Channels with Input-Correlated Synchronization Errors*

Roni Con† João Ribeiro‡

Abstract

"Independent and identically distributed" errors do not accurately capture the noisy behavior of real-world data storage and information transmission technologies. Motivated by this, we study channels with *input-correlated* synchronization errors, meaning that the distribution of synchronization errors (such as deletions and insertions) applied to the i-th input x_i may depend on the whole input string x.

We begin by identifying conditions on the input-correlated synchronization channel under which the channel's information capacity is achieved by a stationary ergodic input source and is equal to its coding capacity. These conditions capture a wide class of channels, including channels with correlated errors observed in DNA-based data storage systems and their multi-trace versions, and generalize prior work. To showcase the usefulness of the general capacity theorem above, we combine it with techniques of Pernice-Li-Wootters (ISIT 2022) and Brakensiek-Li-Spang (FOCS 2020) to obtain explicit capacity-achieving codes for multi-trace channels with *runlength-dependent deletions*, motivated by error patterns observed in DNA-based data storage systems.

*A preliminary version of this work has been accepted for presentation at ISIT 2025.

†Department of Computer Science, Technion - Israel Institute of Technology, Haifa. roni.con93@gmail.com

‡Instituto de Telecomunicações and Departamento de Matemática, Instituto Superior Técnico, Universidade de Lisboa. jribeiro@tecnico.ulisboa.pt

Contents

1 Introduction

Errors which cause loss of synchronization between sender and receiver, such as deletions, insertions, and replications, occur in various communications and data storage technologies, with DNA-based data storage being a notable recent example. Despite considerable effort, pinning down the capacity and designing efficient nearly-optimal codes for channels with synchronization channels remain major problems in information and coding theory. The surveys by Mitzenmacher [Mit09], Mercier, Bhargava, and Tarokh [MBT10], and Cheraghchi and Ribeiro [CR21] provide in-depth discussions of the many challenges encountered when dealing with synchronization errors.

Most prior work on channels with synchronization errors has focused on i.i.d. errors. However, synchronization errors in real-world systems do not satisfy this assumption. For example, it has been observed in empirical analyses of widely used DNA sequencing technologies [RRC+13, HMG19] that short substrings of DNA strands (which are written using a 4-symbol alphabet A, C, G, T) with either a very high (above 75%) or very low (below 25%) concentration of G's and C's experience higher deletion rates [RRC+13, Figure 4], and that longer runs of the same symbol in DNA strands experience higher deletion rates than shorter runs during sequencing [RRC+13, Figure 5]. More broadly, a long series of works has explored, among many other things, the types of errors and error rates that occur in such systems, using different synthesis and sequencing technologies [GBC+13, RRC+13, GHP+15, TYYM+15, BLC+16, EZ17, YGM17, OAC+18, HMG19, PHJ+20, SGPY21].

Motivated by this, we study a general class of channels with synchronization errors where the error distribution of the i-th input symbol x_i may depend on the whole input x. We also consider "multi-trace" versions of these channels, where the input x is sent through multiple independent channels, generating multiple channel outputs (*traces*) at the receiver end, which is especially relevant in DNA-based data storage.

1.1 Our contributions

Capacity theorems for channels with input-correlated synchronization errors Our first contribution, in Section 3, is a capacity theorem for a general class of channels with input-correlated synchronization errors, which we call *admissible channels*. The precise definition of an admissible channel is given in Section 3.1. As we show afterwards in Section 4, this class includes as special cases the channel model of Mao, Diggavi, and Kannan [MDK18], multi-trace channels with input-correlated synchronization errors, and, more concretely, channels with runlength-dependent deletions where bits in runs of length ℓ are deleted independently with probability $d(\ell)$, for an arbitrary function $d : \mathbb{N} \to [0, 1]$. Channels with runlength-dependent deletions will be our concrete running example throughout this paper to showcase the usefulness of this result.

Theorem 1 (Informal, see Theorem 4 for a formal statement)**.** *Let Z be an admissible channel (see Section 3.1). Then, its information capacity equals its coding capacity, and the information capacity is achieved by stationary ergodic sources.*

Efficient capacity-achieving codes for single- and multi-trace runlength-dependent deletion channels. As observed by Pernice, Li, and Wootters [PLW22], a standard but quite useful consequence of Theorem 1 is that admissible channels have capacity-achieving codes with additional structure. In particular, admissible channels Z with binary input alphabet have capacity-achieving codes $\mathcal{C}$ such that every short substring of every codeword $c \in \mathcal{C}$ has not-too-small Hamming weight (see Theorem 5 for a formal statement).

To showcase the usefulness of this consequence of Theorem 1, we use it to obtain efficient capacity-achieving codes for channels with "bounded" runlength-dependent deletions, *in both the*

single-trace and multi-trace settings. More precisely, a bounded runlength-dependent deletion channel is a runlength-dependent deletion channel with a non-decreasing deletion probability function $d : \mathbb{N} \to [0, 1]$, and such that there exists an integer M such that $d(\ell) = d(M) < 1$ for all $\ell \geq M$ (see Definition 11 for a formal definition). Its *T-trace version* is the channel that on input x outputs $Y_1, \ldots, Y_T$ (called *traces*), where the Y_i's are independent and identically distributed like outputs of the bounded runlength-dependent deletion channel on input x.

In the single-trace setting (Section 5) we combine the structured codes resulting from Theorem 1 with an approach originally applied to channels with i.i.d. synchronization errors in [PLW22] to obtain efficient capacity-achieving codes for any bounded runlength-dependent deletion channel.

Theorem 2 (Efficient capacity-achieving single-trace codes, informal. See Theorem 11 for a formal statement)**.** *Let* Ch *be a bounded runlength-dependent deletion channel. Denote its capacity by* Cap *and let* $\varepsilon > 0$ *be an arbitrary constant. Then, there exists a family of binary codes* $\{C_n\}_{n=1}^{\infty}$*, where* C_n *has blocklength* n *and rate* $R_n > \mathsf{Cap} - \varepsilon$ *for all large enough* n*, that is robust for this channel. Moreover, the* C_n*'s are encodable in linear time and decodable in quasi linear time in the message length.*

As we discuss in more detail in Section 5, this result can be generalized well beyond channels with bounded runlength-dependent deletions. Broadly speaking, the theorem can be extended to any runlength-dependent deletion channel that satisfies two key conditions. First, with high probability, a *very long* run of zeros will remain long after passing through the channel. These long zero runs help maintain some degree of synchronization between the sender and the receiver. Second, in any interval that is not too short and contains a high density of 1s, the probability that *all* of the 1s are deleted is very small. In this paper, we focus on the bounded runlength-dependent deletions (where the deletion function is non-decreasing) to keep the analysis simpler and because it is more realistic in practice for longer runs to experience higher deletion rates.

In the multi-trace setting (Section 6) we again rely on Theorem 1, which in particular implies the existence of structured codes attaining the capacity of the *multi-trace* bounded runlength-dependent channel. Then, to achieve efficient codes with matching rate, we combine it with the main idea of Brakensiek, Li, and Spang [BLS20], with some minor modifications, who showed how to compile average-case trace reconstruction algorithms into efficient rate $1 - o(1)$ codes for the coded trace reconstruction problem. This yields efficient capacity-achieving codes for any t-trace bounded runlength-dependent deletion channel.

Theorem 3 (Efficient capacity-achieving multi-trace codes, informal. See Theorem 13 for a formal statement)**.** *Let* Ch *be a T-trace bounded runlength-dependent deletion channel, for an arbitrary constant integer* $T \geq 1$*. Denote its capacity by* Cap *and let* $\varepsilon > 0$ *be an arbitrary constant. Then, there exists a family of binary codes* $\{C_n\}_{n=1}^{\infty}$*, where* C_n *has blocklength* n *and rate* $R_n > \mathsf{Cap} - \varepsilon$ *for all large enough* n*, that is robust for this channel. Moreover, the* C_n*'s are encodable in linear time and decodable in quadratic time in the message length.*

The difference between Theorem 2 and Theorem 3 is that the former guarantees better decoding complexity in the single-trace ($T = 1$) setting. We see Theorem 2 as a natural warmup towards the multi-trace result in Theorem 3, as its proof is significantly simpler.

Capacity lower bounds for runlength-dependent deletion channels. Our results presented above effectively allow us to turn any capacity lower bound for bounded runlength-dependent deletion channels into efficiently encodable and decodable codes with that rate. To complement this, in Section 7 we study concrete (single-trace) capacity lower bounds on arguably the simplest class

of bounded runlength-dependent deletion channels: For a threshold $\tau \geq 1$ and $d \in [0,1]$, consider the runlength-dependent deletion channel that independently deletes each bit in a run of length at least τ with probability d, and does not apply any deletions to bits in runs of length less than τ. The case $\tau = 1$ recovers the standard i.i.d. deletion channel.

A naive approach towards lower bounding the capacity of these channels is to take the largest *runlength-limited* code, whose codewords only have runs of length less than τ. The maximal rate of such codes as a function of τ is well known. We obtain capacity lower bounds that improve on this baseline approach for a large range of d, as illustrated in Figures 1 and 2 for thresholds $\tau = 2$ and $\tau = 3$, respectively. By Theorem 2 we automatically get nearly-linear time encodable and decodable codes with that rate.

1.2 Related work

Capacity theorems for channels with synchronization errors. The first work to study this topic was by Dobrushin [Dob67], who obtained capacity theorems for channels that apply i.i.d. synchronization errors. Recently, there has been interest in extending such capacity theorems beyond i.i.d. errors. Mao, Diggavi, and Kannan [MDK18] consider channels combining synchronization errors and (bounded) intersymbol interference as a model of nanopore-based sequencing. More precisely, the behavior of the channel on input x_i is some function of $x_i, x_{i-1}, \ldots, x_{i-\ell}$, for some memory threshold ℓ. They show that the information and coding capacity of these channels coincide, generalizing Dobrushin's result [Dob67], but do not show that capacity is achieved by a stationary ergodic (or Markov) source. Capacity theorems for a related (more concrete) model of nanopore-based sequencing with noisy duplications have also been studied by McBain, Saunderson, and Viterbo [MVS24, MSV24]. Li and Tan [LT21] consider a channel obtained from the concatenation of a standard deletion channel with i.i.d. deletions and a finite-state discrete memoryless channel. They show that the capacity of this channel is achieved by Markov processes, which implies that the polar codes developed by Tal, Pfister, Fazeli, and Vardy [TPFV22] achieve capacity on the i.i.d. deletion channel (a special case of this result was proved earlier in [TPFV22]). Morozov and Duman [MD24, MD25] show that information and coding capacities coincide for channels that introduce deletions and insertions with Markovian memory, in the sense that the behavior of the channel on the i-th input bit depends on the current state of an underlying stationary ergodic finite state Markov chain (whose states are updated independently of past inputs). In [MD25] they also give capacity upper bounds for the special case of a deletion channel with Markov memory where the deletion probability applied independently to each input bit varies between a "low value" and a "high value" according to a 2-state Markov chain (that evolves independently of the channel input).

The models we study are incomparable to those of [LT21, MD24, MD25], and our models and capacity theorems generalize those of [MDK18]. We discuss the relationship to the model and results of [MDK18] in more detail. In [MDK18], the channel behavior on the i-th input bit x_i may depend only on a *bounded* window of input bits. In contrast, in our channel model the error distribution for the i-th input bit x_i is some function of *the whole input* x, satisfying some additional assumptions. We show in Section 4.1 that the channel model from [MDK18] satisfies the assumptions required for the application of our capacity theorems, and so our results generalize the corresponding results of [MDK18]. Moreover, we show that stationary ergodic sources achieve the information capacity of these channels, which is particularly relevant for constructing efficient capacity-achieving codes.

Furthermore, our framework also captures interesting scenarios that fall outside the scope of [MDK18]. In Section 4.2 we show that our framework implies capacity theorems for *multi-trace* channels with correlated synchronization errors, where on input x the receiver learns multiple i.i.d. realizations of the channel output $Z(x)$. This setting is especially relevant in the context of

DNA-based data storage systems with nanopore-based sequencing [CGMR20, BLS20]. Also, in Section 4.4 we show that our framework includes as special cases deletion channels where the deletion of a bit x_i may depend arbitrarily on the length of the run where x_i is included (in particular, beyond a bounded window around x_i).

Efficient coding for channels with synchronization errors. There has been significant interest in the design of efficiently encodable and decodable codes for channels with synchronization errors. We discuss the progress most relevant to our work. Guruswami and Li [GL19] and later Con and Shpilka [CS22] obtained efficient codes for the i.i.d. binary deletion channel with rate $\Theta(1-d)$, where d is the deletion probability. Later, Tal, Pfister, Fazeli, and Vardy [TPFV22, PT21] and later Tal and Arava [AT23] (combined with a result from [LT21]) designed efficient polar codes achieving the capacity of a family of channels with i.i.d. insertions, deletions, and substitutions, generalizing the i.i.d. deletion channel. Other constructions of efficient capacity-achieving codes for channels with i.i.d. synchronization errors were presented in [Rub22, PLW22], which achieve slightly faster decoding and slightly smaller decoding error probability than the polar coding constructions. Of particular note, the general framework of Pernice, Li, and Wootters [PLW22] yields efficient capacity-achieving codes for a large class of *repeat channels* – these are channels that independently replicate each input bit according to some replication distribution over the naturals (e.g., Bernoulli, Poisson, geometric).[1] This was accomplished by combining a marker-based construction with the capacity theorem of Dobrushin [Dob67], which applies to i.i.d. repeat channels.

Some works [CGMR20, BLS20] have studied efficiently encodable and decodable codes for the multi-trace i.i.d. deletion channel. Their focus is on the case where the number of traces is allowed to grow with the blocklength of the code. In contrast, in this work we consider the number of traces to be a fixed constant, and we are then interested in devising efficient codes for that fixed number of traces with rate as large as possible. These two settings are incomparable. Srinivasavaradhan, Gopi, Pfister, and Yekhanin [SGPY21] study, among other things, codes for multi-trace channels with i.i.d. insertions, deletions, and substitutions with a fixed number of traces. However, their focus is different from ours and they only provide heuristic reconstruction procedures.

A common feature of the works discussed above is that they only consider channels with i.i.d. synchronization errors. In contrast, we study channels with correlated synchronization errors. In particular, we obtain a capacity theorem that applies to a wide class of (multi-trace) channels with correlated synchronization errors, which we show can be used to design efficient capacity-achieving codes for synchronization channels with relevant correlations.

2 Preliminaries

2.1 Notation

We denote random variables by uppercase roman letters such as X, Y, and Z. In this work, we will only work with random variables supported on discrete sets. We use $X \to Y \to Z$ to denote the fact that these three random variables form a Markov chain (i.e., Z is conditionally independent of X given Y), and write $X \sim Y$ if random variables X and Y follow the same distribution. We use $\mathbb{E}[X]$ to denote the expected value of a random variable X supported on a subset of $\mathbb{R}$, and $H(X)$ to denote its Shannon entropy.

[1] The existence of such efficient capacity-achieving codes does not mean that we now can *determine* the capacity of these channels. We see the contribution of [TPFV22, PT21, Rub22, PLW22] mainly as turning capacity lower bounds into efficient codes with the corresponding rate.

For a sequence $x = (x_i)_{i \in \mathbb{N}}$, we use x_m^n to denote the subsequence $x_m, x_{m+1}, \ldots, x_n$. We use log to denote the base-2 logarithm. For an integer $n \geq 1$, we write $[n] = \{1, 2, \ldots, n\}$.

2.2 Channels

We consider channels Z with finite input alphabet Σ_{in} and discrete output alphabet Σ_{out}. On input $x \in \Sigma_{\mathsf{in}}^n$, the channel outputs $Z(x)$ according to some probabilistic rule. We allow the support of $Z(x)$ to be quite general – we will consider settings where $Z(x) \in \Sigma_{\mathsf{out}}^*$ for some discrete output alphabet Σ_{out}, but also settings where $Z(x)$ corresponds to multiple channel traces with input x, in which case $Z(x) \in (\Sigma_{\mathsf{out}}^*)^t$ for some integer $t \geq 1$.

Later on we will impose constraints on the behavior of $Z(x)$ to obtain capacity theorems. We will also write "$Z(X), Z(Y)$" to mean that Z is applied independently to the inputs X and Y.

2.3 Entropy and information rates for stochastic processes and notions of capacity

In this section, we define various types of "channel capacity", and state some basic bounds.

Definition 1 (Entropy rate)**.** *For a process $X = (X_i)_{i \in \mathbb{N}}$, we define the* entropy rate *of X, denoted by $H(X)$, as*

$$H(X) = \liminf_{n \to \infty} \frac{H(X_1^n)}{n}.$$

Definition 2 (Information rate)**.** *For a channel Z and input process $X = (X_i)_{i \in \mathbb{N}}$, we define the* information rate *achievable by X over Z as*

$$I(X; Z(X)) = \liminf_{n \to \infty} \frac{I(X_1^n; Z(X_1^n))}{n}.$$

Definition 3 (Information capacity)**.** *Given a channel Z, we define its* information capacity*, denoted by $\mathsf{ICap}(Z)$, as*

$$\mathsf{ICap}(Z) = \liminf_{n \to \infty} \sup_{P_{X_1^n}} \frac{I(X_1^n; Z(X_1^n))}{n},$$

where the supremum is taken over all distributions of input processes $X = (X_i)_{i \in \mathbb{N}}$.

We now introduce some useful definitions about stochastic processes.

