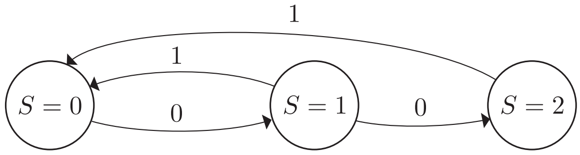

Fig. 13. State diagram describing all sequences that can be generated while satisfying the (1,2)-RLL constraint: every '1' is followed by a '0', and two consecutive '0's are followed by a '1'.

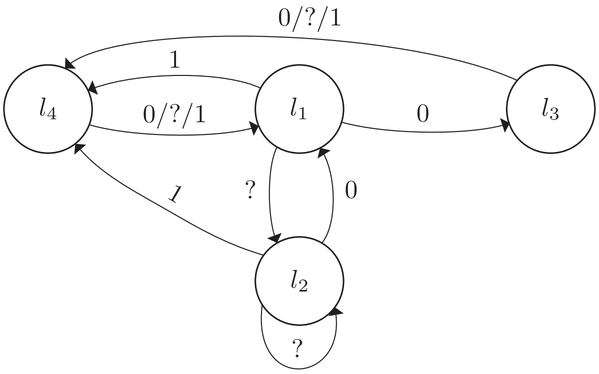

Fig. 14. FSM which defines the coding scheme. The nodes describe the instantaneous labelling that is used by the encoder. Edges correspond to channel outputs. In node 1 $\Pr(X=0)=\overline{\delta}$, in node 2 $\Pr(X=0)=\delta$, in node 3 $\Pr(X=0)=0$ and in node 4 $\Pr(X=0)=1$.

The construction of this coding scheme follows closely that of the scheme presented in Section IV. Fig. 14 contains a finite state machine we use in this case. The scheme is defined by the FSM in Fig. 14 and the following channel input distributions:

- $\Pr(X=0\mid L=l_1)=\overline{\delta}$.
- $\Pr(X=0\mid L=l_2)=\delta$.
- $\Pr(X=0\mid L=l_3)=0$.
- $\Pr(X=0\mid L=l_4)=1$.

The partitions of $[0,1)$, i.e., labeling, are not presented because the amount of different labelings increases with time. The next lemma shows that there exists a coding scheme that is defined by the FSM in Fig. 14 and the aforementioned input distributions. It also states the conditions under which this scheme does not violate the input constraint.

Lemma 9. *For $\frac{1}{2}\le\delta\le\frac{2}{3}$, the coding scheme in Fig. 14 does not violate the $(1,2)$-RLL input constraint and achieves:*

$$R=\frac{H_2(\delta)}{\frac{1}{1-\varepsilon}+\overline{\varepsilon}+\overline{\delta}}. \tag{20}$$

The proof of Lemma 9 is presented in Appendix C. Thus, a lower bound on the feedback capacity is:

$$C^{\text{fb}}_{(1,2)}(\varepsilon)\ge\max_{\frac{1}{3}\le\delta\le\frac{1}{2}}\frac{H_2(\delta)}{\frac{1}{1-\varepsilon}+\overline{\varepsilon}+\delta}.$$

For the upper bound, we use the same Q-graph technique from Section VI in Theorem 3. This time, the coding scheme graph presented in Fig. 14 is chosen as our Q-graph. Calculating $\sup_{p(x|s,q)} I(X;Y\mid Q)$ we get:

$$C^{\text{fb}}_{(1,2)}\le\max_{0\le\delta_1,\delta_2\le 1}\frac{\overline{\varepsilon}^2H_2(\delta_1)+\varepsilon\overline{\varepsilon}H_2(\delta_2)}{1+\overline{\varepsilon}+\overline{\varepsilon}\delta_1+\varepsilon\overline{\varepsilon}\delta_2}. \tag{21}$$

The following lemma shows that the upper and lower bounds coincide:

Lemma 10. *The feedback capacity of the* $(1, 2)$*-RLL input constrained BEC is upper bounded by:*

$$C^{\mathrm{fb}}_{(1,2)} \leq \max_{\frac{1}{3}\leq\delta\leq\frac{1}{2}} \frac{H_2(\delta)}{\frac{1}{1-\varepsilon}+\overline{\varepsilon}+\delta}.$$

This proof also appears in Appendix C. This completes the derivation of the capacity of the $(1, 2)$-RLL input constrained BEC.

B. Future Research

As indicated by the $(1, 2)$-RLL example, the most logical course of future research is to study the feedback capacity of the general (d, k)-RLL input constrained BEC for any natural $d < k$. Our method of tackling the various input constraints discussed in this paper consisted of first running numerical evaluations of the equivalent DP problems, and then trying to draw conclusions as to the capacity achieving coding scheme. However, it is important to notice that the amount of variables we need to numerically evaluate grows linearly with the parameters d and k. Thus, this somewhat naive approach will probably not suffice to find the capacity expression for the general case. The feedback capacity of the general (d, k)-RLL input constrained BEC is still open, in particular for $d = 1, k \geq 3$ and for $2 \leq d < k$.

We have also invested efforts in solving the second famous family of the (d, k)-RLL constraints, the (d,∞)-RLL constraints. As illustrated in the previous section, the non causal capacity is not a tight upper bound on the capacity, so alternative methods to the ones presented in this paper should be applied to tackle these constraints. Based on numerical experiments, we tend to believe that the capacity in this case will be an optimization over more than one parameter.

Another topic to consider for future research is the introduction of memory to the erasures process. It is well known that the unconstrained capacity of the BEC with memory is equal to $1 - \varepsilon$ with and without feedback. In other words, for the BEC neither memory nor feedback increase channel capacity. However, in the constrained case one can think of an example where the introduction of memory to the erasures process decreases the capacity relative to that of the same channel with an i.i.d. erasures process. The question of what types of memory increase or decrease the constrained capacity merits further investigation.

Appendix A
Equality of the Bounds

In this appendix, we prove Lemma 1. It states that the lower and upper bounds, calculated in Sections IV and V, respectively, are equal. In addition, it shows that $(\delta_0, \ldots, \delta_{k-1})$ that maximize $R_\varepsilon\,(\delta_0, \ldots, \delta_{k-1})$ are connected to each other in a series of equations that allow us to compute $(\delta_0, \ldots, \delta_{k-2})$ once the maximizing δ_{k-1} is known.

Denote:

$$D_1 = \left\{(\delta_0, \ldots, \delta_{k-1}) \in \mathbb{R}^k \,|\, 0 \leq \delta_0, \ldots, \delta_{k-1} \leq 1\right\}, \quad (22)$$

$$D_2 = \left\{(\delta_0, \ldots, \delta_{k-1}) \in \mathbb{R}^k \,|\, 0 \leq \delta_0, \ldots, \delta_{k-1} \leq \frac{1}{2}\right\}. \quad (23)$$

Define $\vec{\delta}^* = (\delta^*_0, \ldots, \delta^*_{k-1}) \overset{\mathrm{def}}{=} \arg\max_{D_1} R_\varepsilon\,(\delta_0, \ldots, \delta_{k-1})$. We aim to show that $\vec{\delta}^* \in D_2$. The proof is spread across several lemmas, which show the following:

- In Lemma 11 we prove that $\nabla R_\varepsilon(\vec{\delta}) = 0 \iff \vec{\delta}$ satisfies Eqs. (6). We also show that Eqs. (6) imply that $\delta_0 \geq \ldots \geq \delta_{k-1}$.
- Lemma 12 proves that for any $(\delta_1, \ldots, \delta_{k-1}) \in D_1$ there exists a unique $0 \leq \delta_0(\delta_1, \ldots, \delta_{k-1}) \leq \frac{1}{2}$, which is denoted by δ^*_0, such that $\frac{\partial R_\varepsilon(\delta_0,\ldots,\delta_{k-1})}{\partial\delta_0}\Big|_{\delta_0=\delta^*_0} = 0$. Lemmas 11 and 12 together show that there exists a unique $\vec{\delta} \in D_2$ such that $\nabla R_\varepsilon(\vec{\delta}) = 0$.
- Lemma 13 proves that $R_\varepsilon(\delta_0, \ldots, \delta_{k-1})$ has no maximum on the boundary of D_1, and hence, $\vec{\delta}^* \in D_2$.