Definition 4 (Block-independent process)**.** *We say that a stochastic process $X = (X_i)_{i \in \mathbb{N}}$ is* block-independent *with blocklength b if for any integer $t \geq 1$ and $n = tb$ we have*

$$\Pr[X_1^n = x_1^n] = \prod_{i=1}^{t} \Pr[X_1^b = x_{(i-1)b+1}^{ib}].$$

Definition 5 (Stationary ergodic process)**.** *We say that a stochastic process $X = (X_i)_{i \in \mathbb{N}}$ is* stationary *if $(X_1, \ldots, X_n) \sim (X_{1+\tau}, \ldots, X_{n+\tau})$ for any integers $n, \tau \in \mathbb{N}$. Moreover, we say that X is* stationary ergodic *if X is stationary and for any function $f \in L^1$ we almost surely have*

$$\mathbb{E}[f(X_1)] = \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} f(X_j).$$

Definition 6 (Stationary capacity)**.** *Given a channel Z, we define its* stationary capacity*, denoted by* $\mathsf{SCap}(Z)$*, as*

$$\mathsf{SCap}(Z) = \sup_X I(X; Z(X)),$$

where the supremum is taken over all stationary ergodic input processes $X = (X_i)_{i\in\mathbb{N}}$.

An input process $X = \{X_i\}_{i\in\mathbb{N}}$ is *m-th order Markov* if

$$\Pr[X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1] = \Pr[X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_{n-m} = x_{n-m}]$$

for any n and $x_1, \ldots, x_n$.

Definition 7 (m-th order Markov capacity)**.** *Given a channel Z, we define its* m-th order Markov capacity*, denoted by* $\mathsf{SCap}^{(m)}(Z)$*, as*

$$\mathsf{SCap}^{(m)}(Z) = \sup_X I(X; Z(X)),$$

where the supremum is taken over all m-th order stationary Markov input processes $X = (X_i)_{i\in\mathbb{N}}$.

Remark 1. We have $\mathsf{ICap}(Z) \geq \mathsf{SCap}(Z) \geq \mathsf{SCap}^{(m)}(Z)$ for any channel Z and any integer $m \geq 0$.

Before we define the coding capacity of a channel Z, we need some auxiliary definitions.

Definition 8 ((n, R, ε)-code for a channel)**.** *Let Z be a channel with finite input alphabet Σ and output alphabet Σ_{out}. We say that $\mathcal{C} \subseteq \Sigma^n$ is an (n, R, ε)-*code for Z *if $|\mathcal{C}| \geq 2^{Rn}$ and there exists a deterministic function $\mathsf{Dec} : \Sigma_{\mathsf{out}}^* \to \Sigma^n$ such that $\Pr[\mathsf{Dec}(Z(C)) \neq C] \leq \varepsilon$, where C is uniformly distributed over $\mathcal{C}$ (i.e., the average decoding error probability of $\mathcal{C}$ is at most ε).*

Definition 9 (Achievable rate)**.** *Let Z be a channel. We say that a real number $R > 0$ is an* achievable rate for Z *if there exists a family of codes $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ and an integer n_0 such that each $\mathcal{C}_n$ is an (n, R_n, ε_n)-code for Z with $R_n \geq R$ for all $n \geq n_0$ and $\lim_{n\to\infty} \varepsilon_n = 0$.*

Definition 10 (Coding capacity)**.** *Given a channel Z, we define its* coding capacity*, denoted by* $\mathsf{CCap}(Z)$*, as the supremum of all $R \geq 0$ that are achievable rates for Z.*

Remark 2. It follows via Fano's inequality that $\mathsf{CCap}(Z) \leq \mathsf{ICap}(Z)$ for any channel Z. See, e.g., [PW24, Theorem 19.7].

2.4 A strengthening of Fekete's lemma

We will use the following strengthening of Fekete's lemma due to de Bruijn and Erdős [dBE52] (also used in [MD24]). See [FR20] for an excellent discussion on this topic.

Lemma 1 (Strengthened Fekete's lemma [dBE52])**.** *Let $(a_n)_{n\in\mathbb{N}}$ be a sequence that is "almost" subadditive, in the sense that*

$$a_{n+m} \leq a_n + a_m + f(n+m)$$

for all $n \leq m \leq 2n$ and some f such that $\sum_{n\in\mathbb{N}} f(n)/n^2$ converges. Then, $\lim_{n\to\infty} a_n/n$ exists.

2.5 Concentration inequalities

In this paper, we shall use the following standard concentration inequalities.

Lemma 2 (Multiplicative Chernoff bound; see, e.g., [MU17, Theorems 4.4 and 4.5])**.** *Suppose that $X_1, \dots, X_n$ are i.i.d. random variables taking values in $\{0,1\}$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then, for any $0 < \alpha < 1$ we have*

$$\Pr[X > (1+\alpha)\mu] < e^{-\frac{\mu\alpha^2}{3}}$$

and

$$\Pr[X < (1-\alpha)\mu] < e^{-\frac{\mu\alpha^2}{2}}.$$

Lemma 3 (Hoeffding's inequality; see, e.g., [Ver18, Theorem 2.2.6])**.** *Suppose that $X_1, \dots, X_n$ are independent random variables with finite first and second moments and $a_i \le X_i \le b_i$ for $1 \le i \le n$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then, for any $t > 0$ we have*

$$\Pr[X - \mu > t] < \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

3 Capacity theorems for channels with input-correlated synchronization errors

3.1 Admissible channels

We will show capacity theorems for all channels Z (with finite input alphabet) for which there exists a special associated channel $Z^\star$ with the properties below. We call Z *admissible* (with respect to $Z^\star$) if the following holds.

1. **Bounded output entropy:** There exists a constant $c > 0$ such that for every input process $X = (X_i)_{i \in \mathbb{N}}$ and every integer $n \ge 1$ we have $H(Z^\star(X_1^n)) \le cn$.
2. **Rate preservation:** For any process $X = \{X_i\}_{i \in \mathbb{N}}$ we have $I(X; Z(X)) = I(X; Z^\star(X))$.
3. **Concatenation:** For any process $X = \{X_i\}_{i \in \mathbb{N}}$ and indices $1 \le m \le n$, we have that
$$X_1^n \to Z^\star(X_1^m), Z^\star(X_{m+1}^n) \to Z^\star(X_1^n).$$
4. **Partition:**
   (a) **Prefix/suffix-partition:** Fix any integer $\tau \ge 1$. Then, there exists a non-decreasing sequence $\gamma_m = o(m)$ such that $\sum_{m \in \mathbb{N}} \gamma_m / m^2$ converges and for any process $X = \{X_i\}_{i \in \mathbb{N}}$ and integer $n \ge \tau$ there exist random variables W_{pre} and W_{suf} such that (1) $X_1^n \to Z^\star(X_1^n), W_{\text{pre}} \to Z^\star(X_1^\tau), Z^\star(X_{\tau+1}^n)$, (2) $X_1^n \to Z^\star(X_1^n), W_{\text{suf}} \to Z^\star(X_1^{n-\tau}), Z^\star(X_{n-\tau+1}^n)$, and (3) $H(W_{\text{pre}}), H(W_{\text{suf}}) \le \gamma_n$.
   (b) **Amortized block-partition:** There exists a sequence $\alpha_m = o(m)$ such that for any process $X = \{X_i\}_{i \in \mathbb{N}}$, blocklength b, and number of blocks t there exists a random variable W such that $X_1^{tb} \to Z^\star(X_1^{tb}), W \to Z^\star(X_1^b), \dots, Z^\star(X_{(t-1)b+1}^{tb})$, and $H(W) \le t \cdot \alpha_b$.

5. **Amortized preimage size:** There exists a sequence $\beta_m = o(m)$ such that for any process $X = \{X_i\}_{i\in\mathbb{N}}$, blocklength b, and number of blocks t, there exists a random variable Y such that $X_1^{tb} \to Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}) \to Y$ and a deterministic function ϕ such that

$$Z^\star(X_1^{tb}) = \phi(Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}), Y)$$

and $\max_z \log|\phi^{-1}(z)| \le t \cdot \beta_b$.

We will prove the following results for admissible channels.

Theorem 4 (Capacity theorem for admissible channels)**.** *Suppose that Z is an admissible channel. Then,*

$$\mathsf{ICap}(Z) = \mathsf{SCap}(Z) = \lim_{m\to\infty} \mathsf{SCap}^{(m)}(Z) = \mathsf{CCap}(Z).$$

The proof of Theorem 4 proceeds via a series of theorems, where we adapt and generalize approaches from [Dob67, LT21, MD24]. The equality $\mathsf{ICap}(Z) = \mathsf{SCap}(Z)$ corresponds to Theorem 6 in Section 3.3. The equality $\mathsf{ICap}(Z) = \lim_{m\to\infty} \mathsf{SCap}^{(m)}(Z)$ corresponds to Theorem 7 in Section 3.4. The equality $\mathsf{ICap}(Z) = \mathsf{CCap}(Z)$ corresponds to Theorem 9 in Section 3.5.

The next result follows in a standard manner from Theorem 4 using the approach of [PLW22], and is relevant for the construction of efficient capacity-achieving codes for admissible channels. For simplicity, we present it for channels with binary input alphabet. We prove this result in Section 3.5.3.

Theorem 5 (Dense capacity-achieving codes for admissible channels)**.** *Let Z be an admissible channel with binary input alphabet. Then, for any $\varepsilon, \zeta > 0$ there exist $\gamma \in (0, 1/2)$ and integers $b = b(\varepsilon, \zeta)$ and $t(\varepsilon, \zeta)$ depending only on ε and ζ such that for all $t \ge t(\varepsilon, \zeta)$ there exists a code $\mathcal{C}$ with blocklength $n = t \cdot b$, rate $R \ge \mathsf{ICap}(Z) - \varepsilon$, and maximal decoding error probability ε over Z such that for all codewords $c \in \mathcal{C}$ we have $\gamma\zeta n \le w(c_i^{i+\zeta n}) \le (1-\gamma)\zeta n$ for all $i \in [(1-\zeta)n]$, where $w(\cdot)$ denotes the Hamming weight.*

The properties for admissibility laid out above are sufficient for Theorems 4 and 5 to hold, but it is conceivable that they are not necessary. We leave it as an interesting direction for future work to simplify the set of sufficient properties.

3.2 Existence of relevant limits for admissible channels

Let Z be an arbitrary admissible channel with respect to $Z^\star$. Via applications of Fekete's lemma (Lemma 1), we begin by showing that the limit inferior in the definitions of capacities and information rates can be replaced by a limit.

Lemma 4. *If Z is an admissible channel with respect to $Z^\star$, then*

$$\mathsf{ICap}(Z) = \mathsf{ICap}(Z^\star) = \lim_{n\to\infty} \sup_{X_1^n} \frac{I(X_1^n; Z^\star(X_1^n))}{n},$$

and the limit on the right-hand side exists.

Proof. By Lemma 1, it suffices to show that the sequence

$$a_n = \sup_{X_1^n} I(X_1^n; Z^\star(X_1^n))$$

is subadditive, i.e., $a_{n+m} \le a_n + a_m$ for all n, m. Using Item 3, the data processing inequality gives

$$\begin{aligned}
a_{n+m} &= \sup_{X_1^{n+m}} I(X_1^{n+m}; Z^\star(X_1^{n+m})) \\
&\le \sup_{X_1^{n+m}} I(X_1^{n+m}; Z^\star(X_1^n), Z^\star(X_{n+1}^{n+m})) \\
&= \sup_{X_1^{n+m}} [I(X_1^{n+m}; Z^\star(X_1^n)) + I(X_1^{n+m}; Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n))] \\
&= \sup_{X_1^{n+m}} [I(X_1^{n}; Z^\star(X_1^n)) + I(X_1^{n+m}; Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n))] \\
&\le \sup_{X_1^{n+m}} [I(X_1^{n}; Z^\star(X_1^n)) + I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m}))] \\
&\le \sup_{X_1^{n}} I(X_1^{n}; Z^\star(X_1^n)) + \sup_{X_{n+1}^{n+m}} I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m})) \\
&= a_n + a_m.
\end{aligned}$$

The first inequality uses Item 3. The second inequality uses the fact that $X_1^n, X_{n+1}^{n+m} \to X_{n+1}^{n+m} \to Z^\star(X_{n+1}^{n+m})$, and so

$$\begin{aligned}
I(X_1^n, X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n)) &= H(Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n)) - H(Z^\star(X_{n+1}^{n+m}) | X_1^n, X_{n+1}^{n+m}, Z^\star(X_1^n)) \\
&= H(Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n)) - H(Z^\star(X_{n+1}^{n+m}) | X_{n+1}^{n+m}) \\
&\le H(Z^\star(X_{n+1}^{n+m})) - H(Z^\star(X_{n+1}^{n+m}) | X_{n+1}^{n+m}) \\
&= I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m})).
\end{aligned}$$

The third inequality holds because the quantity on the fifth line is maximized by taking X_1^n and X_{n+1}^{n+m} to be independent, since $Z^\star$ is acting independently on each of X_1^n and X_{n+1}^{n+m}. □

Lemma 5. *Let X be an arbitrary stationary process and Z an admissible channel with respect to $Z^\star$. Then,*

$$I(X; Z(X)) = I(X; Z^\star(X)) = \lim_{n\to\infty} \frac{I(X_1^n; Z^\star(X_1^n))}{n},$$

the limit on the right-hand side exists, and $\mathsf{SCap}(Z) = \mathsf{SCap}(Z^\star)$.

Proof. Again, by Lemma 1 it suffices to show that $a_n = I(X_1^n; Z^\star(X_1^n))$ is a subadditive sequence. We have

$$\begin{aligned}
a_{n+m} &= I(X_1^{n+m}; Z^\star(X_1^{n+m})) \\
&\le I(X_1^{n+m}; Z^\star(X_1^n), Z^\star(X_{n+1}^{n+m})) \\
&= I(X_1^{n}, X_{n+1}^{n+m}; Z^\star(X_1^n)) + I(X_1^{n}, X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n)) \\
&= I(X_1^{n}; Z^\star(X_1^n)) + I(X_1^{n}, X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m}) | Z^\star(X_1^n)) \\
&\le I(X_1^{n}; Z^\star(X_1^n)) + I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m})) \\
&= I(X_1^{n}; Z^\star(X_1^n)) + I(X_1^{m}; Z^\star(X_1^m)) \\
&= a_n + a_m,
\end{aligned}$$

as desired. The first inequality uses Item 3. The second equality follows from the chain rule for mutual information. The third equality uses the fact that $X_1^n, X_{n+1}^{n+m} \to X_1^n \to Z^\star(X_1^n)$. The second inequality uses the fact that $X_1^n, X_{n+1}^{n+m} \to X_{n+1}^{n+m} \to Z^\star(X_{n+1}^{n+m})$, analogously to the proof of Lemma 4. The fourth equality uses the fact that X is stationary, and so $X_{n+1}^{n+m} \sim X_1^m$. □

Lemma 6. *Let X be a block independent process and suppose that Z is admissible with respect to $Z^\star$. Then,*

$$I(X;Z(X)) = I(X;Z^\star(X)) = \lim_{n\to\infty} \frac{I(X_1^n; Z^\star(X_1^n))}{n},$$

and the limit on the right-hand side exists.

Proof. Consider the sequence $a_n = I(X_1^n; Z^\star(X_1^n))$. Let b be the blocklength of X. Then, by Item 3, for any $m, n > b$ we have

$$\begin{aligned} a_{n+m} &= I(X_1^{n+m}; Z^\star(X_1^{n+m})) \\ &\le I(X_1^{n+m}; Z^\star(X_1^n), Z^\star(X_{n+1}^{n+m})) \\ &\le I(X_1^n; Z^\star(X_1^n)) + I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m})). \end{aligned}$$

Construct X' by trimming bits from the beginning of X_{n+1}^{n+m} so that $X' = X_{tb+1}^{n+m}$ for an integer t for which $n+1-(tb+1) \le b$. If m' denotes the length of X', from the block independence of X we get that

$$I(X'; Z^\star(X')) = I(X_1^{m'}; Z^\star(X_1^{m'})).$$

Since we trim $r < b$ symbols from X_{n+1}^{n+m} to obtain X', we have

$$\begin{aligned} I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{n+m})) &\le I(X_{n+1}^{n+m}; Z^\star(X_{n+1}^{tb}), Z^\star(X_{tb+1}^{n+m})) \\ &\le I(X_{tb+1}^{n+m}; Z^\star(X_{tb+1}^{n+m})) + H(Z^\star(X_{n+1}^{tb})) \\ &\le I(X_{tb+1}^{n+m}; Z^\star(X_{tb+1}^{n+m})) + cb \\ &= I(X_1^{m'}; Z^\star(X_1^{m'})) + cb \\ &\le I(X_1^m; Z^\star(X_1^{m'}), Z^\star(X_{m'+1}^m)) + cb \\ &\le I(X_1^m; Z^\star(X_1^m)) + \gamma_m + cb \end{aligned}$$

for a non-decreasing sequence $\gamma_m = o(m)$ such that $\sum_{m\in\mathbb{N}} \gamma_m/m^2$ converges. The first inequality uses Item 3. The third inequality uses Item 1 (more precisely, $H(Z^\star(X_{n+1}^{tb})) \le c(tb-(n+1)) \le cb$ for a fixed constant $c > 0$. The first equality uses the block independence of X. The fourth inequality follows from the chain rule for mutual information. The fifth inequality uses Item 4a with $\tau = r < b$.

Therefore, we conclude that for any $b < n \le m \le 2n$ we have $a_{n+m} \le a_n + a_m + f(n+m)$, where $f(n) = \gamma_n + cb$ (here, we use that γ_m is non-decreasing). Since $\sum_{n\in\mathbb{N}} f(n)/n^2$ converges (because $\sum_{n\in\mathbb{N}} \gamma_n/n^2$ converges by Item 4a), Lemma 1 implies that $\lim_{n\to\infty} a_n/n$ exists, as desired. □

3.3 Information capacity of admissible channels is achieved by stationary ergodic process

We show the following.

Theorem 6. *Suppose that the channel Z is admissible. Then,*

$$\mathsf{ICap}(Z) = \mathsf{SCap}(Z).$$

Suppose that Z is admissible with respect to $Z^\star$. By Item 2, it suffices to show that Theorem 6 holds for $Z^\star$. Fix an arbitrary $\varepsilon > 0$. Recalling Lemma 4, let b be a sufficiently large integer so that

$$\sup_{\tilde{X}_1^b} \frac{I(\tilde{X}_1^b; Z^\star(\tilde{X}_1^b))}{b} \ge \lim_{n\to\infty} \sup_{\tilde{X}_1^n} \frac{I(\tilde{X}_1^n; Z^\star(\tilde{X}_1^n))}{n} - \varepsilon = \mathsf{ICap}(Z^\star) - \varepsilon.$$

Furthermore, let X_1^b be such that

$$\frac{I(X_1^b; Z^\star(X_1^b))}{b} \geq \sup_{\tilde{X}_1^b} \frac{I(\tilde{X}_1^b; Z^\star(\tilde{X}_1^b))}{b} - \varepsilon \geq \mathsf{ICap}(Z^\star) - 2\varepsilon. \tag{1}$$

Let $p_{X_1^b}(\cdot)$ denote the correspoding PMF. Using the approach of [Fei59, LT21], consider the following process $\overline{X}$. First, define the block independent process $\hat{X} = \{\hat{X}_i\}_{i\in\mathbb{N}}$ with blocklength b and probability mass function

$$p_{\hat{X}_1^{tb}}(x_1^{tb}) = \prod_{i=1}^{t} p_{X_1^b}(x_{(i-1)b+1}^{ib}).$$

Write $\hat{X}_i^{[j]} = \hat{X}_{i+j}$. Let V be uniformly distributed over $\{0, 1, \ldots, b-1\}$, and set $\overline{X}_i = \hat{X}_i^{[V]} = \hat{X}_{i+V}$ for every $i \in \mathbb{N}$. Then, it holds that $\overline{X}$ is stationary and ergodic.

By Lemma 5, we know that $I(\overline{X}; Z^\star(\overline{X})) = \lim_{n\to\infty} \frac{I(\overline{X}_1^n; Z^\star(\overline{X}_1^n))}{n}$ since $\overline{X}$ is stationary. We will show that

$$I(\overline{X}; Z^\star(\overline{X})) \geq \frac{I(X_1^b; Z^\star(X_1^b))}{b} - \varepsilon. \tag{2}$$

Combined with Equation (1), this implies that

$$\mathsf{SCap}(Z^\star) \geq I(\overline{X}; Z^\star(\overline{X})) \geq \mathsf{ICap}(Z^\star) - 3\varepsilon.$$

Since ε was arbitrary, it follows that $\mathsf{SCap}(Z^\star) = \mathsf{ICap}(Z^\star)$, and so also $\mathsf{SCap}(Z) = \mathsf{ICap}(Z)$. This yields Theorem 6.

It remains to show Equation (2). We do this by a combination of two lemmas.

Lemma 7. *We have*

$$I(\overline{X}; Z^\star(\overline{X})) = \sum_{j=0}^{b-1} \frac{1}{b} I(\hat{X}^{[j]}; Z^\star(\hat{X}^{[j]})) = I(\hat{X}; Z^\star(\hat{X})),$$

and these limits exist.

Proof. First, we prove that

$$I(\hat{X}^{[j]}; Z^\star(\hat{X}^{[j]})) = I(\hat{X}; Z^\star(\hat{X}))$$

for all $j \in \{0, 1, \ldots, b-1\}$ (in particular, these quantities exist for all $0 \leq j < b$, since $\hat{X}$ is stationary). For the sake of exposition we focus on $j = 1$. The argument is analogous for other choices of j. First, note that by Item 4a with $\tau = b - 1$ there exists a sequence γ_m such that $\gamma_m = o(m)$ and $\sum_{m\in\mathbb{N}} \gamma_m/m^2$ converges and a random variable W such that $H(W) \leq \gamma_{tb}$ and $Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})$ is completely determined by $Z^\star(\hat{X}_1^{[1]tb-1}), W$. Furthermore, by Item 3, $Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})$ completely determine $Z^\star(\hat{X}_1^{[1]tb-1})$. This means that

$$I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]tb-1})) \leq I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) \tag{3}$$

and

$$\begin{aligned} I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]tb-1})) &\geq I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) - H(W) \\ &\geq I(\hat{X}_1^{n+m}; Z^\star(\hat{X}_1^n), Z^\star(\hat{X}_{n+1}^{n+m})) - \gamma_{tb}. \end{aligned} \tag{4}$$

Therefore,

$$\begin{aligned}
I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]tb-1})) &\geq I(\hat{X}_1^{[1]b-1}, \hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) - \gamma_{tb} \\
&= I(\hat{X}_1^{[1]b-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) \\
&+ I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})|\hat{X}_1^{[1]b-1}) - \gamma_{tb} \\
&\geq I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})|\hat{X}_1^{[1]b-1}) - \gamma_{tb} \\
&= I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_b^{[1]tb-1})) - \gamma_{tb} \\
&= I(\hat{X}_{b+1}^{tb}; Z^\star(\hat{X}_{b+1}^{tb})) - \gamma_{tb} \\
&= I(\hat{X}_1^{(t-1)b}; Z^\star(\hat{X}_1^{(t-1)b})) - \gamma_{tb}.
\end{aligned}$$

The first inequality uses Equation (4). The second and fourth equalities use the fact that $\hat{X}$ is block-independent with blocklength b, and so $\hat{X}_b^{[1]tb-1}$ is independent of $\hat{X}_1^{[1]b-1}$ and is identically distributed to $\hat{X}_1^{(t-1)b}$. Similarly,

$$\begin{aligned}
I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]tb-1})) &\leq I(\hat{X}_1^{[1]b-1}, \hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) \\
&= I(\hat{X}_1^{[1]b-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})) \\
&+ I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})|\hat{X}_1^{[1]b-1}) \\
&\leq I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]b-1}), Z^\star(\hat{X}_b^{[1]tb-1})|\hat{X}_1^{[1]b-1}) + b\log|\Sigma_{\mathsf{in}}| \\
&= I(\hat{X}_b^{[1]tb-1}; Z^\star(\hat{X}_b^{[1]tb-1})) + b\log|\Sigma_{\mathsf{in}}| \\
&= I(\hat{X}_{b+1}^{tb}; Z^\star(\hat{X}_{b+1}^{tb})) + b\log|\Sigma_{\mathsf{in}}| \\
&= I(\hat{X}_1^{(t-1)b}; Z^\star(\hat{X}_1^{(t-1)b})) + b\log|\Sigma_{\mathsf{in}}|,
\end{aligned}$$

where the first inequality uses Equation (3) and the second inequality uses the fact that $H(\hat{X}_1^{[1]b-1}) \leq b\log|\Sigma_{\mathsf{in}}|$. As a result, we conclude that (below, we write $a \pm \delta$ for a real number in the interval $[a-\delta, a+\delta]$)

$$\begin{aligned}
I(\hat{X}^{[1]}; Z^\star(\hat{X}^{[1]})) &= \lim_{t\to\infty} \frac{I(\hat{X}_1^{[1]tb-1}; Z^\star(\hat{X}_1^{[1]tb-1}))}{tb-1} \\
&= \lim_{t\to\infty} \frac{I(\hat{X}_1^{(t-1)b}; Z^\star(\hat{X}_1^{(t-1)b})) \pm (b\log|\Sigma_{\mathsf{in}}| + \gamma_{tb})}{tb-1} \\
&= \lim_{t\to\infty} \frac{(t-1)b}{tb-1} \cdot \frac{I(\hat{X}_1^{(t-1)b}; Z^\star(\hat{X}_1^{(t-1)b})) \pm (b\log|\Sigma_{\mathsf{in}}| + \gamma_{tb})}{(t-1)b} \\
&= \lim_{t\to\infty} \frac{I(\hat{X}_1^{(t-1)b}; Z^\star(\hat{X}_1^{(t-1)b}))}{(t-1)b} \\
&= I(\hat{X}; Z(\hat{X})).
\end{aligned}$$

The fourth equality uses the fact that $|\Sigma_{\mathsf{in}}|$ is a finite constant and $\lim_{t\to\infty} \frac{\gamma_{tb}}{(t-1)b} = 0$, since $\gamma_{tb} = o(tb)$. Furthermore, Lemma 6 guarantees that $I(\hat{X}; Z(\hat{X}))$ exists.

It remains to see the leftmost inequality of the lemma statement. First, note that

$$I(\overline{X}; Z^\star(\overline{X})) = I(\overline{X}; Z^\star(\overline{X})|V) = \lim_{n\to\infty} \frac{I(\overline{X}_1^n; Z^\star(\overline{X}_1^n)|V)}{n}. \tag{5}$$

This holds since $H(V) \leq \log b$ and b is a fixed constant. Furthermore,

$$\begin{aligned} I(\overline{X}; Z^\star(\overline{X})|V) &= \lim_{n\to\infty} \frac{I(\overline{X}_1^n; Z^\star(\overline{X}_1^n)|V)}{n} \\ &= \lim_{n\to\infty} \frac{\frac{1}{b}\sum_{j=0}^{b-1} I(\overline{X}_1^n; Z^\star(\overline{X}_1^n)|V=j)}{n} \\ &= \lim_{n\to\infty} \frac{\frac{1}{b}\sum_{j=0}^{b-1} I(\hat{X}_1^{[j]n}; Z^\star(\hat{X}_1^{[j]n}))}{n}. \end{aligned} \tag{6}$$

As we saw above, the limits

$$I(\hat{X}^{[j]}; Z^\star(\hat{X}^{[j]})) = \lim_{n\to\infty} \frac{I(\hat{X}_1^{[j]n}; Z^\star(\hat{X}_1^{[j]n}))}{n}$$

exist and equal $I(\hat{X}; Z^\star(\hat{X}))$ for all j. Therefore, since the sum over j is finite, we can swap limit and sum and conclude from Equations (5) and (6) that

$$\begin{aligned} I(\overline{X}; Z^\star(\overline{X})) &= I(\overline{X}; Z^\star(\overline{X})|V) \\ &= \frac{1}{b}\sum_{j=0}^{b-1} \lim_{n\to\infty} \frac{I(\hat{X}_1^{[j]n}; Z^\star(\hat{X}_1^{[j]n}))}{n} \\ &= \frac{1}{b}\sum_{j=0}^{b-1} I(\hat{X}^{[j]}; Z^\star(\hat{X}^{[j]})) \\ &= I(\hat{X}; Z^\star(\hat{X})), \end{aligned}$$

as desired. □

Lemma 8. *Suppose that $\hat{X}$ is a block independent process with blocklength b. Then, we have*

$$\left| I(\hat{X}; Z^\star(\hat{X})) - \frac{I(\hat{X}_1^b; Z^\star(\hat{X}_1^b))}{b} \right| \leq \alpha_b/b,$$

where $\lim_{b\to\infty} \frac{\alpha_b}{b} = 0$.

Proof. By Item 4b, there exists a random variable W with $H(W) \leq t \cdot \alpha_b$ such that the sequences $Z^\star(\hat{X}_1^b), Z^\star(\hat{X}_{b+1}^{2b}), \ldots, Z^\star(\hat{X}_{(t-1)b+1}^{tb})$ are completely determined by $Z^\star(\hat{X}_1^{tb})$ and W. Therefore,

$$\begin{aligned} I(\hat{X}_1^{tb}; Z^\star(\hat{X}_1^{tb})) &\geq I(\hat{X}_1^{tb}; Z^\star(\hat{X}_1^b), Z^\star(\hat{X}_{b+1}^{2b}), \ldots, Z^\star(\hat{X}_{(t-1)b+1}^{tb})) - t \cdot \alpha_b \\ &= \sum_{i=1}^{t} I(\hat{X}_{(i-1)b+1}^{ib}; Z^\star(\hat{X}_{(i-1)b+1}^{ib})) - t \cdot \alpha_b \\ &= \sum_{i=1}^{t} I(\hat{X}_1^b; Z^\star(\hat{X}_1^b)) - t \cdot \alpha_b \\ &= t(I(\hat{X}_1^b; Z^\star(\hat{X}_1^b)) - \alpha_b). \end{aligned}$$

Consequently,

$$I(\hat{X}; Z^\star(\hat{X})) = \lim_{t\to\infty} \frac{I(\hat{X}_1^{tb}; Z^\star(\hat{X}_1^{tb}))}{tb} \geq \frac{I(\hat{X}_1^b; Z^\star(\hat{X}_1^b)) - \alpha_b}{b}.$$

On the other hand, by Item 3 we have

$$\frac{I(\hat{X}_1^b; Z^\star(\hat{X}_1^b))}{b} = \frac{I(\hat{X}_1^{tb}; Z^\star(\hat{X}_1^b), Z^\star(\hat{X}_{b+1}^{2b}), \ldots, Z^\star(\hat{X}_{(t-1)b+1}^{tb}))}{tb} \geq \frac{I(\hat{X}_1^{tb}; Z^\star(\hat{X}_1^{tb}))}{tb}$$

for all t and b, and so $\frac{I(X_1^b; Z^\star(X_1^b))}{b} \geq I(\hat{X}; Z^\star(\hat{X}))$. □

3.4 Information capacity of admissible channels is achieved by Markov process

We use Theorem 6 to show the following.

Theorem 7. *Suppose that the channel Z is admissible. Then,*

$$\mathsf{ICap}(Z) = \lim_{m\to\infty} \mathsf{SCap}^{(m)}(Z).$$

Proof. Given $\varepsilon > 0$, let X be a stationary ergodic input process and b a large enough integer such that

$$\frac{I(X_1^b; Z^\star(X_1^b))}{b} \geq \mathsf{ICap}(Z) - \varepsilon.$$

Such a process X and integer b are guaranteed to exist by Item 2 and Theorem 6. Let $\hat{X}$ be the stationary $(b-1)$-th order Markov process satisfying

$$p_{\hat{X}_1^b}(x_1^b) = p_{X_1^b}(x_1^b).$$

We will show that

$$I(\hat{X}; Z(\hat{X})) = I(\hat{X}; Z^\star(\hat{X})) \geq \frac{I(X_1^b; Z^\star(X_1^b))}{b} - \frac{\alpha_b}{b}, \tag{7}$$

with $\lim_{b\to\infty} \frac{\alpha_b}{b} = 0$, where the first equality holds by Item 2. Taking b to be large enough, we get that $I(\hat{X}; Z(\hat{X})) \geq \mathsf{ICap}(Z) - 2\varepsilon$, and, since ε was arbitrary, the theorem statement follows.

Since $\hat{X}$ is a Markov process, we have $H(\hat{X}) \geq H(X)$. Therefore, it is enough to prove that

$$H(\hat{X}|Z^\star(\hat{X})) \leq \frac{H(X_1^b|Z^\star(X_1^b))}{b} + \frac{\alpha_b}{b}. \tag{8}$$

We have

$$\begin{aligned}
H(\hat{X}|Z^\star(\hat{X})) &= \lim_{t\to\infty} \frac{H(\hat{X}_1^{tb}|Z^\star(\hat{X}_1^{tb}))}{tb} \\
&\leq \lim_{t\to\infty} \frac{H(\hat{X}_1^{tb}|Z^\star(\hat{X}_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})) + t\cdot\alpha_b}{tb} \\
&= \lim_{t\to\infty} \frac{\sum_{i=1}^t H(\hat{X}_{(i-1)b+1}^{ib}|\hat{X}_1^{(i-1)b}, Z^\star(\hat{X}_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})) + t\cdot\alpha_b}{tb} \\
&\leq \lim_{t\to\infty} \frac{\sum_{i=1}^t H(\hat{X}_{(i-1)b+1}^{ib}|Z^\star(\hat{X}_{(i-1)b+1}^{ib})) + t\cdot\alpha_b}{tb} \\
&= \lim_{t\to\infty} \frac{\sum_{i=1}^t H(\hat{X}_1^b|Z^\star(\hat{X}_1^b)) + t\cdot\alpha_b}{tb} \\
&= \frac{H(X_1^b|Z^\star(X_1^b))}{b} + \frac{\alpha_b}{b},
\end{aligned}$$

as desired. The first inequality uses Item 4b. The second equality uses the chain rule for conditional entropy. The second inequality holds since further conditioning does not increase entropy. The third equality uses the stationarity of X. □

3.5 Coding capacity equals information capacity, and existence of dense codes from stationary ergodic processes

In this section, we begin by establishing suitable convergence of the information density of block-independent processes to their information rate for admissible channels. We use this result in two ways. First, we use it to show that $\mathsf{CCap}(Z) = \mathsf{ICap}(Z)$. Second, focusing on admissible channels with binary input for simplicity, we apply this result to stationary ergodic processes (which by Theorem 6 achieve information capacity on admissible channels) to conclude that there exist "dense" codes $\mathcal{C}$ that achieve capacity on Z, where "dense" means that every short substring of $c \in \mathcal{C}$ contains a decent fraction of 1s. As already pointed out in [PLW22], this property is relevant for the construction of efficient capacity-achieving codes for these channels.

3.5.1 Convergence of information density for block-independent process

For two random variables X, Y, we define their *information density* $i_{X,Y}$ as

$$i_{X,Y}(x,y) = \log\left(\frac{p_{XY}(x,y)}{p_X(x) \cdot p_Y(y)}\right).$$

Note that $\mathbb{E}_{(x,y)\sim p_{X,Y}}[i_{X,Y}(x,y)] = I(X;Y)$. We show the following.