To simplify notation, for $k > l$ we define $\prod_{m=k}^{l}(\cdot) \overset{\mathrm{def}}{=} 1$ and $\sum_{i=k}^{l}(\cdot) \overset{\mathrm{def}}{=} 0$.

Lemma 11. *A k-tuple* $\vec{\delta} = (\delta_0, \ldots, \delta_{k-1}) \in D_1$ *satisfies* $\nabla R_\varepsilon(\vec{\delta}) = 0$ *if and only if*

$$\delta_j = \frac{\delta_{j+1}}{\delta_{j+1} + \overline{\delta}_{j+1}\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}}} \quad j = 0, 1, \ldots, k-2,$$

where we define $\overline{\delta}_k = 1$*. In addition* $\delta_0 \geq \ldots \geq \delta_{k-1}$*.*

Proof. First prove that $\nabla R_\varepsilon(\vec{\delta}) = 0$ if and only if the following relation holds:

$$\log\left(\frac{\overline{\delta}_j}{\delta_j}\right) = \log\left(\frac{\overline{\delta}_{j+1}}{\delta_{j+1}}\right) + \overline{\varepsilon}\log\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right),\ j = 0, 1, \ldots, k-2.$$

Denote:

$$N = \sum_{i=0}^{k-1}\left(\overline{\varepsilon}^{i+1} H_2(\delta_i) \prod_{m=0}^{i-1}\delta_m\right),$$

$$D = 1 + \sum_{i=0}^{k-1}\left(\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right). \quad (24)$$

So

$$R_\varepsilon\,(\delta_0, \ldots, \delta_{k-1}) = \frac{N}{D}$$

and,

$$\frac{\partial R_\varepsilon(\delta_0, \ldots, \delta_{k-1})}{\partial\delta_0} = \frac{\frac{\partial N}{\partial\delta_0}D - N\frac{\partial D}{\partial\delta_0}}{D^2}.$$

We write the partial derivative $\frac{\partial R_\varepsilon(\delta_0\ldots\delta_{k-1})}{\partial\delta_j}$ for $j = 0, 1, \ldots, k-1$ using the notations introduced in (24):

$$\frac{\partial R_\varepsilon(\delta_0\ldots\delta_{k-1})}{\partial\delta_j} = \frac{1}{D}\left(\overline{\varepsilon}^{j+1}\prod_{m=0}^{j-1}\delta_m \log\left(\frac{\overline{\delta}_j}{\delta_j}\right) + \sum_{i=j+1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{\substack{m=0\\m\neq j}}^{i-1}\delta_m\right) - \frac{N}{D^2}\sum_{i=j}^{k-1}\overline{\varepsilon}^{i+1}\prod_{\substack{m=0\\m\neq j}}^{i}\delta_m. \quad (25)$$

We will prove the lemma using an inductive argument starting from δ_{k-1} and working our way back to δ_0.

Base case: by simplifying the equation $\frac{\partial R_\varepsilon}{\delta_{k-1}} = 0$ we immediately get:

$$N = D \log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right). \tag{26}$$

Note that we arrive at (26) by dividing both sides of the equation by $\prod_{m=0}^{k-2} \delta_m$. We know that this is allowed since for any $j = 0, \ldots, k-1$ we have that $\lim_{\delta_j \to 0^+} \frac{\partial R_\varepsilon(\delta_0 \ldots \delta_{k-1})}{\partial \delta_j} = \infty$. This means that if $\nabla R_\varepsilon(\delta_0 \ldots \delta_{k-1}) = 0$ then $\delta_j \neq 0$ for all $j = 0, \ldots, k-1$.

Next we write the equation $\frac{\partial R_\varepsilon}{\partial \delta_{k-2}} = 0$ and substitute N using (26):

$$\begin{aligned}
0 &= \left(\overline{\varepsilon}^{k-1} \prod_{m=0}^{k-3} \delta_m \log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) + \overline{\varepsilon}^{k} H_2(\delta_{k-1}) \prod_{m=0}^{k-3} \delta_m\right) \frac{1}{D} \\
&\quad - \frac{1}{D} \log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right) \left[\overline{\varepsilon}^{k-1} \prod_{m=0}^{k-3} \delta_m + \overline{\varepsilon}^{k} \prod_{m=0}^{k-3} \delta_m \delta_{k-1}\right] \\
&= \log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) - \overline{\varepsilon}\delta_{k-1} \log(\delta_{k-1}) - \overline{\varepsilon}\overline{\delta}_{k-1} \log(\overline{\delta}_{k-1}) \\
&\quad - \log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right) - \overline{\varepsilon}\delta_{k-1} \log(\overline{\delta}_{k-1}) + \overline{\varepsilon}\delta_{k-1} \log(\delta_{k-1}).
\end{aligned}$$

So

$$\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) = \log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right) + \overline{\varepsilon} \log\left(\frac{\overline{\delta}_{k-1}}{1}\right) \tag{27}$$

and the base case is proven.

Inductive step: we assume that the claim holds for $\delta_{k-2}, \delta_{k-3}, \ldots, \delta_{j+1}$ and we will now prove it for δ_j. Substituting (27) into (26) we get:

$$N = D\left(\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) - \overline{\varepsilon} \log(\overline{\delta}_{k-1})\right). \tag{28}$$

We start by writing the equation

$$\begin{aligned}
0 &= \frac{\partial R_\varepsilon(\delta_0, \ldots, \delta_{k-1})}{\partial \delta_j} \\
&= \frac{1}{D^2} \left(\left(\overline{\varepsilon}^{j+1} \prod_{m=0}^{j-1} \delta_m \log\left(\frac{\overline{\delta}_j}{\delta_j}\right)\right.\right. \\
&\quad \left. + \sum_{i=j+1}^{k-1} \overline{\varepsilon}^{i+1} H_2(\delta_i) \prod_{\substack{m=0 \\ m \neq j}}^{i-1} \delta_m \right) D \\
&\quad \left. - N \sum_{i=j}^{k-1} \overline{\varepsilon}^{i+1} \prod_{\substack{m=0 \\ m \neq j}}^{i} \delta_m \right).
\end{aligned}$$

We can divide by $\varepsilon^{j+1} \prod_{m=0}^{j-1} \delta_m$ and use (28) to get:

$$\begin{aligned}
0 &= \left(\log\left(\frac{\overline{\delta}_j}{\delta_j}\right) + \sum_{i=j+1}^{k-1} \overline{\varepsilon}^{i-j} H_2(\delta_i) \prod_{m=j+1}^{i-1} \delta_m\right) D \\
&\quad - D\left(\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) - \overline{\varepsilon} \log(\overline{\delta}_{k-1})\right) \sum_{i=j}^{k-1} \overline{\varepsilon}^{i-j} \prod_{m=j+1}^{i} \delta_m.
\end{aligned}$$