Theorem 8. *Let Z be an admissible channel with respect to $Z^\star$. Fix $\varepsilon > 0$ and let $(\alpha_n)_{n\in\mathbb{N}}$ and $(\beta_n)_{n\in\mathbb{N}}$ be the sequences guaranteed by Item 4b and Item 5. Let X be a block-independent process with blocklength b such that*

$$\max(\alpha_b/b, \beta_b/b) \le \varepsilon^2/3.$$

Then, there exists a constant t_0 (possibly depending on ε, b, and ε) such that for all $t \ge t_0$ we have

$$\Pr_{(x,z)\sim X_1^{tb}, Z(X_1^{tb})}\left[\left|\frac{i_{X_1^{tb},Z(X_1^{tb})}(x,z)}{tb} - I(X;Z(X))\right| \le \varepsilon\right] \ge 1-\varepsilon.$$

Before we prove Theorem 8 we establish some useful lemmas. We will start by working with the $Z^\star$ channel.

Lemma 9. *Suppose that X is a block-independent process with blocklength b. Let*

$$\eta_t^\star = (Z^\star(X_1^b), Z^\star(X_{b+1}^{2b}), \dots, Z^\star(X_{(t-1)b+1}^{tb})).$$

Then,

$$i_{X_1^{tb},\eta_t^\star}(x,z) = \sum_{i=1}^{t} i_{X_1^b,Z^\star(X_1^b)}(x^{(i)}, z^{(i)}), \tag{9}$$

where $x^{(i)}, z^{(i)}$ denote the i-th blocks of x and z, respectively. In particular, $\frac{1}{tb} i_{X_1^{tb},\eta_t^\star}(X_1^{tb}, \eta_t^\star)$ converges almost surely to $\frac{I(X_1^b;Z^\star(X_1^b))}{b}$ as $t \to \infty$.

Proof. Equation (9) follows from the fact that $p_{X_1^{tb},\eta_{t,b}^\star}(x,z) = \prod_{i=1}^t p_{X_1^b,Z^\star(X_1^b)}(x^{(i)}, z^{(i)})$ by block-independence of X (with blocklength b). The statement about convergence then follows from the strong law of large numbers, since the random variables $i_{X_1^b,Z^\star(X_1^b)}(X_{(i-1)b+1}^{ib}, Z^\star(X_{(i-1)b+1}^{ib}))$ are i.i.d. for all $i \in [t]$ and their expectation is $I(X_1^b; Z^\star(X_1^b))$. □

For a given input X_1^{tb}, we will couple the $\eta_t^\star$ and $Z^\star(X_1^{tb})$ processes with the help of Item 5. Let Y_t be the random variable such that $\eta_t^\star \to Y_t$ and ϕ the deterministic function such that $Z^\star(X_1^{tb}) = \phi(\eta_t^\star, Y_t)$ and $\max_z \log|\phi^{-1}(z)| \le t \cdot \beta_b$, guaranteed by Item 5.

Lemma 10. *We have*

$$\begin{aligned}
&\mathbb{E}_{(x,z,y)\sim X_1^{tb},\eta_t^\star,Y_t}\left[\left|i_{X_1^{tb},\eta_t^\star}(x,z) - i_{X_1^{tb},Z^\star(X_1^{tb})}(x,\phi(z,y))\right|\right] \\
&= \mathbb{E}_{(x,z,y)\sim X_1^{tb},\eta_t^\star,Y_t}\left[\left|i_{X_1^{tb},\eta_t^\star,Y_t}(x,z,y) - i_{X_1^{tb},Z^\star(X_1^{tb})}(x,\phi(z,y))\right|\right] \\
&\le \max_z \log|\phi^{-1}(z)| \\
&\le t \cdot \beta_b.
\end{aligned}$$

The proof of this lemma relies on the following simple but useful fact about information densities, due to Dobrushin [Dob67].

Lemma 11 ([Dob67])**.** *Let $\phi : \mathcal{A} \to \mathcal{B}$ be a deterministic function with $\mathcal{A}$ and $\mathcal{B}$ finite sets, and write $M_\phi = \max_{z\in\mathcal{B}}|\phi^{-1}(z)|$ for the size of the largest preimage of ϕ. Let A be supported on $\mathcal{A}$, and define $B = \phi(A)$. Then,*

$$\mathbb{E}_{(x,a)\sim X,A}[|i_{X,A}(x,a) - i_{X,B}(x,\phi(a))|] \le \log M_\phi.$$

We are now ready to proceed with the proof of Lemma 10.

Proof of Lemma 10. The first equality follows from the fact that

$$i_{X_1^{tb},\eta_t^\star,Y_t}(x,z,y) = i_{X_1^{tb},\eta_t^\star}(x,z)$$

for all (x,z,y) in the support of $(X_1^{tb},\eta_t^\star,Y_t)$, since $X_1^{tb} \to \eta_t^\star \to \eta_t^\star, Y_t$. The inequality follows from Lemma 11 with $X = X_1^{tb}$ and $A = (\eta_t^\star, Y_t)$. The last equality follows from the hypothesis on ϕ, guaranteed by Item 5. □

We are now ready to prove Theorem 8.

Proof of Theorem 8. Since X is block-independent with blocklength b, by Lemma 8 we have

$$\left|\frac{I(X_1^b; Z^\star(X_1^b))}{b} - I(X; Z^\star(X))\right| \le \frac{\alpha_b}{b} \le \varepsilon/3, \tag{10}$$

where the last inequality uses the hypothesis on b from the theorem statement. Therefore, it is enough to show that

$$\Pr_{(x,z)\sim X_1^{tb},Z^\star(X_1^{tb})}\left[\left|\frac{i_{X_1^{tb},Z^\star(X_1^{tb})}(x,z)}{tb} - \frac{I(X_1^b; Z^\star(X_1^b))}{b}\right| \le 2\varepsilon/3\right] \ge 1-\varepsilon. \tag{11}$$

First, by the triangle inequality, we have

$$\begin{aligned}
&\Pr_{(x,z)\sim X_1^{tb},Z^\star(X_1^{tb})}\left[\left|\frac{i_{X_1^{tb},Z^\star(X_1^{tb})}(x,z)}{tb} - \frac{I(X_1^b; Z^\star(X_1^b))}{b}\right| \le 2\varepsilon/3\right] \\
&\le \Pr_{(x,z^\star,y)\sim X_1^{tb},\eta_t^\star,Y_t}\left[\left|\frac{i_{X_1^{tb},\eta_t^\star}(x,z^\star)}{tb} - \frac{i_{X_1^{tb},Z^\star(X_1^{tb})}(x,\phi(z^\star,y))}{tb}\right| \le \varepsilon/3\right]
\end{aligned}$$

$$+ \Pr_{(x,z^\star)\sim X_1^{tb},\eta_t^\star}\left[\left|\frac{i_{X_1^{tb},\eta_t^\star}(x,z^\star)}{tb} - \frac{I(X_1^b;Z^\star(X_1^b))}{b}\right| \le \varepsilon/3\right]. \tag{12}$$

We analyze the two terms in the sum separately. For the first term, combining Lemma 10 with Markov's inequality yields

$$\begin{aligned}\Pr_{(x,z^\star,w)\sim X_1^{tb},\eta_t^\star,W_t}\left[\left|\frac{i_{X_1^{tb},\eta^\star}(x,z^\star)}{tb} - \frac{i_{X_1^{tb},Z^\star(X_1^{tb})}(x,\phi(z^\star,w))}{tb}\right| > \varepsilon/3\right] &\le \frac{\max_z \log|\phi^{-1}(z)|}{\varepsilon tb}\\ &\le \frac{\beta_b}{\varepsilon b}\\ &\le \varepsilon/2, \end{aligned}\tag{13}$$

where the last inequality holds by the hypothesis on b from the theorem statement. For the second term, by Lemma 9, for all $t \ge t_0$ with t_0 a sufficiently large constant depending on ε, b, and X, we have

$$\Pr_{(x,z^\star)\sim X_1^{tb},\eta_t^\star}\left[\left|\frac{i_{X_1^{tb},\eta_t^\star}(x,z^\star)}{tb} - \frac{I(X_1^b;Z^\star(X_1^b))}{b}\right| > \varepsilon/3\right] \le \varepsilon/2. \tag{14}$$

Combining Equation (12) with Equations (13) and (14) yields Equation (11), as desired. □

3.5.2 Capacity-achieving codes for admissible channels

Theorem 8 implies, via standard methods, that the coding capacity and information capacity of admissible channels coincide. For completeness, we discuss this in detail. Later in Section 3.5.3 we combine Theorems 6 and 8 to show the existence of "dense" capacity-achieving codes suitable for bootstrapping efficient constructions.

We begin by relying on the following well-known theorem that formalizes the guarantees of MAP decoding.

Lemma 12 ([PW24, Theorem 18.5], adapted)**.** *Fix an input random variable X supported on Σ_{in}^n and a channel Z with input alphabet Σ_{in}. Then, for any $\tau > 0$ there exists a code $\mathcal{C}$ with blocklength n, size M, and average decoding error probability ε satisfying*

$$\varepsilon \le \Pr_{(x,z)\sim X,Z(X)}[i_{X,Z(X)}(x,z) \le \log M + \tau] + 2^{-\tau}.$$

The code guaranteed by Lemma 12 is obtained by sampling M codewords i.i.d. according to X. We briefly discuss how Lemma 12 can be combined with Theorem 8 to obtain codes with arbitrarily small decoding error probability and the desired rate for an arbitrary admissible channel.

Corollary 1. *Let Z be an admissible channel. Fix an arbitrary $\varepsilon > 0$. Then, there exists a constant $b(\varepsilon)$ such that for any $b \ge b(\varepsilon)$ and any block-independent input process X with blocklength b there exists a constant $t(\varepsilon, b, X)$ such that for any $t \ge t(\varepsilon, b, X)$ there exists an $(n = tb, R, \varepsilon)$-code for Z with $R \ge I(X;Z(X)) - \varepsilon$.*

Proof. Fix $\varepsilon > 0$ and let $b(\varepsilon), t(\varepsilon)$ be the constants guaranteed by Theorem 8. Consider any block-independent input process X with blocklength $b \ge b(\varepsilon)$ and number of blocks $t \ge t(\varepsilon)$. Set $n = tb$, $M = 2^{n(I(X;Z(X))-3\varepsilon)}$, and $\tau = \varepsilon n$. By Lemma 12, there exists an $(n = tb, R, \lambda)$-code for Z with

$$\lambda \le \Pr_{(x,z)\sim X_1^n,Z(X_1^n)}[i_{X_1^n,Z(X_1^n)}(x,z) \le \log M + \tau] + 2^{-\tau}$$

$$= \Pr_{(x,z)\sim X_1^n, Z(X_1^n)}\left[\frac{i_{X_1^n, Z(X_1^n)}(x,z)}{n} \le I(X;Z(X)) - 2\varepsilon\right] + 2^{-\varepsilon n}$$
$$\le \varepsilon + 2^{-\varepsilon n},$$

where the last inequality follows from Theorem 8. Now, we may set $b(\varepsilon)$ and $t(\varepsilon)$ large enough as a function of ε so that $2^{-\varepsilon n} \le \varepsilon$, and so $\lambda \le 2\varepsilon$. □

Extension to all blocklengths. We now argue how Corollary 1 can be extended to all blocklengths, which shows that for any admissible channel Z the coding capacity equals the information capacity.

Theorem 9. *Suppose that Z is an admissible channel. Then, $\mathsf{ICap}(Z) = \mathsf{CCap}(Z)$.*

Proof. We begin by considering the following "trimming" version of Z, which we denote by Z'. On input x, this channel first sends x through Z to obtain output $Z(x)$, and then trims the maximal substring of 0s at the end of $Z(x)$. Since Z' is a degraded version of Z, for any input process X we have $I(X;Z'(X)) \le I(X;Z(X))$. Also, $Z(X_1^n)$ is completely determined by $Z'(X_1^n)$ and the length L of the run of 0s trimmed from the end of $Z(X_1^n)$ to obtain $Z'(X_1^n)$. Since $H(L) \le \log n$, this means that $I(X_1^n; Z'(X_1^n)) \ge I(X_1^n; Z(X_1^n)) - \frac{\log n}{n} \to I(X;Z(X))$ as $n \to \infty$. We conclude that Z' is also admissible with respect to $Z^\star$ and $\mathsf{ICap}(Z') = \mathsf{ICap}(Z)$.

Fix an arbitrary $\varepsilon > 0$ and a block-independent process X with block length $b = b(\varepsilon)$ such that $I(X;Z'(X)) \ge \mathsf{ICap}(Z') - \varepsilon$ (the existence of such a process follows, e.g., from the discussion after Equation (1)). By Corollary 1 applied to Z' and X, there exists a family of codes $\{\mathcal{C}_{tb}\}_{t\in\mathbb{N}}$ such that $\mathcal{C}_{tb}$ is a $(tb, R_{tb}, \varepsilon_{tb})$-code with $R_{tb} \ge I(X;Z'(X)) - \varepsilon$ and $\varepsilon_{tb} \le \varepsilon$ for all large enough $t \ge t(\varepsilon, X)$. We extend this family to all blocklengths n as follows. For $n < t(\varepsilon)b(\varepsilon)$, we set $\mathcal{C}_n = \Sigma_{\mathsf{in}}^n$. For $tb < n < (t+1)b$ with $t \ge t(\varepsilon, X)$, we construct an (n, R_n, ε_n)-code $\mathcal{C}_n$ by appending a run of $n - tb < b$ 0s to codewords of $\mathcal{C}_{tb}$. First, note that since Z' trims all 0s at the end of a codeword, the decoding error probability of $\mathcal{C}_n$ equals that of $\mathcal{C}_{tb}$, and so $\varepsilon_n = \varepsilon_{tb} \le \varepsilon$ for all large enough n depending only on ε. Second, $R_n = R_{tb} \cdot \frac{tb}{(t+1)b} \ge R_{tb} - \varepsilon$ for all large enough n (again, depending on ε and X).

Now, for each $k \in \mathbb{N}$ define $\varepsilon^{(k)} = 1/k$ and the associated family $\{\mathcal{C}_n^{(k)}\}_{n\in\mathbb{N}}$ of $(n, R_n^{(k)}, \varepsilon_n^{(k)})$-codes guaranteed by the last paragraph, where $R_n^{(k)} \ge \mathsf{ICap}(Z') - 2\varepsilon^{(k)}$ and $\varepsilon_n^{(k)} \le \varepsilon^{(k)}$ for all large enough $n \ge n(k)$ (note that the choice of input process X is fixed for each $k \in \mathbb{N}$, so $n(k)$ really only depends on k). Taking codes from the k-th family for every blocklength n between $n(k)$ and $n(k+1)$, we get a family of codes $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ where each $\mathcal{C}_n$ is an (n, R_n, ε_n)-code for Z' with $R_n \to I(X;Z'(X))$ and $\varepsilon_n \to 0$ as $n \to \infty$. Furthermore, since Z' is a degraded version of Z, we also have that each $\mathcal{C}_n$ is an (n, R_n, ε_n)-code for Z. Since $\mathsf{ICap}(Z) = \mathsf{ICap}(Z')$, we conclude that $\mathsf{CCap}(Z) = \mathsf{CCap}(Z') = \mathsf{ICap}(Z') = \mathsf{ICap}(Z)$. □

3.5.3 Dense capacity-achieving codes for admissible channels

In the previous section we showed that the coding capacity and information capacity of an admissible channel are the same. However, this alone is not sufficient if we wish to obtain *efficiently encodable and decodable* capacity-achieving codes for an admissible channel. In this section, following the approach of [PLW22], we combine the fact that capacity is achieved by stationary ergodic processes (Theorem 6) with Theorem 8 to show the existence of capacity-achieving codes with density properties for admissible channels as stated in Theorem 5, useful for constructing efficient

capacity-achieving codes.[2] For simplicity we will focus on admissible channels with binary input alphabet, although our discussion generalizes further. We restate Theorem 5 here for convenience.

Theorem 5 (Dense capacity-achieving codes for admissible channels)**.** *Let Z be an admissible channel with binary input alphabet. Then, for any $\varepsilon, \zeta > 0$ there exist $\gamma \in (0, 1/2)$ and integers $b = b(\varepsilon, \zeta)$ and $t(\varepsilon, \zeta)$ depending only on ε and ζ such that for all $t \geq t(\varepsilon, \zeta)$ there exists a code $\mathcal{C}$ with blocklength $n = t \cdot b$, rate $R \geq \mathsf{ICap}(Z) - \varepsilon$, and maximal decoding error probability ε over Z such that for all codewords $c \in \mathcal{C}$ we have $\gamma\zeta n \leq w(c_i^{i+\zeta n}) \leq (1-\gamma)\zeta n$ for all $i \in [(1-\zeta)n]$, where $w(\cdot)$ denotes the Hamming weight.*

Our proof of Theorem 5 will rely on the following lemma proved in [PLW22].

Lemma 13 ([PLW22, Proposition 3.4], adapted)**.** *Let X be a stationary ergodic process supported on $\{0,1\}$ with $\Pr[X_1 = 1] \in (0,1)$. Then, for any $\zeta > 0$ there exists $\gamma \in (0, 1/2)$ and an integer $n_0 > 0$ such that the following holds for all $n \geq n_0$. With probability at least 0.99 over the sampling of $x \sim X_1^n$, we have $\gamma\zeta n \leq w(x_i^{i+\zeta n}) \leq (1-\gamma)\zeta n$ for all $i \in [(1-\zeta)n]$, where $w(\cdot)$ denotes the Hamming weight.*

Proof of Theorem 5. Let Z be an arbitrary admissible channel with binary input alphabet. Let X be a stationary ergodic process such that $I(X; Z(X)) > 0$. Then, it must be the case that $\Pr[X_1 = 1] \in (0,1)$, and so Lemma 13 applies to X with some constants $\zeta > 0$, $\gamma \in (0, 1/2)$, and n_0.

It will be slightly easier to work with a block-independent process. Fix $\varepsilon > 0$. From the proof of Theorem 6 in Section 3.3, we know that there is a stationary ergodic process $\overline{X}$ such that $I(\overline{X}; Z(\overline{X})) \geq \mathsf{ICap}(Z) - \varepsilon$, and moreover $\overline{X}$ is created by choosing an appropriate block-independent process $\hat{X}$ with blocklength $b(\varepsilon)$, then choosing a uniformly random starting point in the first block of $\hat{X}$, and starting $\hat{X}$ from that point. Since $\overline{X}$ is obtained from $\hat{X}$ by trimming at most b bits from the beginning of $\hat{X}$, we conclude that $\hat{X}$ also satisfies the properties laid out in Lemma 13 with possibly a slightly smaller γ and slightly larger n_0. Recalling that the code $\mathcal{C}$ guaranteed by Corollary 1 applied to Z and $\hat{X}$ is obtained by sampling codewords i.i.d. according to $\hat{X}_1^n$ yields Theorem 5. This is because with high probability more than a 0.9-fraction of codewords $c \in \mathcal{C}$ will satisfy $\gamma\zeta n \leq w(c_i^{i+\zeta n}) \leq (1-\gamma)\zeta n$ for all $i \in [(1-\zeta)n]$, and throwing away all codewords of $\mathcal{C}$ that do not satisfy this property will not affect the asymptotic rate. □

4 Some special cases of our capacity theorems

4.1 The Mao-Diggavi-Kannan ISI model

Consider the *ℓ-ISI-synchronization channel* Z from [MDK18], for an arbitrary fixed integer $\ell \geq 0$. This channel replaces the i-th input bit x_i by a string $y_i \in \{0,1\}^*$ with probability

$$p(y_i | x_i, x_{i-1}, \ldots, x_{i-\ell}).$$

For simplicity, we focus on the case where $p(\cdot | x_i, x_{i-1}, \ldots, x_{i-\ell})$ is supported on $\{0,1\}^{\leq a} = \bigcup_{j=0}^{a} \{0,1\}^j$ for some integer $a \geq 1$ and any choice of $x_i, x_{i-1}, \ldots, x_{i-\ell}$, although our argument below generalizes further. We show that our capacity theorems apply to this channel, and so they generalize the corresponding results of [MDK18].

[2] Pernice, Li, and Wootters [PLW22] focused on channels with i.i.d. deletions and replications. The analog of Theorem 6 for these channels was already shown in [Dob67].

Consider the special channel $Z^\star$ that behaves like Z, except that it does not corrupt the first ℓ input bits and separately outputs the last ℓ input bits. We show that Z is admissible with respect to $Z^\star$.

- **Item 1:** Since the output associated to the i-th input bit has length at most a fixed constant a, we have $H(Z^\star(X_1^n)) \le n \cdot (a+1)$.
- **Item 2:** Fix an arbitrary input process X. Note that

$$X_1^n \to Z^\star(X_1^n) \to Z(X_1^n).$$

  Let W denote, for each $j \in [\ell]$, the string v_j that X_j was replaced by in $Z(X_1^n)$. Then,

$$X_1^n \to Z(X_1^n), W, X_{n-\ell+1}^n \to Z^\star(X_1^n).$$

  Note that there are at most 2^{a+1} choices for each v_j. Therefore, $H(W, X_{n-\ell+1}^n) \le \ell + \ell \cdot (a+1)$, and so

$$I(X_1^n; Z^\star(X_1^n)) - \ell(a+2) \le I(X_1^n; Z(X_1^n)) \le I(X_1^n; Z^\star(X_1^n)).$$

  As a result,

$$\lim_{n\to\infty} \frac{|I(X_1^n; Z(X_1^n)) - I(X_1^n; Z^\star(X_1^n))|}{n} \le \lim_{n\to\infty} \frac{\ell(a+2)}{n} = 0.$$

- **Item 3:** Note that $Z^\star(X_1^m)$ does not corrupt the first ℓ bits of X_1^m, and furthermore it outputs $X_{m-\ell+1}^m$. Otherwise, it behaves exactly like Z. Analogously, $Z^\star(X_{m+1}^n)$ does not corrupt the first ℓ bits of X_{m+1}^n and it outputs $X_{n-\ell+1}^n$. Therefore, from $Z^\star(X_1^m), Z^\star(X_{m+1}^n)$ we have the necessary information to apply the correct errors to the first ℓ bits X_{m+1}^n, and we also know the last ℓ input bits $X_{n-\ell+1}^n$. This means that we fully determine $Z^\star(X_1^n)$.
- **Item 4:**
  1. Fix integers τ and $n \ge \tau$, and an input process X. Let N_1 denote the number of output bits corresponding to X_1^τ in $Z^\star(X_1^n)$. Also, let W_{pre} include for each $j \in \{\tau+1, \dots, \tau+\ell\}$ the string v_j that X_j was replaced by in $Z^\star(X_1^n)$. Then, $Z^\star(X_1^n), N, W_{\mathsf{pre}}, X_{\tau-\ell+1}^\tau, X_{\tau+1}^{\tau+\ell}$ completely determine $Z^\star(X_1^\tau), Z^\star(X_{\tau+1}^n)$, and

$$H(N, W_{\mathsf{pre}}, X_{\tau+1}^{\tau+\ell}, X_{n-\ell+1}^n) \le \log(\tau \cdot 2^{a+1}) + \ell \cdot (a+1) + 2\ell.$$

     Therefore, the prefix-partitioning half of Item 4a holds with $\gamma_m = \log\tau + (a+1) + \ell(a+3)$. An analogous argument establishes the suffix-partitioning property with the same γ_m.
  2. Fix a blocklength b, number of blocks t, and an input process X. For each $i \in [t]$, let N_i denote the number of output bits corresponding to $X_{(i-1)b+1}^{ib}$. Also, let W_i include, for each $j \in \{(i-1)b+1, (i-1)b+\ell\}$ the string v_j that X_j is replaced by. Let

$$W = (N_i, W_i, X_{ib-\ell+1}^{ib})_{i\in[t]}.$$

     Then, $Z^\star(X_1^{tb}), W$ completely determines the sequence $Z^\star(X_1^b), \dots, Z^\star(X_{(t-1)b+1}^{tb})$, and

$$H(W) \le \sum_{i=1}^{t} (\log(b \cdot 2^{a+1}) + \ell \cdot (a+1) + \ell) = t \cdot (\log(b \cdot 2^{a+1}) + \ell \cdot (a+1) + \ell).$$

     Therefore, Item 4b holds with $\alpha_m = \log(m \cdot 2^{a+1}) + \ell \cdot (a+1) + \ell = o(m)$.

- **Item 5:** Fix a blocklength b and a number of blocks t. Consider the random variable $Y = (Y_i)_{i\in[t]}$, where each Y_i includes, for each $j \in \{(i-1)b+1, \ldots, (i-1)b+\ell\}$, the string v_j that X_j should be replaced by in $Z^\star(X_1^n)$. Then $X_1^{tb} \to Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}) \to Y$, since the distribution of Y_i is completely determined by $X_{(i-1)b-\ell+1}^{(i-1)b}$ and $X_{(i-1)b+1}^{(i-1)b+\ell}$, where the former is revealed by the $(i-1)$-th output block $Z^\star(X_{(i-2)b+1}^{(i-1)b})$ and the latter is revealed by the i-th output block $Z^\star(X_{(i-1)b+1}^{ib})$. Moreover, we have $Z^\star(X_1^{tb}) = \phi(Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}), Y)$ for the deterministic function ϕ that discards the final t bits $X_{ib-\ell+1}^{ib}$ from each output block $Z^\star(X_{(i-1)b+1}^{ib})$ with $i < t$, applies the corruptions dictated by Y to $Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})$, and then concatenates these blocks.

  It remains to upper bound $\max_z \log|\phi^{-1}(z)|$ appropriately. First, note that there are at most $\binom{t(b2^{a+1}+1)}{t}$ ways of splitting z into t blocks of length at most $b2^{a+1}$ each. Second, for each block there are at most $2^{(a+1)\cdot\ell}$ choices for the first t input bits and the corresponding strings in $\{0,1\}^{\le a}$ that they were replaced by at the output, and there are at most 2^ℓ possibilities for the last t input bits. Putting these observations together implies that

$$\max_z |\phi^{-1}(z)| \le \binom{t(b2^{a+1}+1)}{t} \cdot 2^{(a+1)\cdot\ell} \cdot 2^\ell \le 2^{t(b2^{a+1}+1)h\left(\frac{1}{b2^{a+1}+1}\right)} \cdot 2^{(a+1)\ell},$$

  where we recall that h is the binary entropy function. Therefore,

$$\max_z \log|\phi^{-1}(z)| \le t(b2^{a+1}+1)h\left(\frac{1}{b2^{a+1}+1}\right) + (a+1)\ell,$$

  and so Item 5 holds with $\gamma_m = (m2^{a+1}+1)h\left(\frac{1}{m2^{a+1}+1}\right) + (a+1)\ell = o(m)$.

4.2 Multi-trace channels with input-correlated synchronization errors

We argue how our capacity theorems above apply to a wide class of *multi-trace* input-correlated synchronization channels. Fix an integer $T \ge 1$ (the number of traces), and consider the multi-trace channel Z given by

$$Z(X_1^n) = (Z_1(X_1^n), \ldots, Z_T(X_1^n)),$$

where the Z_i's are possibly distinct input-correlated synchronization channels, and each $Z_i(X_1^n)$ is an independent trace of X_1^n (i.e., the $Z_i(X_1^n)$'s are conditionally independent given X_1^n).

Theorem 10. *Suppose that each Z_i is admissible with respect to $Z_i^\star$, and that*

$$I(X; Z_1(X), \ldots, Z_T(X)) = I(X; Z_1^\star(X), \ldots, Z_T^\star(X)) \tag{15}$$

for all input processes X. Then, the T-trace channel Z is admissible with respect to the T-trace channel $Z^\star$ given by

$$Z^\star(X_1^n) = (Z_1^\star(X_1^n), \ldots, Z_T^\star(X_1^n)).$$

In particular, we have $\mathsf{ICap}(Z) = \mathsf{SCap}(Z) = \mathsf{CCap}(Z)$.

The assumption in Equation (15) appears stronger than simply requiring that each Z_i satisfy Item 2 with respect to $Z_i^\star$. Nevertheless, it still seems reasonable. Concretely, it is natural (as in all of our applications, for example) that $Z_i^\star(X_1^n)$ is completely determined by $Z_i(X_1^n)$ and some additional side information W_i satisfying $H(W_i) = o(n)$. In this case, we get that $Z^\star(X_1^n)$ is

completely determined by $Z(X_1^n)$ and the side information $W = (W_1, \dots, W_T)$, which satisfies $H(W) \le \sum_{i=1}^T H(W_i) = T \cdot o(n) = o(n)$, since the number of traces T is constant. This means that

$$\frac{|I(X_1^n; Z(X_1^n)) - I(X_1^n; Z^\star(X_1^n))|}{n} \le \frac{H(W)}{n} \to 0$$

as $n \to \infty$.

Proof of Theorem 10. We verify that Z is admissible with respect to $Z^\star$. Item 2 is already guaranteed by Equation (15), and so we focus on showing the other properties.

- **Item 1:** Fix an arbitrary input process X and integer $n \ge 1$. For each i we know that $H(Z_i^\star(X_1^n)) \le c_i n$ for some constant $c_i > 0$. Let $c^\star = \max_{i \in [T]} c_i$. Then, $H(Z^\star(X_1^n)) = H((Z_i^\star(X_1^n)_{i \in [T]}) \le \sum_{i=1}^T H(Z_i^\star(X_1^n)) \le T \cdot c^\star n$, and so Item 1 holds with constant $c = T \cdot c^\star$.
- **Item 3:** For an arbitrary input process X, indices $1 \le m \le n$ and each $i \in [T]$, we have that $X_1^n \to Z_i^\star(X_1^m), Z_i^\star(X_{m+1}^n) \to Z_i^\star(X_1^n)$. In particular, this means that

$$X_1^n \to (Z_i^\star(X_1^m))_{i \in [T]}, (Z_i^\star(X_{m+1}^n)_{i \in [T]} = Z^\star(X_1^m), Z^\star(X_{m+1}^n) \to (Z_i^\star(X_1^n))_{i \in [T]} = Z^\star(X_1^n).$$

- **Item 4:**
  - Item 4a: Fix any integers $\tau \ge 1$ and $n \ge \tau$. For each $i \in [T]$ let $\gamma_m^{(i)} = o(m)$ and $W_{\mathsf{pre},i}$ and $W_{\mathsf{suf},i}$ be the sequence and random variables guaranteed by Item 4a applied to Z_i. Consider $\gamma_m = \sum_{i=1}^T \gamma_m^{(i)}$ and $W_{\mathsf{pre}} = (W_{\mathsf{pre},i})_{i \in [T]}$ and $W_{\mathsf{suf}} = (W_{\mathsf{suf},i})_{i \in [T]}$. Then, $\gamma_m = o(m)$ and $\sum_{m \in \mathbb{N}} \gamma_m / m^2$ converges, and $H(W_{\mathsf{pre}}), H(W_{\mathsf{suf}}) \le \gamma_n$. Furthermore, $(Z^\star(X_1^n), W_{\mathsf{pre}}) = ((Z_i^\star(X_1^n))_{i \in [T]}, W_{\mathsf{pre}})$ completely determines $(Z^\star(X_1^\tau), Z^\star(X_{\tau+1}^n)) = (Z_i^\star(X_1^\tau), Z_i^\star(X_{\tau+1}^n))_{i \in [T]}$. The reasoning for W_{suf} is analogous.
  - Item 4b: For each $i \in [T]$ let $\alpha_m^{(i)} = o(m)$ and W_i be the sequence and random variable, respectively, guaranteed by Item 4b applied to Z_i. Consider $W = (W_1, \dots, W_T)$ and $\alpha_m = \sum_{i=1}^T \alpha_m^{(i)}$. Then, $\alpha_m = o(m)$ and $H(W) \le \sum_{i=1}^T H(W_i) \le t\alpha_b$, and $(Z^\star(X_1^{tb}), W) = ((Z_i^\star(X_1^{tb}), W_i))_{i \in [T]}$ completely determines

$$Z^\star(X_1^b), \dots, Z^\star(X_{(t-1)b+1}^{tb}) = (Z_i^\star(X_1^b), \dots, Z_i^\star(X_{(t-1)b+1}^{tb}))_{i \in [T]}\,.$$

- **Item 5:** For each $i \in [T]$, let $\beta_m^{(i)} = o(m)$, Y_i, and ϕ_i be the sequence, random variable, and deterministic function guaranteed by Item 5 applied to Z_i. Set $\beta_m = \sum_{i=1}^T \beta_m^{(i)} = o(m)$, $Y = (Y_i)_{i \in [T]}$, and

$$\phi(Z^\star(X_1^b), \dots, Z^\star(X_{(t-1)b+1}^{tb}), Y) = (\phi_i(Z_i^\star(X_1^b), \dots, Z_i^\star(X_{(t-1)b+1}^{tb}), Y_i))_{i \in [T]}.$$

  Then,

$$X_1^{tb} \to Z^\star(X_1^b), \dots, Z^\star(X_{(t-1)b+1}^{tb}) = (Z_i^\star(X_1^b), \dots, Z_i^\star(X_{(t-1)b+1}^{tb}))_{i \in [T]} \to (Y_i)_{i \in [T]} = Y$$

  and

$$\max_z \log |\phi^{-1}(z)| \le \max_z \log \left(\prod_{i=1}^T |\phi_i^{-1}(z_i)| \right) \le \sum_{i=1}^T \max_{z_i} \log |\phi_i^{-1}(z_i)| \le t \sum_{i=1}^T \beta_b^{(i)} = t\beta_b. \qquad \square$$

4.3 Capacity theorems for trimming synchronization channels

In this section, we argue that our capacity theorems are robust to additional "trimming" of channel outputs. This property is relevant for the marker-based construction of efficient capacity-achieving codes based on our capacity theorems.

For concreteness, let Z be an arbitrary channel with binary input and output alphabets. We consider the "0-trimming" version of Z, denoted by Z_0, which on input x first sends it through Z to obtain output $Z(x)$, and then trims the runs of 0s at the beginning and end of $Z(x)$.

Lemma 14. *Let Z be admissible with respect to $Z^\star$. Then, Z_0 is also admissible with respect to $Z^\star$. In particular, the capacities of Z and Z_0 are all equal.*

Proof. It suffices to show that $I(X; Z_0(X)) = I(X; Z(X))$ for any input process X. Fix an arbitrary integer $n > 0$. First, since $X_1^n \to Z(X_1^n) \to Z_0(X_1^n)$, we have that $I(X_1^n; Z_0(X_1^n)) \leq I(X_1^n; Z(X_1^n))$. On the other hand, if L_0 and L_1 denote the number of 0s trimmed by Z_0 from the beginning and end of $Z(X_1^n)$, we have that $X_1^n \to Z_0(X_1^n), L_0, L_1 \to Z(X_1^n)$, and $H(L_0, L_1) \leq 2\log(n+1)$. Therefore,

$$\lim_{n\to\infty} \frac{|I(X_1^n; Z_0(X_1^n)) - I(X_1^n; Z(X_1^n))|}{n} \leq \lim_{n\to\infty} \frac{2\log(n+1)}{n} = 0,$$

which implies the desired result. □

The simple proof of Lemma 14 can be easily extended to other trimming channels that trim different prefixes and suffixes. In particular, consider the "01-trimming" version of Z, denoted by Z_{01}, which on input x first sends it through Z to obtain output $Z(x)$, and then trims the run of 0s at the beginning of $Z(x)$ and the run of 1s at the end of $Z(x)$. A simple modification to the proof of Lemma 14 yields the following.