Next we use the definition of the binary entropy function to replace $H_2(\delta_{k-2}), H_2(\delta_{k-1})$ with an explicit expression:

$$\begin{aligned}
0 &= \log\left(\frac{\overline{\delta}_j}{\delta_j}\right) + \sum_{i=j+1}^{k-3} \overline{\varepsilon}^{i-j} H_2(\delta_i) \prod_{m=j+1}^{i-1} \delta_m \\
&\quad - \overline{\varepsilon}^{k-2-j} \prod_{m=j+1}^{k-3} \delta_m \delta_{k-2} \log(\delta_{k-2}) \\
&\quad - \overline{\varepsilon}^{k-2-j} \prod_{m=j+1}^{k-3} \delta_m \overline{\delta}_{k-2} \log(\overline{\delta}_{k-2}) \\
&\quad - \overline{\varepsilon}^{k-1-j} \prod_{m=j+1}^{k-2} \delta_m \delta_{k-1} \log(\delta_{k-1}) \\
&\quad - \overline{\varepsilon}^{k-1-j} \prod_{m=j+1}^{k-2} \delta_m \overline{\delta}_{k-1} \log(\overline{\delta}_{k-1}) \\
&\quad - \log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) \sum_{i=j}^{k-3} \overline{\varepsilon}^{i-j} \prod_{m=j+1}^{i} \delta_m \\
&\quad - \overline{\varepsilon}^{k-2-j} \prod_{m=j+1}^{k-2} \delta_m \log(\overline{\delta}_{k-2}) \\
&\quad + \overline{\varepsilon}^{k-2-j} \prod_{m=j+1}^{k-3} \delta_m \delta_{k-2} \log(\delta_{k-2}) \\
&\quad - \log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) \overline{\varepsilon}^{k-1-j} \prod_{m=j+1}^{k-1} \delta_m \\
&\quad + \log(\overline{\delta}_{k-1}) \sum_{i=j}^{k-3} \overline{\varepsilon}^{i-j+1} \prod_{m=j+1}^{i} \delta_m \\
&\quad + \log(\overline{\delta}_{k-1}) \overline{\varepsilon}^{k-1-j} \prod_{m=j+1}^{k-2} \delta_m \\
&\quad + \log(\overline{\delta}_{k-1}) \overline{\varepsilon}^{k-j} \prod_{m=j+1}^{k-1} \delta_m.
\end{aligned}$$

Recall that $\overline{\delta} = 1 - \delta$, so we can simplify this expression:

$$\begin{aligned}
0 &= \log\left(\frac{\overline{\delta}_j}{\delta_j}\right) + \sum_{i=j+1}^{k-3} \overline{\varepsilon}^{i-j} H_2(\delta_i) \prod_{m=j+1}^{i-1} \delta_m \\
&\quad - \overline{\varepsilon}^{k-2-j} \prod_{m=j+1}^{k-3} \delta_m \log(\overline{\delta}_{k-2})
\end{aligned}$$

$$
\begin{aligned}
&\overbrace{+\log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right)\overline{\varepsilon}^{k-1-j}\prod_{m=j+1}^{k-1}\delta_m}^{(*)}\\
&-\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right)\sum_{i=j}^{k-3}\overline{\varepsilon}^{i-j}\prod_{m=j+1}^{i}\delta_m\\
&\overbrace{-\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right)\overline{\varepsilon}^{k-1-j}\prod_{m=j+1}^{k-1}\delta_m}^{(*)}\\
&+\log(\overline{\delta}_{k-1})\sum_{i=j}^{k-3}\overline{\varepsilon}^{i-j+1}\prod_{m=j+1}^{i}\delta_m\\
&\overbrace{+\log(\delta_{k-1})\overline{\varepsilon}^{k-j}\prod_{m=j+1}^{k-1}\delta_m}^{(*)}\,. \qquad (29)
\end{aligned}
$$

The three expression marked with $(*)$ cancel each other as a result of (27). Now we will use the induction assumption again by substituting

$$
\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right)=\log\left(\frac{\overline{\delta}_{k-3}}{\delta_{k-3}}\right)-\overline{\varepsilon}\log\left(\frac{\overline{\delta}_{k-2}}{\overline{\delta}_{k-1}}\right),
$$

so

$$
\begin{aligned}
0=\;&\log\left(\frac{\overline{\delta}_j}{\delta_j}\right)+\sum_{i=j+1}^{k-4}\overline{\varepsilon}^{i-j}H_2(\delta_i)\prod_{m=j+1}^{i-1}\delta_m\\
&\overbrace{-\overline{\varepsilon}^{k-3-j}\prod_{m=j+1}^{k-4}\delta_m\delta_{k-3}\log(\delta_{k-3})}^{(*)}\\
&-\overline{\varepsilon}^{k-3-j}\prod_{m=j+1}^{k-4}\delta_m\overline{\delta}_{k-3}\log(\overline{\delta}_{k-3})\\
&\overbrace{-\overline{\varepsilon}^{k-2-j}\prod_{m=j+1}^{k-3}\delta_m\log(\overline{\delta}_{k-2})}^{(*)}\\
&-\log\left(\frac{\overline{\delta}_{k-3}}{\delta_{k-3}}\right)+\overline{\varepsilon}\log(\overline{\delta}_{k-2})-\overbrace{\overline{\varepsilon}\log(\overline{\delta}_{k-1})}^{(*)}\\
&-\overline{\varepsilon}\delta_{j+1}\log\left(\frac{\overline{\delta}_{k-3}}{\delta_{k-3}}\right)+\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{k-2})\\
&-\overbrace{\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{k-1})}^{(*)}-\log\left(\frac{\overline{\delta}_{k-3}}{\delta_{k-3}}\right)\\
&\sum_{i=j+2}^{k-4}\overline{\varepsilon}^{i-j}\prod_{m=j+1}^{i}\delta_m+\log(\overline{\delta}_{k-2})\sum_{i=j+2}^{k-4}\overline{\varepsilon}^{i-j+1}\prod_{m=j+1}^{i}\delta_m\\
&\overbrace{-\log(\overline{\delta}_{k-1})\sum_{i=j+2}^{k-4}\overline{\varepsilon}^{i-j+1}\prod_{m=j+1}^{i}\delta_m}^{(*)}
\end{aligned}
$$

$$
\begin{aligned}
&-\log(\overline{\delta}_{k-3})\overline{\varepsilon}^{k-3-j}\prod_{m=j+1}^{k-3}\delta_m\\
&\overbrace{+\log(\delta_{k-3})\overline{\varepsilon}^{k-3-j}\prod_{m=j+1}^{k-3}\delta_m}^{(*)}\\
&\overbrace{+\log(\overline{\delta}_{k-2})\overline{\varepsilon}^{k-2-j}\prod_{m=j+1}^{k-3}\delta_m}^{(*)}\\
&\overbrace{-\log(\overline{\delta}_{k-1})\overline{\varepsilon}^{k-2-j}\prod_{m=j+1}^{k-3}\delta_m}^{(*)}\\
&\overbrace{+\log(\overline{\delta}_{k-1})\sum_{i=j}^{k-3}\overline{\varepsilon}^{i-j+1}\prod_{m=j+1}^{i}\delta_m}^{(*)}\,.
\end{aligned}
$$