Lemma 15. *Let Z be admissible with respect to $Z^\star$. Then, Z_{01} is also admissible with respect to $Z^\star$. In particular, the capacities of Z and Z_{01} are all equal.*

4.4 Channels with runlength-dependent deletions

In this section, we apply our framework to binary channels Z with runlength-dependent deletions, in the sense that Z deletes each bit in a run of length ℓ independently with some probability $d(\ell)$ (and so we may see d as a function $d : \mathbb{N} \to [0, 1]$). Consider the special channel $Z^\star$ that on input x behaves exactly like Z except that it does not apply deletions to the first and last runs of x, and additionally reveals the lengths of these runs. We show that Z is admissible with respect to $Z^\star$.

- **Item 1:** Fix an arbitrary input process X and integer $n \geq 1$. Since $Z^\star$ only applies deletions, we have $H(Z^\star(X_1^n)) \leq n + 2\log n \leq 3n$.
- **Item 2:** Fix an arbitrary input process X. Note that

  $$Z^\star(X_1^n) \to Z(X_1^n).$$

  Let L_1, B_1 (resp. L_2, B_2) denote the number of bits deleted by Z from the first (resp. last) input run and the bit value of this run. Let also V denote whether the first and last input runs are distinct. Then,

  $$Z(X_1^n), L_1, B_1, L_2, B_2, V \to Z^\star(X_1^n).$$

Therefore, since $H(L_1, B_1, L_2, B_2, V) \le 1 + 2(\log(n+1)+1)$, we have

$$I(X_1^n; Z^\star(X_1^n)) - (1 + 2(\log(n+1)+1)) \le I(X_1^n; Z(X_1^n)) \le I(X_1^n; Z^\star(X_1^n)),$$

and so

$$\lim_{n\to\infty} \frac{|I(X_1^n; Z(X_1^n)) - I(X_1^n; Z^\star(X_1^n))|}{n} \le \lim_{n\to\infty} \frac{1 + 2(\log(n+1)+1)}{n} = 0.$$

This implies that $I(X; Z(X)) = I(X; Z^\star(X))$.

- **Item 3:** Fix arbitrary integers $1 \le m \le n$ and an input process X. Note that $Z^\star(X_1^m)$ does not apply deletions to the first and last runs of X_1^m and also reveals the lengths of these runs, and likewise for $Z^\star(X_{m+1}^n)$. To all other runs of X_1^m and X_{n+1}^m these channels apply the same deletion rate as $Z^\star(X_1^n)$, because these runs are not broken up by the partitioning of X_1^n into X_1^m and X_{m+1}^n. Furthermore, $Z^\star(X_1^m), Z^\star(X_{m+1}^n)$ reveal the lengths of the first and last runs of X_1^n and do not apply deletions to these runs. Therefore, it is enough to argue that knowing $Z^\star(X_1^m), Z^\star(X_{m+1}^n)$ allows us to apply the correct deletion rates to the last run of X_1^m and first run of X_{m+1}^n, which may actually be part of the same run of X_1^n.

  In the special case where the last run of X_1^m is also its first run (i.e., when $X_1^m = b^m$ for some $b \in \{0,1\}$) then we apply no deletions to it, nor to the first run of X_{m+1}^n in case it matches the bit value of X_1^m. In this case, the length of the first run of X_1^n, which is part of the output of $Z^\star(X_1^n)$ can be obtained from the lengths of the first runs of X_1^m and X_{m+1}^n, which are revealed by $Z^\star(X_1^m), Z^\star(X_{m+1}^n)$. An analogous argument holds for the special case where the last run of X_{m+1}^n is also its first run (i.e., when $X_{m+1}^n = b^{n-m}$ for some $b \in \{0,1\}$).

  In all other cases, since $Z^\star(X_1^m)$ reveals the length of the last run of X_1^m and $Z^\star(X_{m+1}^n)$ reveals the length of the first run of X_{m+1}^n, we know the length of the corresponding run(s) of X_1^n and so we also know the deletion rate that must be applied. Moreover, since no deletions were applied to the last run of X_1^m and the first run of X_{m+1}^n by $Z^\star(X_1^m)$ and $Z^\star(X_{m+1}^n)$, respectively, we can perfectly emulate the behavior of $Z^\star(X_1^n)$ on the corresponding runs.

- **Item 4:**

  1. Fix integers τ and $n \ge \tau$ and an input process X. Let N denote the number of bits deleted by $Z^\star$ from X_1^τ. Furthermore, let L_1, B_1 (resp. L_2, B_2) denote the length of the last run of X_1^τ (resp. first run of $X_{\tau+1}^n$) and the bit value of this run, respectively, and let N_1 (resp. N_2) denote the number of bit deleted from the last run of X_1^τ (resp. first run of $X_{\tau+1}^n$). Set $W_{\mathsf{pre}} = (N, L_1, B_1, N_1, L_2, B_2, N_2)$. Then, from $Z^\star(X_1^n), W_{\mathsf{pre}}$ we can exactly locate the output bits corresponding to the last run of X_1^τ and to the first run of $X_{\tau+1}^n$, and restore them to their original lengths. Therefore, $Z^\star(X_1^n), W_{\mathsf{pre}}$ completely determine $Z^\star(X_1^\tau), Z^\star(X_{\tau+1}^n)$, and $H(W_{\mathsf{pre}}) \le \log(\tau+1) + 4\log(n+1) + 2$, and so Item 4a holds with $\gamma_m = \log(\tau+1) + 4\log(m+1) + 2$, which satisfies the required properties.

     An analogous argument establishes the suffix-partitioning property with the same γ_m.

  2. Fix a blocklength b, number of blocks t, and an input process X. For each $i \in [t]$, let N_i denote the number of bits deleted from $X_{(i-1)b+1}^{ib}$ by $Z^\star(X_1^n)$, $L_{1,i}, N_{1,i}, B_{1,i}$ (resp. $L_{2,i}, N_{2,i}, B_{2,i}$) denote the length of the first (resp. last) run of $X_{(i-1)b+1}^{ib}$, the number of bits deleted from this run by $Z^\star(X_1^n)$, and the bit value of this run, respectively, and V_i denote whether the first and last runs of $X_{(i-1)b+1}^{ib}$ are distinct runs. Note that

     $$H(L_{1,i}, N_{1,i}, B_{1,i}, L_{2,i}, N_{2,i}, B_{2,i}, V_i) \le 4\log(b+1) + 3$$

for every $i \in [t]$. Let $W = (L_{1,i}, N_{1,i}, B_{1,i}, L_{2,i}, N_{2,i}, B_{2,i}, V_i)_{i\in[t]}$. Then, using a similar argument to previous items, we see that $Z^\star(X_1^{tb}), W$ completely determines the sequence $Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})$, and

$$H(W) \le \sum_{i=1}^{t} (4\log(b+1)+3) = t \cdot (4\log(b+1)+3).$$

Therefore, Item 4b holds with $\alpha_m = 4\log(m+1)+3 = o(m)$.

- **Item 5:** Fix a blocklength b and a number of blocks t. Consider the random variable $Y = (L_{1,i}, L_{2,i})_{i\in[t]}$ where $L_{1,i}, L_{2,i}$ denote the number of bits to be deleted from the first and last runs of $Z^\star(X_{(i-1)b+1}^{ib})$ so that we can transform $Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})$ into $Z^\star(X_1^{tb})$. Then $X_1^{tb} \to Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}) \to Y$, since the distribution of Y is completely determined by the lengths of the first and last runs of each block $Z^\star(X_{(i-1)b+1}^{ib})$. Moreover, we obtain $Z^\star(X_1^{tb}) = \phi(Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb}), Y)$ for the deterministic function ϕ that applies the deletions dictated by Y to the first and last runs of $Z^\star(X_1^b), \ldots, Z^\star(X_{(t-1)b+1}^{tb})$, concatenates these blocks, and uses the lengths of the first and last runs of each block $X_{(i-1)b+1}^{ib}$ revealed by $Z^\star(X_{(i-1)b+1}^{ib})$ to compute the lengths of the first and last runs of X_1^n. Together, these form $Z^\star(X_1^{tb})$.

  It remains to upper bound $\max_z \log|\phi^{-1}(z)|$ appropriately. First, note that there are at most $\binom{t(b+1)}{t}$ ways of splitting z into t blocks of length at most b each. Second, for each of the t blocks of length at most b, there are at most $(b+1)^2$ possibilities for the number of bits that were deleted from the first and last input runs of each block, $(b+1)^2$ possibilities for the lengths of these runs, 2^2 possibilities for the bit values of these runs, and 2 possibilities for whether these two runs are distinct runs or not. Putting these observations together implies that

$$\max_z |\phi^{-1}(z)| \le \binom{t(b+1)}{t} \cdot (b+1)^{4t} \cdot 2^{3t} \le 2^{t(b+1)h\left(\frac{1}{b+1}\right)} \cdot (b+1)^{2t} \cdot 2^{3t},$$

  where we have used the standard inequality $\binom{n}{k} \le 2^{nh(k/n)}$, with $h(p) = -p\log p - (1-p)\log(1-p)$ the binary entropy function. Therefore,

$$\begin{aligned}\max_z \log|\phi^{-1}(z)| &\le t(b+1)h\left(\frac{1}{b+1}\right) + 4t\log(b+1) + 3t \\ &= t\cdot\left((b+1)h\left(\frac{1}{b+1}\right) + 4\log(b+1) + 3\right),\end{aligned}$$

  and so Item 5 holds with $\beta_m = (m+1)h\left(\frac{1}{m+1}\right) + 4\log(m+1) + 3 = o(m)$.

We would like to apply Theorem 5 to the 0-*trimming multi-trace* version of any runlength-dependent deletion channel. Fix an arbitrary process $X = (X_i)_{i\in\mathbb{N}}$. For an arbitrary $n \in \mathbb{N}$, set

$$W = (L_1, L_2, D_1, D_2, B_1, B_2)$$

with (L_1, D_1, B_1) (resp. (L_2, D_2, B_2)) denoting, respectively, the length of the first (resp. last) run of X_1^n, the number of bits deleted from that run by $Z(X_1^n)$, and the bit value of that run. Then, we have

$$X_1^n \to Z(X_1^n), W \to Z^\star(X_1^n).$$

Furthermore, $H(W) \leq 4\log(n+1)+2 = o(n)$. This means that Equation (15), and thus Theorem 10, applies to $Z_1 = \cdots = Z_T = Z$ for any number of traces T. Combined with Lemma 15, we get that Theorem 5 applies to the trimming T-trace version of any runlength-dependent deletion channel. Formally, we get the following corollary.

Corollary 2. *Let Z be a 01-trimming T-trace runlength-dependent deletion channel. Then, for any $\varepsilon, \zeta > 0$ there exist $\gamma = \gamma(\varepsilon, \zeta) \in (0, 1/2)$ and integers $b = b(\varepsilon, \zeta)$ and $t(\varepsilon, \zeta)$ that depend only on ε and ζ such that for any $t \geq t(\varepsilon, \zeta)$ there exists a code $\mathcal{C}$ with blocklength $n = tb$, rate $R \geq \mathsf{ICap}(Z) - \varepsilon$, decoding error probability at most ε such that for all codewords $c \in \mathcal{C}$ we have $\gamma\zeta n \leq w(c_i^{i+\zeta n}) \leq (1-\gamma)\zeta n$ for all $i \in [(1-\zeta)n]$.*

Remark 3. Clearly, Corollary 2 also holds for the 10, 11, 00-trimming T-trace runlength-dependent deletion channel.

5 Warmup: efficient capacity-achieving codes for channels with runlength-dependent deletions

In Corollary 2, for the case of $T = 1$, we showed that there is a code that achieves capacity on the 00-trimming runlength-dependent channel. [3] Furthermore, we showed that this code has a large density of 1s in every "not too short" interval.

In this section, we will closely follow the arguments of Pernice, Li, and Wootters [PLW22] who showed how one can transform any code for the binary deletion channel (and, in fact, more i.i.d. synchronization channels) into an explicit and efficient code with a negligible loss in the rate. With some needed modifications, we will show how to transform (non-explicit and non-efficient) capacity-achieving codes for a smaller class of runlength-dependent channels into explicit and efficient capacity achieving codes.

We start by defining the class of runlength-dependent channels for which we aim to construct efficient codes.

Definition 11. *Let $M \in \mathbb{N}$ and $\mu \in (0, 1)$. A runlength-dependent deletion channel with deletion probability function $d : \mathbb{N} \to [0, 1]$ is called* BDC-RL-Bounded(d, μ, M) *if d is non-decreasing and for all $\ell \geq M$, we have $d(\ell) = d(M) < 1 - \mu$.*[4]

The theorem we will prove in this section is as follows.

Theorem 11. *Let $\varepsilon > 0$. There exists an explicit family of binary codes $\{C_i\}_{i=1}^{\infty}$ for the channel* BDC-RL-Bounded(d, μ, M) *where the block length of C_i goes to infinity as $i \to \infty$ and*[5]

1. *C_i is encodable in linear time and decodable in quasi-linear time.*
2. *The decoding failure probability is $\exp(-\Omega(n_i))$ where n_i is the block length of C_i.*
3. *The rate of the C_i is $R >$* Cap(BDC-RL-Bounded$(d, \mu, M)) - \varepsilon$.

[3] Recall that this means that the after the deletions that are performed by the channel, the channel also trims the first and last run of zeros.

[4] The monotonicity assumption is for simplicity and also since it makes sense to assume that longer runs are more likely to suffer from higher deletion rate.

[5] The channel parameters are fixed and do not depend on the block lengths of the codes.

Before formally presenting our construction, we briefly recall, in an informal way, the construction of [PLW22]. As in [GL19, CS22], the construction of [PLW22] is based on code concatenation where the outer code, C_{out}, is taken to be the code of [HS21b]. The inner code of [PLW22], denoted as C_{in}, is taken to be a capacity-achieving code over the trimming binary deletion channel that is "dense". More precisely, our Corollary 2 for the single trace setting can be seen as an extension of their [PLW22, Proposition 3.4] (stated here as Lemma 13). Thus, a concatenated codeword in $C_{\text{out}} \circ C_{\text{in}}$ is of the form $C_{\text{in}}(\sigma_1) \circ \cdots \circ C_{\text{in}}(\sigma_n)$ where $(\sigma_1, \ldots, \sigma_n)$ is an outer codeword.

Then, in the concatenated codeword, every two adjacent inner codewords are separated using a large run of zeros (termed as buffers). Adding these buffers was done also in [GL19, CS22] and the goal is to reduce the loss of synchronization between the receiver and the sender. The final codeword that is transmitted through the channel is

$$C_{\text{in}}(\sigma_1) \circ 0^B \circ \cdots \circ 0^B \circ C_{\text{in}}(\sigma_n)$$

where B is the length of the buffer.

Now, the decoder which receives a corrupted version of the transmitted codeword consists of three steps. First, identifying the buffers following the simple rule: every run of zeros of length greater than some parameter is identified as a buffer. With standard concentration bounds (and appropriate parameters), one can show that almost all buffers are correctly identified as buffers. Moreover, with high probability there are very few "spurious" buffers inside the inner codewords (that is, the channel created a long run of zeros inside an inner codeword and the decoder mistakenly identified it as a buffer). Here we use the fact that in every "not too short" interval in an inner codeword there are many 1s. After this step, every string between two buffers is decoded using decoder of the inner code. Note here that the buffer identification step might trim the codeword from the left and right. However, the inner code achieves capacity on the trimming version of the channel and thus most of the corrupted inner codeword are decoded successfully.

Finally, we run the decoder of the outer codeword which can correct from a small amount of insertions and deletions (insdel errors). Observe that each deleted buffer and spurious buffer contributes at most 3 insdel errors in the outer codeword symbols, and that a failure in the inner code's decoding algorithm results in 2 insdel errors. Thus, as long as with high probability the number of all of these errors can be made as small as we want, then the outer code of rate $1 - \varepsilon - \delta$ that can correct from δ insdel errors can handle those errors. Concluding, we have that with high probability decoding is successful and the rate is close as we want to the rate of the inner code which achieves capacity.

5.1 Construction

Let ε be the desired gap to capacity.

Outer and inner codes. The coding scheme uses code concatenation. For the outer code, as in previous works that construct binary codes for synchronization channels, we shall use a code that can correct from *adversarial* (worst-case) indel (insertions and deletions) errors. We start by defining the relevant metric.

Definition 12. *The* edit distance *between two strings s, s', denoted by* $\text{ED}(s, s')$*, is the minimal number of insertions and deletions needed to convert s into s'.*

we use the code by Haeupler, Rubinstein, and Shahrasbi [HRS19] that can correct adversarial insdel errors.