All expressions marked with $(*)$ cancel each other out. Again, using $\overline{\delta}=1-\delta$ we can simplify and arrive at:

$$
\begin{aligned}
0=\;&\log\left(\frac{\overline{\delta}_j}{\delta_j}\right)+\sum_{i=j+1}^{k-4}\overline{\varepsilon}^{i-j}H_2(\delta_i)\prod_{m=j+1}^{i-1}\delta_m\\
&-\overline{\varepsilon}^{k-3-j}\prod_{m=j+1}^{k-4}\delta_m\log(\overline{\delta}_{k-3})\\
&-\log\left(\frac{\overline{\delta}_{k-3}}{\delta_{k-3}}\right)\sum_{i=j}^{k-4}\overline{\varepsilon}^{i-j}\prod_{m=j+1}^{i}\delta_m\\
&+\log(\overline{\delta}_{k-2})\sum_{i=j}^{k-4}\overline{\varepsilon}^{i-j+1}\prod_{m=j+1}^{i}\delta_m. \qquad (30)
\end{aligned}
$$

When we compare (30) to (29) we see a pattern emerging. Continuing to perform these substitutions we reach:

$$
\begin{aligned}
0=\;&\log\left(\frac{\overline{\delta}_j}{\delta_j}\right)-\overline{\varepsilon}\delta_{j+1}\log(\delta_{j+1})\\
&-\overline{\varepsilon}\overline{\delta}_{j+1}\log(\overline{\delta}_{j+1})-\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{j+2})\\
&-\log\left(\frac{\overline{\delta}_{j+2}}{\delta_{j+2}}\right)(1+\overline{\varepsilon}\delta_{j+1})\\
&+\log(\overline{\delta}_{j+2})(\overline{\varepsilon}+\overline{\varepsilon}^2\delta_{j+1}).
\end{aligned}
$$

Performing the final substitution and simplifying further we get:

$$
\begin{aligned}
0=\;&\log\left(\frac{\overline{\delta}_j}{\delta_j}\right)-\overline{\varepsilon}\delta_{j+1}\log(\delta_{j+1})-\overline{\varepsilon}\overline{\delta}_{j+1}\log(\overline{\delta}_{j+1})\\
&-\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{j+2})-\log\left(\frac{\overline{\delta}_{j+1}}{\delta_{j+1}}\right)+\overline{\varepsilon}\log(\overline{\delta}_{j+2})\\
&-\overline{\varepsilon}\log(\overline{\delta}_{j+3})-\overline{\varepsilon}\delta_{j+1}\log(\overline{\delta}_{j+1})+\overline{\varepsilon}\delta_{j+1}\log(\delta_{j+1})\\
&+\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{j+2})-\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{j+3})+\overline{\varepsilon}\log(\overline{\delta}_{j+3})\\
&+\overline{\varepsilon}^2\delta_{j+1}\log(\overline{\delta}_{j+3}),
\end{aligned}
$$

and, finally, we arrive at:

$$\log\left(\frac{\overline{\delta}_j}{\delta_j}\right) = \log\left(\frac{\overline{\delta}_{j+1}}{\delta_{j+1}}\right) + \overline{\varepsilon}\log\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right). \tag{31}$$

Now we will use induction again to prove that

$$\delta_j = \frac{\delta_{j+1}}{\delta_{j+1} + \overline{\delta}_{j+1}\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}}} \quad j = 0, 1, \ldots, k-2$$

and that $\delta_0 \geq \delta_1 \geq \ldots \geq \delta_{k-1}$.

Base case: we will start by showing that $\delta_{k-2} \geq \delta_{k-1}$. In (27) we have that

$$\log\left(\frac{\overline{\delta}_{k-2}}{\delta_{k-2}}\right) = \log\left(\frac{\overline{\delta}_{k-1}}{\delta_{k-1}}\right) + \overline{\varepsilon}\log\left(\overline{\delta}_{k-1}\right).$$

By rearranging this equation we get:

$$\frac{\overline{\delta}_{k-2}}{\delta_{k-2}} = \frac{\overline{\delta}_{k-1}^{1+\overline{\varepsilon}}}{\delta_{k-1}},$$

which means that:

$$\begin{aligned}\delta_{k-2} =& \frac{1}{1 + \frac{\overline{\delta}_{k-1}^{1+\overline{\varepsilon}}}{\delta_{k-1}}}\\ =& \frac{\delta_{k-1}}{\delta_{k-1} + \overline{\delta}_{k-1}^{1+\overline{\varepsilon}}}.\end{aligned}$$

Note that assuming $\delta_{k-1} > 0$:

$$\delta_{k-1} + \overline{\delta}_{k-1}^{1+\overline{\varepsilon}} < 1 \iff \overline{\delta}_{k-1}^{\overline{\varepsilon}} < 1$$

and the right hand side of this equivalence surely holds (under said assumption). We have proven the base case.

Inductive step: we assume that the claim holds for $\delta_{k-2}, \delta_{k-3}, \ldots, \delta_{j+1}$ and we will now prove it for δ_j. In (31) we have:

$$\log\left(\frac{\overline{\delta}_j}{\delta_j}\right) = \log\left(\frac{\overline{\delta}_{j+1}}{\delta_{j+1}}\right) + \overline{\varepsilon}\log\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right).$$

Following the same steps as in the base case we arrive at:

$$\delta_j = \frac{\delta_{j+1}}{\delta_{j+1} + \overline{\delta}_{j+1}\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}}}.$$

Now,

$$\delta_{j+1} + \overline{\delta}_{j+1}\left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}} < 1 \iff \left(\frac{\overline{\delta}_{j+1}}{\overline{\delta}_{j+2}}\right)^{\overline{\varepsilon}} < 1,$$

and we know that the right hand side of the equivalence holds thanks to the induction assumption (note that $\delta_{j+1} > \delta_{j+2} \implies \overline{\delta}_{j+1} < \overline{\delta}_{j+2}$). □

This lemma shows that for any $(\delta_1, \ldots, \delta_{k-1})$ satisfying $0 \leq \delta_1, \ldots, \delta_{k-1} \leq 1$ there exists $0 \leq \bar{\delta}_0 \leq \frac{1}{2}$ for which $\frac{\partial R_\varepsilon(\delta_0,\ldots,\delta_{k-1})}{\partial\delta_0}\Big|_{\delta_0=\bar{\delta}_0} = 0$ and that this $\bar{\delta}_0$ is unique.

Lemma 12. *This lemma has two parts:*

1) *For any* $(\delta_1, \ldots, \delta_{k-1})$ *satisfying* $0 \leq \delta_1, \ldots, \delta_{k-1} \leq 1$ *there exists* $0 \leq \delta_0(\delta_1, \ldots, \delta_{k-1}) < \frac{1}{2}$*, which we denote* $\bar{\delta}_0$*, such that:*

$$\frac{\partial R_\varepsilon\left(\delta_0, \ldots, \delta_{k-1}\right)}{\partial\delta_0}\Bigg|_{\delta_0=\bar{\delta}_0} = 0.$$

2) *The partial derivative* $\frac{\partial R_\varepsilon(\delta_0,\ldots,\delta_{k-1})}{\partial\delta_0}$ *is monotonic non-increasing in* δ_0.