Theorem 12 ([HRS19]). *For every $\epsilon_{\text{out}}, \delta_{\text{out}} \in (0,1)$ there exists a family of codes $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$, where $\mathcal{C}_n$ has blocklength n, of rate $\mathcal{R}_{\text{out}} = 1 - \delta_{\text{out}} - \epsilon_{\text{out}}$ over an alphabet Σ of size $|\Sigma| = O_{\epsilon_{\text{out}}}(1)$ that can correct $\delta_{\text{out}} n$ adversarial insdel errors. The codes $\mathcal{C}_n$ support linear time encoding and quasi-linear time decoding.*

In other words, this code has the property that the edit distance between any two distinct codewords is at least $2\delta_{\text{out}} n$. We will denote by C_{out} the outer code given by Theorem 12 where we set $\delta_{\text{out}} = \varepsilon/5$. By C_{in}, we denote the inner code that achieves capacity on the 0-trimming BDC-RL-Bounded(d, μ, M) channel, given by Corollary 2 by setting ε (in the corollary) to $\varepsilon/50$ and ζ (in the corollary) to be $\nu/3$ where $\nu = \varepsilon \cdot \mu/5$. We also make sure that each codeword in C_{in} starts and ends with a 1 simply by adding a 1 bit at the beginning and at the end. Clearly, this does not affect the asymptotics. Formally, our encoding process is as follows:

Encoding. Given as input a message $x \in \Sigma^{\mathcal{R}_{\text{out}} n}$, we encode it with the code given in Theorem 12 to obtain an outer codeword $c^{(\text{out})} = (\sigma_1, \ldots, \sigma_n) \in C_{\text{out}} \subset \Sigma^n$. Then, every symbol in $c^{(\text{out})}$, $\sigma_i \in \Sigma = \{0,1\}^{m \cdot \mathcal{R}_{\text{in}}}$, is encoded using the inner code to a codeword that we denote $c^{(\text{in})}_{\sigma_i}$. We thus get a codeword in the concatenated code

$$\left(c^{(\text{in})}_{\sigma_1}, \ldots, c^{(\text{in})}_{\sigma_n}\right) \in C_{\text{out}} \circ C_{\text{in}} \ .$$

Now that we have a codeword in the concatenated code, we add an additional layer of encoding that is crucial for preserving synchronization. Every two adjacent inner codewords are separated by a buffer of zeros of length $\lceil \nu \cdot m/(1 - d(M)) \rceil$ where recall that ν is a small constant such that $\nu = \mu \cdot \varepsilon/5$ (recall that, by Definition 11, μ is such that $d(M) \leq 1 - \mu$ and that $d(M)$ is the largest deletion probability the channel can impose).

Remark 4. We note that in order for the concatenation to make sense, we must have $\mathcal{R}_{\text{in}} m = \lceil \log_2 |\Sigma| \rceil = \lceil \log_2(O_{\epsilon_{\text{out}}}(1)) \rceil$. We shall choose a small enough ϵ_{out} so that the length of the buffer will be $\geq M$, which would imply that the channel will apply deletions to it with probability $d(M)$.

Rate. The codeword length is $mn + (n-1)\lceil \nu m/(1 - d(M)) \rceil$ and, therefore, the rate of the code is

$$\begin{aligned}
\mathcal{R} &= \frac{\log_2 |\Sigma|^{\mathcal{R}_{\text{out}} n}}{mn + (n-1)\lceil \nu m/(1-d(M)) \rceil} \\
&\geq \frac{\mathcal{R}_{\text{in}} \mathcal{R}_{\text{out}}}{1 + \nu/(1-d(M)) + 1/m} \\
&\geq \frac{(1 - \varepsilon/5 - \epsilon_{\text{out}})\mathcal{R}_{\text{in}}}{1 + \varepsilon/5 + 1/m} \\
&\geq \frac{(1 - \varepsilon/4)\mathcal{R}_{\text{in}}}{1 + \varepsilon/4} \\
&\geq (1 - \varepsilon/2)\mathcal{R}_{\text{in}}
\end{aligned}$$

where the second inequality follows since $1 - d(M) > \mu$. The third inequality follows by our assumption that $\delta_{\text{out}} = \varepsilon/5$ and taking ϵ_{out} to be small enough (so that $\varepsilon/5 + \epsilon_{\text{out}} \leq \varepsilon/4$ and $1/m$ is at most $\varepsilon/5$). Now, since $\mathcal{R}_{\text{in}} = \text{Cap}(\text{BDC-RL-Bounded}(d, \mu, M)) - \varepsilon/50$, we get that $\mathcal{R} \geq \text{Cap}(\text{BDC-RL-Bounded}(d, \mu, M)) - \varepsilon$.

Decoding. The decoding consists of the following step:

1. Decoding buffers: Identify all runs of the symbol 0 in the received string that are of length at least $\frac{\nu m}{2}$ and declare that these runs are buffers.

   Denote by $\tilde{c}_1, \ldots, \tilde{c}_s$ the strings in between the buffers (and discard the buffers). Note that each $\tilde{c}_i$ starts and ends with a 1.

2. Inner decoding: Decode each $\tilde{c}_i$ (brute force) into an outer code symbol $\tilde{\sigma}_i$.

3. Outer decoding: Run the outer code decoding algorithm on $\tilde{\sigma}^{(\mathsf{out})} = (\tilde{\sigma}_1, \ldots, \tilde{\sigma}_s)$ to obtain $\tilde{x}$ and return it.

5.2 Analysis

We start by upper bounding the probability of identifying a sent buffer.

Claim 1. *The probability that a specific buffer is not identified is at most* $\exp(-\Omega_{\varepsilon,\mu}(m))$.

Proof. This happens when the channel deletes too many bits from a buffer so that less than $\nu m/2$ bits of the transmitted buffer survived the channel, and we did not identify this buffer when we identify buffers in Step 1.

Let Z be the random variable that corresponds to the number of bits from that buffer that survived the transmission through the BDC-RL-Bounded(d, μ, M). Clearly, $Z \sim \mathrm{Bin}(\lceil \nu m/(1-d(M)) \rceil, 1-d(M))$ and then, By Chernoff's bound

$$\Pr\left[Z < \frac{\nu m}{2}\right] \leq \Pr\left[Z < \left(1 - \frac{1}{2}\right)\nu m\right] < \exp\left(-\frac{1}{8}\nu m\right) .$$

□

Now, we bound the probability that a *spurious buffer* is created by the channel.

Claim 2. *Let* $C_{\mathrm{in}}(\sigma_i)$ *be an inner codeword that is transmitted through* BDC-RL-Bounded(d, μ, M) *and let* y *denote its output. The probability that* y *contains a run of zeros of length at least* $\nu m/2$ *is at most* $\exp(-\Omega(m))$.

Proof. Clearly, in order for the channel to create a run of zeros of length $\nu m/2$, it must be that the channel deleted all 1s from an interval of length at least $\nu m/2$. Fix an interval of length $\nu m/3$ in an inner codeword. By Corollary 2 and recalling that $\zeta = \nu/3$, this interval contains at least $\gamma\nu m/3$ ones for some $\gamma = \gamma(\varepsilon/2, \zeta)$. Let $a_i, i \in [M-1]$ be the number of ones in this interval that belong to runs of length i and denote by a_M the number of bits that belong to runs of length $\geq M$ in this interval. Deleting all these 1s happens with probability

$$d(1)^{a_1} d(2)^{a_2} \cdots d(M)^{a_M} \leq \left(\frac{3 \cdot \left(\sum_{i=1}^{M} a_i d(i)\right)}{\gamma\nu m}\right)^{\gamma\nu m/3} \leq (1-\mu)^{\gamma\nu m/3} = \exp(-\Omega_{\varepsilon,\mu,\gamma}(m)) ,$$

where the first inequality follows from the weighted AM-GM inequality and the second inequality is due to the assumption that $d(1) \leq d(2) \leq \cdots \leq d(M) \leq 1-\mu$ and that $\sum_{i=1}^{M} a_i = \gamma\nu m/3$. Union bounding over all such intervals, we get also that the probability of the existence of a spurious buffer inside an inner codeword is at most $m \cdot \exp(-\Omega_{\varepsilon,\mu,\gamma}(m)) = \exp(-\Omega_{\varepsilon,\mu,\gamma}(m))$. □

Remark 5. Note that the above argument bounds the probability that a spurious buffer is identified inside an inner codeword. However, it can be that the decoder identifies several spurious buffer. The maximal number of spurious buffers the decoder can identify inside an inner codeword is at most $2/\nu = O(1)$.

Claim 3. *Let $0^B \circ C_{\text{in}}(\sigma_i) \circ 0^B$ be an inner codeword surrounded by two buffers and assume that it is transmitted through the channel* BDC-RL-Bounded(d, μ, M)*. Assume that the buffers were identified by the algorithm and that no spurious buffers were created. Denote by $\tilde{c}_j$ the corresponding string obtained after removing the buffers. Then, the probability that $\tilde{c}_j$ is decoded correctly to σ_i is at least $1 - \varepsilon/2$.*

Proof. Observe that $\tilde{c}_j$ is the output of the 00-trimming BDC-RL-Bounded(d, μ, M) on the string $C_{\text{in}}(\sigma_i)$. Thus, by Corollary 2, the decoding failure probability is $\varepsilon/2$. □

With these claims, we can now prove Theorem 11, exactly as in [PLW22] (the proof is given here for completeness).

Proof of Theorem 11. Our goal is to show that with probability $1 - \exp(-\Omega(n))$, it holds that $\text{ED}(\sigma^{(\text{out})}, \tilde{\sigma}^{(\text{out})}) < \delta_{\text{out}} n$. This would imply that Step 3 in our decoding algorithm succeeded and the correct message is returned. There are three types of error that can increase the edit distance between $\sigma^{(\mathsf{out})}$ and $\tilde{\sigma}^{(\mathsf{out})}$: A deleted buffer, a spurious buffer, and wrong inner decoding.

We analyze the contribution of each of the error types (deleted buffer, spurious buffer and wrong inner decoding) on $\text{ED}(\sigma^{(\text{out})}, \tilde{\sigma}^{(\text{out})})$. A deleted buffer causes two inner codewords to merge, and thus be decoded incorrectly by the inner code's decoding algorithm. This introduces two deletions and one insertion in the outer code level. Similarly, one can verify that a single spurious buffer inside an inner codeword introduces one deletion and two insertions. Furthermore, b spurious buffers inside an inner codeword introduce at most $b + 1$ insertions and one deletion. Finally, a wrong inner decoding causes a substitution which is equivalent to one deletion followed by one insertion.

As mentioned above, the outer decoding algorithm fails if $\text{ED}(\sigma^{(\text{out})}, \tilde{\sigma}^{(\text{out})})$. For this to happen, at least one of the following must occur:

1. $\delta_{\text{out}} n/9$ deleted buffers.
2. $\delta_{\text{out}} n/9$ spurious buffers.
3. $\delta_{\text{out}} n/9$ wrong inner decodings.

Observe that for large enough m (equivalently, for small enough ϵ_{out}), we get that the probability for a deleted buffer $\exp(-\Omega_{\varepsilon,\mu}(m)) < \delta_{\text{out}}/10$ and thus, by Chernoff bound, the probability that there are at least $\delta_{\text{out}} n/9$ deleted buffers is at most $\exp(-\Omega(n))$. Furthermore, since the expected number of spurious buffers between two adjacent real buffers is $\exp(-\Omega_{\varepsilon,\mu,\gamma}(m)) < \delta_{\text{out}}/10$ for large enough m and the maximal number of such spurious buffers is $2/\nu$, we can apply Hoeffding's bound (Lemma 3), and get that the total number of all spurious buffers is $< \delta_{\text{out}} n/9$ with probability $1 - \exp(-\Omega(n))$. Finally, since a decoding of an inner codeword fails with probability $\varepsilon/50 = \delta_{\text{out}}/10$ (assuming the buffers surrounding it were detected and there were no spurious buffers inside), we get that the probability that more than $\delta_{\text{out}} n/9$ of these inner decodings failed is at most $\exp(-\Omega(n))$. The claim about the decoding failure probability follows by applying a simple union bound.

We now justify the time complexity of our encoding and decoding algorithms. The complexity of the encoder is as follows. The encoder of the outer code runs in linear time. Then, the encoding of each outer code symbol using the inner code is performed in constant time, and thus the encoding

of all the n symbols is done in $O(n)$. Finally, adding the buffers also takes linear time. The decoding complexity is dominated by the outer code's decoding time which is quasi-linear. Indeed, identifying the buffers and decoding all corrupted inner codewords takes $O(n)$. □

Remark 6. We emphasize the order in which we choose the parameters of the scheme and, in particular, we make sure that there is no circular dependency. First, observe that μ and M are given by the channel BDC-RL-Bounded(d, μ, M).

We first choose the constant ε which is the gap to capacity we want to achieve. This sets δ_{out}, ν, ζ and γ. Then, we choose small enough ϵ_{out} to ensures that all the failure probabilities computed in the proofs of Claim 1 and Claim 2 are indeed smaller than $\delta_{\text{out}}/10$ (recall that deleted buffer and spurious buffer happen with probability $\exp(-\Omega(m)) = \exp(-\Omega(\log_2(|\Sigma|/\mathcal{R}_{\text{in}}))) = (O_{\epsilon_{\text{out}}}(1))^{-\Omega(1)}$).

Remark 7. We remark that the proof technique applied in this section (which is based on [PLW22]) can be applied to any runlength-dependent channel for which we can design buffers to separate inner codewords and prove that with high probability we can identify "most" of these buffers correctly and that with high probability the number of spurious buffers is "small". More specifically, all one needs to do is prove analogous claims to Claim 1 and Claim 2.

In this section, we followed a relatively easy example, the BDC-RL-Bounded(d, μ, M) channel. Indeed, the buffers we used are simply long runs of zeros, and for the proofs of Claim 1 and Claim 2, we used the nature of channel (the deletions inside a run are i.i.d.) and the fact (given in Corollary 2) that our inner codewords contain many 1s in every not-too-short interval (and thus, spurious buffers are not likely to be created).

6 Efficient capacity-achieving codes for multi-trace channels with runlength-dependent deletions

In the previous section, we constructed efficient capacity-achieving codes for runlength-dependent deletion channel. In this section, our objective is to construct efficient capacity-achieving codes for the multi-trace version of the channel. Thus, one can see the result of the previous section as special case of the result we will obtain in this section. More specifically, our goal is to transform the structured capacity-achieving codes from Corollary 2 for a multi-trace runlength-dependent deletion channel into explicit and efficient capacity-achieving codes for the same channel.

This time, our techniques will follow the main ideas of Brakensiak, Li, and Spang [BLS20] who constructed explicit and efficient codes of rate $1-\varepsilon$ that can be reconstructed from $\exp(O_d(\log^{1/3}(\varepsilon^{-1})))$ traces of the binary deletion channel with parameter d (BDC$_d$, for short).

We start by defining the class of multi-trace, runlength-dependent channels for which we aim to construct efficient codes.

Definition 13. *Let* BDC-MT-RL-Bounded(d, T, μ, M) *be a runlength-dependent channel that produces T traces on each input. As in Definition 11, the deletion function is defined by a non-decreasing function $d(\ell) : \mathbb{N} \to [0, 1]$, number $\mu \in (0, 1)$, and an integer M such that $d(\ell) = d(M) < 1 - \mu$ for all $\ell \geq M$.*

The theorem we will prove in this section is as follows.

Theorem 13. *Let* BDC-MT-RL-Bounded(d, T, μ, M) *be a runlength-dependent channel that complies with Definition 11 where μ and M are constants. For every $\varepsilon > 0$, there exists a family of binary codes $\{C_i\}_{i=1}^{\infty}$ for the* BDC-MT-RL-Bounded(d, T, μ, M) *where the block length of C_i goes to infinity as $i \to \infty$ and*

1. *C_i is encodable in quasi linear time and decodable in quadratic time.*

2. *The decoding failure probability is* $\exp(-\Omega(n))$.

3. *The rate of the C_i is* $R > \mathrm{Cap}(\text{BDC-MT-RL-Bounded}(d, T, \mu, M)) - \varepsilon$.

Remark 8. Note that this section's main result recovers the result of the previous section with worse time complexity. This is because in the decoding procedure, we simply use the matching algorithm given by [HS21b] instead of the improved, more involved indexing scheme given in [HRS19]. It can be the case that these techniques also work in this case. However, for simplicity, we are using the simple matching algorithm of [HS21b] in a black-box way (see Lemma 17).

Before diving into the technical details, we present a broader picture, including a description of the main ideas of [BLS20].

Let x be a random string in $\{0,1\}^n$. The average trace reconstruction problem (which has been extensively studied in the literature [BKKM04, HPP18, Cha21, HPPZ20, CDL+22], just to name a few) asks for the minimal number of traces T for which the reconstruction of x succeeds with high probability (the randomness is over the randomness of the channel and the choice of x). Although there has been significant interest in this question, there is still an exponential gap between the lower bound ($\widetilde{\Omega}(\log^{5/2} n)$ [Cha21]) and upper bound on T ($\exp(\widetilde{O}(\log^{1/5} n))$ [Rub23]).

Motivated by the challenges in developing DNA-based storage systems, [CGMR20] and then a subsequent work [BLS20] introduced the *coded trace reconstruction* problem, a natural extension of the trace reconstruction problem. In this setting, the goal is to design a code with rate close to 1 as possible such that any codeword can be reconstructed from (very) few traces, with high probability. In both works, the authors provide constructions of efficient codes with rate $1-\varepsilon$ that are trace-reconstructable $O_\varepsilon(1)$ (see Remarks 1.6 and 1.7 in [BLS20], which compare the results of both papers).

The main theorem of [BLS20] (Theorem 1.4) turns an upper bound on T in the average case into an explicit and efficient code with rate $1-\varepsilon$ that is efficiently trace-reconstructable. Applying their result with the current best upper bound, they that the number of traces is $\exp(O_d(\log^{1/3}(\varepsilon^{-1})))$.