Proof. We calculate $\frac{\partial R_\varepsilon(\delta_0,\ldots,\delta_{k-1})}{\partial\delta_0}$ and show that:

$$\lim_{\delta_0\to 0^+}\frac{\partial R_\varepsilon\left(\delta_0, \ldots, \delta_{k-1}\right)}{\partial\delta_0} > 0 \tag{32}$$

and

$$\frac{\partial R_\varepsilon\left(\delta_0, \ldots, \delta_{k-1}\right)}{\partial\delta_0}\Bigg|_{\delta_0=\frac{1}{2}} < 0. \tag{33}$$

Since the partial derivative is a continuous function of δ_0 we can use the intermediate value theorem to prove the first part of the lemma. Recall that:

$$\frac{\partial R_\varepsilon(\delta_0, \ldots, \delta_{k-1})}{\partial\delta_0} = \frac{\frac{\partial N}{\partial\delta_0}D - N\frac{\partial D}{\partial\delta_0}}{D^2}.$$

First note that $D^2 > 0$. This means that we only need to determine the sign of $\frac{\partial N}{\partial\delta_0}D - N\frac{\partial D}{\partial\delta_0}$ as $\delta_0 \to 0^+$ and for $\delta_0 = \frac{1}{2}$ to prove that (32) and (33) hold. Since the expression $\frac{\partial R_\varepsilon}{\partial\delta_0}$ is a long one, we will divide it into two parts:

$$\begin{aligned}\frac{\partial N}{\partial\delta_0}D =& \left(\overline{\varepsilon}\log\left(\frac{\overline{\delta}_0}{\delta_0}\right) + \sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\right)\\ &\left(1 + \sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right),\end{aligned} \tag{34}$$

$$N\frac{\partial D}{\partial\delta_0} = \left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=0}^{i-1}\delta_m\right)\left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=1}^{i}\delta_m\right). \tag{35}$$

Simplifying $\frac{\partial N}{\partial\delta_0}D - N\frac{\partial D}{\partial\delta_0}$ we get:

$$\begin{aligned}\frac{\partial N}{\partial\delta_0}D - N\frac{\partial D}{\partial\delta_0} =& \overline{\varepsilon}\log\left(\frac{\overline{\delta}_0}{\delta_0}\right)\\ &+ \log\left(\frac{\overline{\delta}_0}{\delta_0}\right)\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=0}^{i}\delta_m + \sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\\ &+ \left(\sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\right)\left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right)\\ &- \overline{\varepsilon}H_2(\delta_0)\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=1}^{i}\delta_m\\ &- \left(\sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\right)\left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_m\right)\\ =& \log(\overline{\delta}_0)\left[\overline{\varepsilon} + \sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=0}^{i}\delta_m + \overline{\delta}_0\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=1}^{i}\delta_m\right]\\ &+ \log(\delta_0)\left[-\overline{\varepsilon} - \sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=0}^{i}\delta_m\right.\\ &\left.+\delta_0\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=1}^{i}\delta_m\right] + \sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\\ =& \log(\overline{\delta}_0)\left[\overline{\varepsilon} + \sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=1}^{i}\delta_m\right] - \overline{\varepsilon}\log(\delta_0)\end{aligned}$$

$$+\sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m. \qquad (36)$$

It is clear from (36) that $\lim_{\delta_0\to 0^+}\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}=\infty$.

Next we evaluate $\left(\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}\right)\Big|_{\delta_0=\frac{1}{2}}$:

$$\begin{aligned}&\left(\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}\right)\Big|_{\delta_0=\frac{1}{2}}=\\&-\left[\overline{\varepsilon}+\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=1}^{i}\delta_m\right]+\overline{\varepsilon}+\sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\\&=\overline{\varepsilon}^2\left[(H_2(\delta_1)-1)+\overline{\varepsilon}\delta_1\left(H_2(\delta_2)-1\right)+\dots\right.\\&\left.+\overline{\varepsilon}^{k-2}\prod_{m=1}^{k-2}\delta_m\left(H_2(\delta_{k-1})-1\right)-\overline{\varepsilon}^{k-1}\prod_{m=1}^{k-1}\delta_m\right]. \qquad (37)\end{aligned}$$

Note that all the summands are non-positive. It follows that:

$$\begin{aligned}&(H_2(\delta_1)-1)+\overline{\varepsilon}\delta_1\left(H_2(\delta_2)-1\right)+\dots\\&+\overline{\varepsilon}^{k-2}\prod_{m=1}^{k-2}\delta_m\left(H_2(\delta_{k-1})-1\right)=0\end{aligned}$$

if and only if we set $\delta_1=\ldots=\delta_{k-1}=\frac{1}{2}$. However setting $\delta_1=\ldots=\delta_{k-1}=\frac{1}{2}$ we get:
$-\overline{\varepsilon}^{k-1}\prod_{m=1}^{k-1}\delta_m<0$. Thus $\left(\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}\right)\Big|_{\delta_0=\frac{1}{2}}<0$. We now use the intermediate value theorem to prove the first part of the lemma.

In the second part we want to show that the partial derivative $\frac{\partial R_\varepsilon(\delta_0,\ldots,\delta_{k-1})}{\partial\delta_0}$ is monotonic non-increasing in δ_0. It is clear that D^2 is monotonic increasing in δ_0 so we must prove that $\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}$ is non-increasing in δ_0 to complete the proof. In order to achieve this goal we derive $\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}$ again with respect to δ_0:

$$\frac{\partial\left(\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}\right)}{\partial\delta_0}=\frac{-\left[\overline{\varepsilon}+\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+2}\prod_{m=1}^{i}\delta_m\right]}{\overline{\delta}_0}-\frac{\overline{\varepsilon}}{\delta_0}$$

This expression is clearly non-positive and that proves the lemma. □

We have shown that there is a unique $\vec{\delta}\in D_2$ that satisfies $\nabla R_\varepsilon(\vec{\delta})=\vec{0}$. Now, all that remains is to prove that the suspicious point we worked so hard to find is, in fact, a local maximum of $R_\varepsilon(\delta_0,\ldots,\delta_{k-1})$ in the domain D_1. We already know that it is the only critical point in the interior of the domain, so we can safely say that the function gets its maximum value in that point or somewhere on the boundary. The final lemma will show that the function does not get its maximal value on the edge of the domain.

Lemma 13. *The maximum of $R_\varepsilon(\delta_0,\ldots,\delta_{k-1})$ does not occur on the boundary of the domain D_1.*

Proof. To prove this we will use the KKT conditions. First we will write the maximization problem in its standard form. Define $\vec{\delta}=(\delta_0,\ldots,\delta_{k-1})$ and the following constraint functions:

$$\begin{aligned}g_0(\vec{\delta})&=-\delta_0\ ,\ \tilde{g}_0(\vec{\delta})=\delta_0-1\\&\ \vdots\qquad\qquad\vdots\\g_{k-1}(\vec{\delta})&=-\delta_{k-1}\ ,\ \tilde{g}_{k-1}(\vec{\delta})=\delta_{k-1}-1.\end{aligned}$$

We want to maximize

$$R_\varepsilon(\delta_0,\ldots,\delta_{k-1})$$

subject to

$$g_i(\vec{\delta})\le 0\quad,\quad\tilde{g}_i(\vec{\delta})\le 0\qquad i=0,\ldots,k-1.$$

The KKT conditions tell us that if a point $\vec{\delta}^*$ is a local maximum then there exist constants μ_i and $\tilde{\mu}_i$ ($i=0,\ldots,k-1$) such that:

$$\nabla R_\varepsilon(\vec{\delta}^*)=\sum_{i=0}^{k-1}\mu_i\nabla g_i(\vec{\delta}^*)+\sum_{i=0}^{k-1}\tilde{\mu}_i\nabla\tilde{g}_i(\vec{\delta}^*) \qquad (38)$$

and

$$\mu_i\ge 0\quad,\quad\tilde{\mu}_i\ge 0\qquad i=0,\ldots,k-1.$$

Let us assume, by contradiction, that R_ε has a local maximum on the boundary of the domain D_1, and, specifically, that the local maximum is obtained for $\delta_0=0$. Since we assume that $g_0(\vec{\delta})=0$ we know that $\tilde{g}_0(\vec{\delta})=-1$ and so there is no need to address the inequality condition $\tilde{g}_0(\vec{\delta})\le 0$. Eq. (38) above gives us $k-1$ equalities. In this case the equality that we get from differentiating with regard to δ_0 is:

$$\frac{\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}}{D^2}=-\mu_0, \qquad (39)$$

where

$$\begin{aligned}\frac{\partial N}{\partial\delta_0}D&=\left(\log\left(\frac{\overline{\delta}_0}{\delta_0}\right)+\sum_{i=1}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\right)\\&\quad\left(1+\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_i\right)\\N\frac{\partial D}{\partial\delta_0}&=\left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}H_2(\delta_i)\prod_{m=1}^{i-1}\delta_m\right)\left(\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=1}^{i}\delta_i\right)\\D^2&=\left(1+\sum_{i=0}^{k-1}\overline{\varepsilon}^{i+1}\prod_{m=0}^{i}\delta_i\right)^2.\end{aligned}$$

We have already shown in a previous lemma that the left hand side of Eq. (39) tends to $+\infty$ as $\delta_0\to 0^+$ and so we get a negative μ_0 in violation of the KKT conditions. If we assume that a local maximum is obtained for $\delta_0=1$ we will arrive at a similar equation:

$$\frac{\frac{\partial N}{\partial\delta_0}D-N\frac{\partial D}{\partial\delta_0}}{D^2}=\tilde{\mu}_0, \qquad (40)$$

and it is easy to see that the left hand side of Eq. (40) tends to $-\infty$ as $\delta_0\to 1^-$ so, again, we get a negative $\tilde{\mu}_0$. We can conclude that there is no local maximum of R_ε on the boundary of D_1 where $\delta_0=0$ or $\delta_0=1$. In a similar way we can show that there is, in fact, no local maximum on any part of the boundary of D_1. □

We have proven that $R_\varepsilon(\delta_0, \ldots, \delta_{k-1})$ has one local maximum in the domain D_1 and that it satisfies $\frac{1}{2} > \delta_0 \geq \delta_1 \geq \ldots \geq \delta_{k-1}$. This proves that we can substitute D_1 for D_2 as the maximization domain and the maximal value will not change as a result.

Appendix B
Lemma for the Converse

Prior to the proof we present some standard definitions of second order types. A second order type of a sequence $x^n \in \mathcal{X}^n$ is a probability distribution $\hat{P}^{(2)}_{x^n} \in \mathcal{P}_{n-1}(\mathcal{X}^2)$ defined as:

$$\hat{P}^{(2)}_{x^n}(a,b) = \frac{N\left((a,b) \mid x^n\right)}{n-1},$$

for all $(a,b) \in (\mathcal{X}^2)$. Denote by $\mathcal{P}^{(2)}_n(\mathcal{X}, c)$ the set of all possible second order types of sequences $x^n \in \mathcal{X}^n$ with $x_1 = c$. The second order type of a sequence x^n can be viewed as the joint empirical distribution of $x_1, x_2, \ldots, x_{n-1}$ and $x_2, x_3, \ldots, x_n$. For dummy random variables X, Y representing such a second order type (i.e., $P_{X,Y} \in \mathcal{P}^{(2)}_n(\mathcal{X}, c)$), we define a second order type class:

$$T^{n,(2)}(P_{X,Y}, c) = \{x^n \in \mathcal{X}^n : \hat{P}^{(2)}_{x^n} = P_{X,Y}, x_1 = c\}.$$

We also define a second order ε-typical set with respect to a joint distribution $P_{X,Y}$ and c:

$$\begin{aligned} T^{n,(2)}_{\varepsilon}(P_{X,Y}, c) = \Big\{ & x^n \in \mathcal{X}^n : x_1 = c, \\ & \forall (a,b) \in \mathcal{X}^2 \left| \hat{P}^{(2)}_{x^n}(a,b) - P_{X,Y}(a,b) \right| \leq \varepsilon \\ & \text{and } P_{X,Y}(a,b) = 0 \Rightarrow \hat{P}^{(2)}_{x^n}(a,b) = 0 \Big\}. \end{aligned}$$

A known result from [23] is that for large enough n there exists $\tau(\varepsilon) > 0$ such that:

$$2^{n(H(Y|X)-\tau(\varepsilon))} \leq \left| T^{n,(2)}_{\varepsilon}(P_{X,Y}, c) \right| \leq 2^{n(H(Y|X)+\tau(\varepsilon))},$$

where $\lim_{\varepsilon \to 0} \tau(\varepsilon) = 0$.

Proof of Lemma 6 We aim to show that there exists a single letter distribution such that the typical set induced by this distribution contains the typical set induced by the given n-tuple distribution. Since for a given distribution the size of its typical set is closely related to its entropy, that will suffice to prove the lemma.

We emphasize that n is fixed throughout the proof, so defined quantities are implicit functions of n. Consider an n-tuple constrained distribution $\tilde{P}_{Y^n}(y^n) = \mathbb{1}_{\{y_1=1\}} \prod_{i=2}^{n} \tilde{P}^{(i)}_{Y_i|Y_{i-1}}(y_i \mid y_{i-1})$, as stated in the lemma. Set $\varepsilon = \frac{1}{(n-1)|\mathcal{X}|^n}$. Let $x^{nk} \in T^k_{\varepsilon}(\tilde{P}_{Y^n})$, meaning that the sequence x^{nk} contains k "letters" of the n-fold alphabet $\mathcal{X}^n$ and it is first order ε-typical with respect to the distribution $\tilde{P}_{Y^n}$. Since $x^{nk} \in T^k_{\varepsilon}(\tilde{P}_{Y^n})$ we have that $\forall x^n \in \mathcal{X}^n \ |\hat{P}_{x^{nk}}(x^n) - \tilde{P}_{Y^n}(x^n)| \leq \varepsilon$. Equivalently:

$$\left| N(x^n \mid x^{nk}) - k\tilde{P}_{Y^n}(x^n) \right| \leq k\varepsilon. \tag{41}$$

Define a single letter joint distribution:

$$\tilde{P}_{X,Y}(a,b) = \sum_{x^n \in \mathcal{X}^n} \tilde{P}_{Y^n}(x^n) \frac{N\left((a,b) \mid x^n\right)}{n-1}. \tag{42}$$

We want to show that there exists $\delta > 0$ such that $x^{nk} \in T^{nk,(2)}_{\delta}(\tilde{P}_{X,Y}, 1)$. In order to do that we need to calculate the empirical distribution of pairs of letters in x^{nk}. Each n-tuple $x^{n,i}$ contains $n-1$ pairs. The sequence x^{nk} is made up of k n-tuples, so there are an additional $k-1$ pairs that are not contained in a single n-tuple. In total there are $k(n-1)+k-1 = nk-1$ pairs of letters. For $(a,b) \in \mathcal{X}^2$ denote by $\eta(a,b)$ the number of times the pair (a,b) appears in x^{nk} and is not contained in a single n-tuple. Clearly $0 \leq \eta(a,b) \leq k-1$. Now:

$$\begin{aligned} &\left| \hat{P}^{(2)}_{x^{nk}}(a,b) - \tilde{P}_{X,Y}(a,b) \right| = \left| \frac{N\left((a,b) \mid x^{nk}\right)}{nk-1} - \tilde{P}_{X,Y}(a,b) \right| \\ &= \left| \frac{1}{nk-1} \left[\sum_{x^n \in \mathcal{X}^n} N(x^n \mid x^{nk}) N\left((a,b) \mid x^n\right) + \eta(a,b) \right] \right. \\ &\quad \left. - \tilde{P}_{X,Y}(a,b) \right| \\ &= \left| \frac{1}{nk-1} \left[\sum_{x^n \in \mathcal{X}^n} \left(N(x^n \mid x^{nk}) - k\tilde{P}_{Y^n}(x^n) \right) \right. \right. \\ &\quad N\left((a,b) \mid x^n\right) \\ &\quad \left. \left. + \sum_{x^n \in \mathcal{X}^n} k\tilde{P}_{Y^n}(x^n) N\left((a.b) \mid x^n\right) + \eta(a,b) \right] - \tilde{P}_{X,Y}(a,b) \right| \\ &\overset{(a)}{=} \left| \frac{1}{nk-1} \sum_{x^n \in \mathcal{X}^n} \left(N(x^n \mid x^{nk}) - k\tilde{P}_{Y^n}(x^n) \right) N\left((a,b) \mid x^n\right) \right. \\ &\quad \left. + \frac{k(n-1)\tilde{P}_{X,Y}(a,b)}{nk-1} + \frac{\eta(a,b)}{nk-1} - \frac{(nk-1)\tilde{P}_{X,Y}(a,b)}{nk-1} \right| \\ &\overset{(b)}{\leq} \left| \frac{|\mathcal{X}|^n k\varepsilon N\left((a,b) \mid x^n\right)}{nk-1} \right| + \left| \frac{(1-k)\tilde{P}_{X,Y}(a,b)}{nk-1} \right| \\ &\quad + \left| \frac{k-1}{nk-1} \right| \\ &\overset{(c)}{\leq} \left| \frac{k}{nk-1} \right| + \left| \frac{k-1}{nk-1} \right| + \left| \frac{k-1}{nk-1} \right|, \end{aligned} \tag{43}$$

where

(a) Follows from Eq. (42): $\sum_{x^n \in \mathcal{X}^n} \tilde{P}_{Y^n}(x^n) N\left((a.b) \mid x^n\right) = (n-1)\tilde{P}_{X,Y}(a,b)$.
(b) Follows from the triangle inequality, Eq. (41) and $\eta(a,b) \leq k-1$.
(c) Follows from the fact that $N\left((a,b) \mid x^n\right) \leq n-1$, $\tilde{P}_{X,Y}(a,b) \leq 1$ and the definition of ε.

From Eq. (43), for every $\xi > 0$ there exist $k \in \mathbb{N}$ such that for any $(a,b) \in \mathcal{X}^2$

$$\left| \hat{P}^{(2)}_{x^{nk}}(a,b) - P_{X,Y}(a,b) \right| \leq \frac{3}{n} + \xi.$$

Denote $\delta = \frac{3}{n} + \xi$. We have shown that $x^{nk} \in T^k_{\varepsilon}(\tilde{P}_{Y_n}) \Longrightarrow x^{nk} \in T^{nk,(2)}_{\delta}(\tilde{P}_{X,Y}, 1)$. Thus, $\left| T^k_{\varepsilon}(\tilde{P}_{Y_n}) \right| \leq \left| T^{nk,(2)}_{\delta}(\tilde{P}_{X,Y}, 1) \right|$.

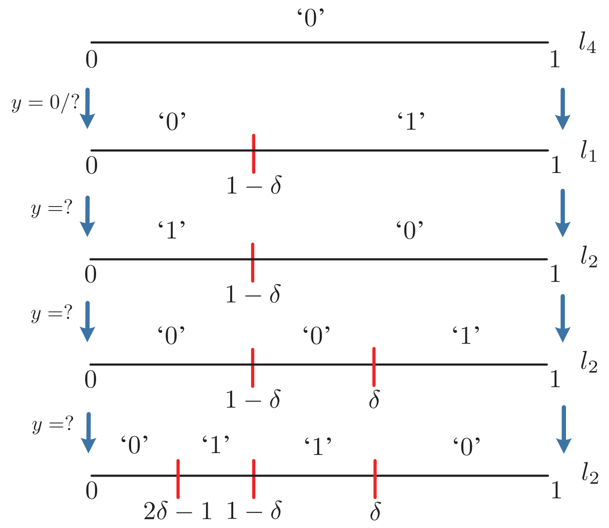

Fig. 15. Induction base case: the initial node is 4 and all channel outputs are erasures. Each partition assigns the required amount of the unit interval to '0' and to '1'. The $(1, 2)$-RLL input constraint is satisfied by these labelings: every '1' is followed by a '0' and no more than two consecutive '0's are allowed.

For any pair $\tau_1(\varepsilon) > 0$, $\tau_2(\delta) > 0$ with $\lim_{\varepsilon\to 0}\tau_1(\varepsilon) = 0$, $\lim_{\delta\to 0}\tau_2(\delta) = 0$ there exist K such that for all $k > K$:

$$\begin{aligned} 2^{k\left(H_{\tilde{P}_{Y^n}}(Y^n)-\tau_1(\varepsilon)\right)} &\le \left|T_\varepsilon^k(\tilde{P}_{Y^n})\right| \\ &\le \left|T_\delta^{nk,(2)}(\tilde{P}_{X,Y},1)\right| \\ &\le 2^{nk\left(H_{\tilde{P}_{Y|X}}(Y|X)+\tau_2(\delta)\right)}. \end{aligned}$$

So

$$\frac{1}{n}\left(H_{\tilde{P}_{Y^n}}(Y^n)-\tau_1(\varepsilon)\right) \le \left(H_{\tilde{P}_{Y|X}}(Y \mid X)+\tau_2(\delta)\right),$$

or, alternatively,

$$H_{\tilde{P}_{Y^n}}(Y^n) \le nH_{\tilde{P}_{Y|X}}(Y \mid X)+\zeta_n \quad , \quad \lim_{n\to\infty}\zeta_n = 0.$$

We can think of the single letter distribution $\tilde{P}_{Y|X}$ as $\tilde{P}_{Y_i,Y_{i-1}}(y_i \mid y_{i-1})$ and define $\tilde{Q}_{Y^n}(y^n) = \prod_{i=1}^n \tilde{P}_{Y_i,Y_{i-1}}(y_i \mid y_{i-1})$. □

Appendix C
Proofs for the Feedback Capacity of the $(1, 2)$-RLL BEC

Proof of Lemma 9 The definition of the scheme clearly shows that if $Y_i = 1$ then $X_{i+1} = 0$, and if $Y_{i-1} = Y_i = 0$ then $X_{i+1} = 1$. All that remains is to prove that it is possible to assign δ of the unit interval to '0' in the case of consecutive erasures. We prove this using induction on the number of erasures, n. As a base case take $n = 3$. Fig. 15 shows possible partitions of the unit interval that comply with the definition of the coding scheme. The last partition in Fig. 15 assumes that $0 \le 2\delta - 1 \le 1 - \delta$. This means that $\frac{1}{2} \le \delta \le \frac{2}{3}$.

For $n \ge 4$ consecutive erasures consider the following:

1) Subintervals of $[0, 1]$ which were labeled '1' during the previous channel use must now be labeled '0'. Additionally, the total length of these subintervals is $1 - \delta$. This means that we must assign an additional $2\delta - 1$ to '0'.
2) Subintervals of $[0, 1]$ which were labeled '1' two channel uses ago are now unconstrained. The total length of these subintervals is also $1 - \delta$.
3) The two sets of subintervals mentioned in items 1) and 2), above, are disjoint.

Given that $\delta \le \frac{2}{3}$ we can assign $2\delta - 1$ of the unconstrained subintervals to '0'.

We calculate the rate achieved by this scheme using the same method shown in Section IV. In the case of this coding scheme, and after exchanging δ with $1 - \delta$, we reach the following rate:

$$R = \frac{H_2(\delta)}{\frac{1}{1-\varepsilon} + \bar{\varepsilon} + \delta}. \tag{44}$$

□

Proof of Lemma 10 Eq. (21) contains an upper bound to $C^{\text{fb}}_{(1,2)}(\varepsilon)$. By partially deriving the RHS of Eq. (21), it can be shown that the maximum is attained for $\delta_2 = 1 - \delta_1$. Substituting this relation into Eq. (21) gives:

$$C^{\text{fb}}_{(1,2)} \le \max_{0\le\delta\le 1} \frac{H_2(\delta)}{\frac{1}{1-\varepsilon} + \bar{\varepsilon} + \delta}. \tag{45}$$

Since $H_2(x) = H_2(1 - x)$, it is clear that the maximum is in $0 \le \delta \le \frac{1}{2}$. The derivative of Eq. (45) is equal to zero only when

$$(1-\delta)^{\frac{1}{1-\varepsilon}+\bar{\varepsilon}+1} = \delta^{\frac{1}{1-\varepsilon}+\bar{\varepsilon}}. \tag{46}$$

The LHS and RHS of Eq. (46) are, respectively, monotonic decreasing and monotonic increasing functions of δ. In order for the maximizing δ to be at least $\frac{1}{3}$, we need to prove that for any ε

$$\left(\frac{2}{3}\right)^{\frac{1}{1-\varepsilon}+\bar{\varepsilon}+1} \ge \left(\frac{1}{3}\right)^{\frac{1}{1-\varepsilon}+\bar{\varepsilon}}, \tag{47}$$

which simplifies to

$$2^{\frac{1}{1-\varepsilon}+\bar{\varepsilon}+1} \ge 3. \tag{48}$$

Since the power increases in ε, it is sufficient to check that Eq. (48) holds for $\varepsilon = 0$, and indeed $2^3 = 8 \ge 3$. □

Acknowledgement

The authors would like to thank the associate editor and the anonymous reviewers for their valuable and constructive comments, which helped to improve the paper and its presentation considerably.

References

[1] O. Peled, O. Sabag, and H. H. Permuter, "Feedback capacity and coding for the $(0, k)$-RLL input-constrained BEC," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1783–1787.

[2] B. H. Marcus, R. M. Roth, and P. H. Siegel, "Constrained systems and coding for recording channels," in *Handbook Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998, pp. 1635–1764.

[3] K. A. S. Immink, "Runlength-limited sequences," *Proc. IEEE*, vol. 78, no. 11, pp. 1745–1759, Nov. 1990.

[4] Y. Kim *et al.*, "Modulation coding for flash memories," in *Proc. IEEE Int Conf. Comput. Netw. Commun. (ICNC)*, Jan. 2013, pp. 961–967.

[5] M. Schwartz and A. Vardy, "New bounds on the capacity of multidimensional run-length constraints," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4373–4382, Jul. 2011.

[6] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on two-dimensional constraints," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 824–838, May 2004.

[7] O. Sabag, H. H. Permuter, and N. Kashyap, "The feedback capacity of the binary erasure channel with a no-consecutive-ones input constraint," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 8–22, Jan. 2016.

[8] S. Tatikonda and S. Mitter, "The capacity of channels with feedback," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 323–349, Jan. 2009.

[9] J. Chen and T. Berger, "The capacity of finite-state Markov channels with feedback," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 780–789, Mar. 2005.

[10] S. Yang, A. Kavčić, and S. Tatikonda, "Feedback capacity of finite-state machine channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 799–810, Mar. 2005.

[11] H. H. Permuter, P. Cuff, B. V. Roy, and T. Weissman, "Capacity of the trapdoor channel with feedback," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3150–3165, Jul. 2009.

[12] J. Wu and A. Anastasopoulos, "On the capacity of the general trapdoor channel with feedback," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 2256–2260.

[13] O. Elishco and H. Permuter, "Capacity and coding for the Ising channel with feedback," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5138–5149, Sep. 2014.

[14] A. Sharov and R. M. Roth, "On the capacity of generalized ising channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2338–2356, Apr. 2017.

[15] O. Sabag, H. H. Permuter, and N. Kashyap, "Feedback capacity and coding for the BIBO channel with a no-repeated-ones input constraint," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 4940–4961, Jul. 2018.

[16] M. Horstein, "Sequential transmission using noiseless feedback," *IEEE Trans. Inf. Theory*, vol. IT-9, no. 3, pp. 136–143, Jul. 1963.

[17] O. Shayevitz and M. Feder, "Optimal feedback communication via posterior matching," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1186–1222, Mar. 2011.

[18] C. T. Li and A. El Gamal, "An efficient feedback coding scheme with low error probability for discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 2953–2963, Jun. 2015.

[19] O. Shayevitz and M. Feder, "A simple proof for the optimality of randomized posterior matching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3410–3418, Jun. 2016.

[20] M. Naghshvar, T. Javidi, and M. Wigger, "Extrinsic Jensen–Shannon divergence: Applications to variable-length coding," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 2148–2164, Apr. 2015.

[21] J. H. Bae and A. Anastasopoulos, "A posterior matching scheme for finite-state channels with feedback," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 2338–2342.

[22] O. Sabag, H. H. Permuter, and H. D. Pfister, "A single-letter upper bound on the feedback capacity of unifilar finite-state channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1392–1409, Mar. 2017.

[23] I. Csiszár, "The method of types [information theory]," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2505–2523, Oct. 1998.

Ori Peled (S'17) received his B.Sc. degrees (cum laude) in Mathematics and in Electrical and Computer Engineering from Ben-Gurion University of the Negev, Israel, in 2015. He is currently pursuing his M.Sc in Electrical and Computer Engineering at the same institution under the guidance of Prof. Permuter, and working at Ceragon Networks as an algorithms engineer.

Ori is a recipient of the ISIT-2017 best student paper award.

Oron Sabag (S'14) received his B.Sc. (cum laude) degree and his M.Sc. (summa cum laude) in Electrical and Computer Engineering from the Ben-Gurion University of the Negev, Israel, in 2013 and 2016, respectively. He is currently pursuing his Ph.D in the direct track for honor students in Electrical and Computer Engineering at the same institution.

Oron is a recipient of several awards, among them are the Lachish Fellowship, SPCOM-2016 best student paper award, ISIT-2017 best student paper award, and a Feder Family Award for outstanding research in communications (3rd prize).

Haim H. Permuter (M'08–SM'13) received his B.Sc. (summa cum laude) and M.Sc. (summa cum laude) degrees in Electrical and Computer Engineering from the Ben-Gurion University, Israel, in 1997 and 2003, respectively, and the Ph.D. degree in Electrical Engineering from Stanford University, California in 2008. Between 1997 and 2004, he was an officer at a research and development unit of the Israeli Defense Forces. Since 2009 he is with the department of Electrical and Computer Engineering at Ben-Gurion University where he is currently a professor, Luck-Hille Chair in Electrical Engineering. Haim also serves as head of the communication track in his department. Prof. Permuter is a recipient of several awards, among them the Fullbright Fellowship, the Stanford Graduate Fellowship (SGF), Allon Fellowship, and the U.S.-Israel Binational Science Foundation Bergmann Memorial Award. Haim served on the editorial boards of the IEEE TRANSACTIONS ON INFORMATION THEORY in 2013-2016.