The major problem in generalizing the concatenated scheme from the previous section to the multi-trace setting is that we need to synchronize the traces. Indeed, assume that we have an inefficient code that achieves capacity in the multi-trace setting and that we use the same encoding process as in the previous section and denote by $a_1, \ldots, a_n$ the encoded inner codewords. Let $z_1, \ldots, z_T$ be the received traces. We can easily show that in most traces, most of the buffers are detected and that there are not too many spurious buffers. Thus, from each trace $t \in [T]$ we extract n_t binary strings $b_1^{(t)}, \ldots, b_{n_t}^{(t)}$ where each one of these strings is a trace of an inner codeword. However, to perform trace reconstruction to recover, say a_i, we need to know in each trace t, which $j \in [n_t]$ is such that $b_j^{(t)}$ is the corresponding trace of a_i in the t-th trace.

To overcome this problem, [BLS20] used two inner codes in their encoding process. We shall do the same thing here. The first code encodes the information symbols and is robust in the multi-trace setting. The second inner code encodes synchronization symbols (these are symbols that do not carry data information but are used to preserve synchronization in the presence of insertions and deletions) and can be reliably recovered from only a *single* trace of the channel. More precisely, the encoding process takes a pair of information symbol and synchronization symbol (r_i, s_i) and output (a_i, b_i) where a_i and b_i are binary codewords in two different inner codes, as described above. Then, we shall place the buffers in a slightly different way than what we did in the previous section (also slightly different than what [BLS20] did), but the main idea is similar.

Note that now, before performing the "inner" trace reconstruction step, we add another step (as in [BLS20]) that aligns each trace using the synchronization symbols (in each trace most of them are decoded correctly). Then, in each trace, we know the right location of most of the corrupted inner codewords, and we can run the inner trace reconstruction.

6.1 Auxiliary results

In this subsection, we define and cite several ingredients that we will use in our construction.

Synchronization strings. Synchronization strings are special *indexing strings* that were first introduced in [HS21b].

Definition 14. *A string $S \in \Sigma^n$ is an η-synchronization string if for every $1 \leq i < j < k \leq n$, it holds that* $\mathrm{ED}(S[i,j), S[j,k)) > (1-\eta)(k-i)$.

The following theorem shows that one can efficiently construct such strings over a 'not too large' alphabet.

Theorem 14 (Theorem 1.2, [CHL+19])**.** *For every natural number n and every $\eta \in (0,1)$, there exists a polynomial-time algorithm that constructs a τ-synchronization sequence over an alphabet of size $O(\eta^{-2})$.*

Synchronization strings were the main ingredient in [HS21b] for constructing codes that can correct insdel errors. The main idea behind [HS21b] is to take a code $\mathcal{C}$ that can correct substitutions and erasures and combine it with a single η-synchronization string $S = s_1 \cdots s_n$ of length n. The resulting code is

$$\mathcal{C}^{\mathrm{ID}} := \{((c_1, s_1), \ldots, (c_n, s_n)) \mid c \in \mathcal{C}\} .$$

Note that the synchronization string is the same for all codewords and therefore does not carry information. Therefore, to ensure that the synchronization string has a negligible impact on the rate, the alphabet size of the code $\mathcal{C}$ must be large compared to the alphabet size of the synchronization string.

The purpose of the synchronization symbols is to transform the insdel errors into erasures and substitutions. This is achieved by a "matching" algorithm given by the following lemma.

Lemma 16 ([HS21a, Lemma 2.2])**.** *Let $S = s_1 s_2 \cdots s_n$ be an η synchronization string. There exists an algorithm that on input $(m'_1, s'_1), \ldots, (m'_{n'}, s'_{n'})$, and $S = s_1 \cdots s_n$, guesses the position of all received symbols in the sent string such that the position of all but $O(\sqrt{\eta} \cdot n)$ of the symbols that are not deleted are guessed correctly. This algorithm runs in time $O_\eta(n^2)$.*

The algorithm which yields this lemma [HS21a, Algorithm 1] essentially transforms the δn insdel errors applied to a codeword $((c_1, s_1), \ldots, (c_n, s_n))$ into erasures and substitution that are applied to $(c_1, \ldots, c_n)$. Specifically, the algorithm outputs $(y_1, \ldots, y_n) \in (\Sigma_{\mathcal{C}} \cup \{\perp\})^n$ such that $(y_1, \ldots, y_n)$ can be obtained from $(c_1, \ldots, c_n)$ by performing e erasures and t substitutions where $e + 2t \leq \delta n + 12\sqrt{\eta} n$.

We will rephrase Lemma 16 so that it fits our decoding algorithm. We start with a definition of a *true indexing matching*.

Definition 15. *Let $S = s_1 \cdots s_n$ be string and let Let $S' = s'_1 \cdots s'_m$ be a string that is obtained from S by performing δn insdel errors. A* true indexing matching *between S and S' consists of a set $I \subseteq [n]$ and a map $\Gamma : I \to [m]$ such that for all $a \in I$:*

- *the symbol in the a-th position in S (s_a) was not deleted.*
- *The new position of s_a in S' is $\Gamma(a)$.*

Lemma 17 ([HS21a, Section 2, implicit]). *Let $S = s_1 s_2 \cdots s_n$ be an η synchronization string and let $S' = s'_1 \cdots s'_m$ be a string that is obtained from S by performing δn insdel errors. Then, there exists a matching algorithm that produces indices $i_1 < \ldots < i_t$ and $j_1 < \ldots < j_t$ such that*

- *$t \geq n - \delta n - 2\sqrt{\eta} n$.*
- *$s_{i_\ell} = s'_{j_\ell}$ for all $\ell \in [t]$.*
- *There are at most $\sqrt{\eta} n$ indices $\ell \in [t]$ such that $\Gamma(i_\ell) \neq j_\ell$.*

Good codes correcting substitutions. For our scheme we will need to use an outer code over a large (but constant) alphabet that can correct substitutions. In [BLS20], the authors adapted the efficient binary codes given in [GI05] into efficient codes over any alphabet that is a power of 2. They used these codes to construct codes over *large* (but constant) alphabets that are trace reconstructible.

Proposition 1 ([BLS20, Proposition 2.17]). *For every ε and Σ whose size is a power of 2, there exists an infinite family of codes over Σ of rate $1-\varepsilon$ encodable in linear time and decodable in linear time from up to a fraction $\frac{1}{40}\varepsilon^3$ of worst-case substitution errors.*

Remark 9. We note that in [BLS20], for their binary codes, the authors used a construction of Justesen [Jus72] that gives rate $1-\varepsilon$ codes correcting $\Theta(\varepsilon^2/\log(\varepsilon^{-1}))$ for *all* large enough block length (see [BLS20, Proposition 2.17]). The caveat is that the run time of the encoder and decoder is quadratic. In this paper, we focus on constructing an infinite family of capacity achieving codes, and thus we choose to use Proposition 1.

6.2 Construction

Parameters We start by introducing the components of our construction which include an outer code that can correct from substitutions and two inner codes that we obtain from Corollary 2. We shall introduce parameters throughout the description of the construction. To aid readability, these are also listed in Table 1. We note here that there was no attempt to optimize the parameters. Let ε be the desired gap to capacity.

Outer code. Let C_{out} be a code of length n_{out} and rate $1 - \varepsilon/4$ over the alphabet Σ_R that can correct $\delta_{\text{out}} = \varepsilon^3/40$ fraction of worst-case substitution errors. This code exists by Proposition 1.

Inner codes. As in [BLS20], we shall encode the data and the synchronization symbols separately using two inner codes. Set $\zeta = (1 - d(M))\nu/4$ and $\varepsilon_R = \varepsilon_S = \varepsilon^4/T$.

1. **Code for the synchronization symbols.** Let $\eta = \varepsilon^8/T$ and let $S = s_1 \cdots s_{n_{\text{out}}}$ be an η-synchronization string of length n_{out} over the alphabet Σ_S where, by Theorem 14, we have that $|\Sigma_S| = O(\varepsilon^{16}/T^2)$. However, we clearly can use a larger alphabet. We shall assume that $\log|\Sigma_S| = (R_S)^{-1} \cdot \log n_R$ where $\mathcal{R}(C_S)$ is a constant in $(0, 1)$ and n_R is a large enough constant to be determined at the end. Let C_S be a code for the BDC-RL-Bounded(d, μ, M) (this is the single trace version, i.e., for $T = 1$) channel guaranteed by Corollary 2 where $\mathcal{R}(C_S)$ denotes its rate ($\mathcal{R}(C_S)$ can be as close to capacity as we want). C_S is equipped with

Parameter	Value	Description/Comments
ε		Gap to capacity
T	Constant	Number of traces
n_{out}	$\to \infty$	Length of C_{out}
δ_{out}	$\varepsilon^3/40$	Substitution correction capability of C_{out}
n_R	Large enough constant	Length of the inner code C_R
n_S	$\log n_R$	Length of the inner code C_S
$\mathcal{R}$	Constant	Rate of the inner code C_R
μ	Constant	Channel parameter
M	Constant	Channel parameter
ν	$\varepsilon \cdot \mu$	small constant for buffer size
B	$\frac{\nu}{16(1-d(M))} \cdot n_R$	Buffer size
ξ	ε^4/T	Number of bad pairs (see Definition 16 below)
$\varepsilon_R, \varepsilon_S$	ε^4/T	Decoding error probability of C_R, C_S, respectively
ζ	$(1-d(M))\nu/4$	For Corollary 2
γ	$\gamma(\varepsilon_R, \zeta)$	Given by Corollary 2
η	ε^8/T	Synchronization parameter

Table 1: Parameters of the scheme. All parameters are constant with respect n_{out}.

an encoder $\mathsf{Enc}_S : \Sigma_S \to \{0,1\}^{n_S}$ and a decoder $\mathsf{Dec}_S : \{0,1\}^* \to \Sigma_S$ such that for every codeword $c \in C_S$ the probability that Dec_S fails to decode c after transmission through the BDC-RL-Bounded(d, μ, M) is at most ε_S. We also assume that C_S starts with a 1 bit and ends with a 0 bit.

By Corollary 2, the length of C_S which is denoted by n_S is at least $n(\varepsilon_S, \zeta)$ for some function n.

2. **Code for the content symbols.** For the content symbols, i.e., the symbols of the outer code C_{out}, we will use again the code guaranteed by Corollary 2, but now for T traces. Namely, we get a code C_R of length n_R, with rate $\mathcal{R}(C_R) = \text{Cap}(\text{BDC-MT-RL-Bounded}(d, T, \mu, M)) - \varepsilon_R$ that is (T, ε_R)-trace reconstructable for the BDC-RL-Bounded(d, μ, M) channel. C_R is equipped with an (inefficient) encoder $\mathsf{Enc}_R : \Sigma_R \to \{0,1\}^{n_R}$ where $\log |\Sigma_R| = \mathcal{R}^{-1} \cdot n_R$ and an (inefficient) decoder $\mathsf{Dec}_S : \{0,1\}^* \to \Sigma_R$ such that for every codeword $c \in C_S$, the probability that Dec_S fails to decode c after transmission through the BDC-MT-RL-Bounded(d, T, μ, M) is at most ε_R. We assume that C_R starts with a 0 bit and ends with a 1 bit.

   Note again that by Corollary 2, n_R is at least $n(\varepsilon_R, \zeta)$ for some function n.

Since n_R and n_S can be as large as we want, starting from some $n(\varepsilon_S, \zeta)$, we make sure that $n_R = 2^{n_S}$. Let γ is the constant obtained by Corollary 2 when given the parameters ε_S and ζ as declared here.

Encoding. Let $\nu = \varepsilon \cdot \mu$ and set $B = \frac{\nu}{16(1-d(M))} \cdot n_R$. By choosing a large enough n_R, we make sure that $B > M$ where recall that M is a channel parameter (for every run of length at least M, the deletion probability applied on each bit of this run is $d(M)$).

The encoding process is given next.

1. Let $m \in \Sigma_R^{k_{\text{out}}}$ be a message. Encode it using the outer code C_{out} to obtain $(r_1, \ldots, r_n) \in \Sigma_R^{n_{\text{out}}}$.

2. Encode each r_i and s_i using the respective inner codes. Namely, let $a_i = \mathsf{Enc}_R(r_i)$ and $b_i = \mathsf{Enc}_S(s_i)$.

3. Add buffers of the symbol 0 of length B between every a_i and b_i and add buffers of the symbol 1 length B between b_i and a_i. The codeword has the following form

$$c = a_1 \circ 0^B \circ b_1 \circ 1^B \circ a_2 \circ 1^B \circ b_2 \circ \cdots \circ a_n \circ 0^B \circ b_n \circ 1^B \ .$$

Rate. The length of the codeword is $(n_R + 2B + n_S) \cdot n_{\text{out}}$. Thus, the rate is

$$\begin{aligned}
\mathcal{R} &= \frac{\log |\Sigma_R|^{(1-\varepsilon/4)n_{\text{out}}}}{(n_R + 2B + n_S) \cdot n_{\text{out}}} && (16)\\
&= \frac{(1-\varepsilon/4) \cdot \mathcal{R}(C_R) \cdot n_R}{n_R + 2B + n_S} \\
&= \frac{(1-\varepsilon/4) \cdot \mathcal{R}(C_R)}{1 + \frac{2\nu}{16(1-d(M))} + \frac{\log n_R}{n_R}} \\
&\geq \frac{(1-\varepsilon/4) \cdot \mathcal{R}(C_R)}{1+\varepsilon/4} \\
&\geq (1-\varepsilon/2) \cdot \mathcal{R}(C_R) && (17)\\
&\geq \text{Cap}(\text{BDC-MT-RL-Bounded}(d, T, \mu, M)) - \varepsilon \ , && (18)
\end{aligned}$$

where the first inequality holds since, for large enough n_R, we have $(\log n_R)/n_R \leq \varepsilon/8$ and also since $1 - d(M) \leq \mu$ (by the channel's definition), $\nu/(4(1-d(M))) \leq \varepsilon/4$. The last inequality follows since $\mathcal{R}(C_R) = \text{Cap}(\text{BDC-MT-RL-Bounded}(d, T, \mu, M)) - \varepsilon_R$ where $\varepsilon_R = \varepsilon^4/T < \varepsilon/2$.

Decoding. Let $z^{(t)}$ be the received string at the t-th trace.

1. Trace alignment using the synchronization symbols.

   (a) Identify 1-buffers. Every run of zeros of length greater than $(1 - d(M))B/2$ is identified as a 1-buffer.
   Let $z_1^{(t)}, \ldots, z_{n_t}^{(t)}$ be the strings in between the buffers.

   (b) Identify 0-buffers. For every $i \in [n_t]$, in $z_i^{(t)}$, every run of zeros of length greater than $(1 - d(M))B/2$ is identified as a 0-buffer. If there are no 0-buffers in $z_i^{(t)}$ or there are more than a single 0-buffer, discard this $z_i^{(t)}$. Otherwise, divide this $z_i^{(t)}$ into two parts according to the 0-buffer that was found.
   Let $(x_1^{(t)}, y_1^{(t)}), \ldots, (x_{n_t'}^{(t)}, y_{n_t'}^{(t)})$ be the pairs that correspond to the $s_i^{(t)}$ containing a single 0-buffer.

   (c) For every $i \in [n_t']$, decode the synchronization symbol from $y_i^{(t)}$. Namely, $\tilde{s}_i^{(t)} = \mathsf{Dec}_S(y_i^{(t)})$.

   (d) Run the algorithm from Lemma 17 on $\tilde{s}_1^{(t)}, \ldots, \tilde{s}_{n_t'}^{(t)}$ to obtain indices $\tilde{i}_1^{(t)}, \ldots, \tilde{i}_{n_t''}^{(t)}$ and $\tilde{j}_1^{(t)}, \ldots, \tilde{j}_{n_t''}^{(t)}$.

   (e) For every $p \in [n_t'']$, let $\tilde{a}^{(t)}_{\tilde{i}_p^{(t)}} := x^{(t)}_{\tilde{j}_p^{(t)}}$ and for $p \notin \{\tilde{i}_1^{(t)}, \ldots, \tilde{i}_{n_t''}^{(t)}\}$, set $\tilde{a}_p^{(t)} = \perp$.

2. Trace reconstruction. For all $i \in [n]$, let $\tilde{r}_i = \mathsf{Dec}(\tilde{a}_i^{(1)}, \ldots, \tilde{a}_i^{(T)})$.

3. Outer code correction. Run $\mathsf{Dec}_{\text{out}}(\tilde{r}_1, \ldots, \tilde{r}_n)$ and return the output.

6.3 Analysis

The following claims (Claims 4-7) bound the probabilities of *bad* events that are related to the correct identification of the buffers. The first claim bounds the probability that a 1-buffer is not detected in Step (1a).

Claim 4. *The probability that a* 1*-buffer is not identified is at most* $\exp(-\Omega_{\varepsilon,\mu}(n_R))$.

Proof. The length of a buffer is $B = \frac{\nu}{16(1-d(M))} \cdot n_R$. In order for the decoding algorithm to consider it as a run in an inner codeword, the channel must delete at least $(1-d(M))B/2$ bits from it. Since every bit in the buffer is deleted independently with probability $d(M)$ (recall that $B > M$), the expected length of a buffer after going through the channel is $(1-d(M))B$. By applying the Chernoff bound, we get that the probability that less than $(1-d(M))B/2$ bits of buffer survived the transmission is at most

$$\exp(-(1-d(M))B/8) \leq \exp(-\nu n_R/128) \leq \exp(-\Omega_{\varepsilon,\mu}(n_R)) \ .$$

□

We are now interested in bounding the probability that a 0-buffer is not identified.

Claim 5. *Let* $z_i^{(t)}$ *be a string obtained after Step (1a). Assume that* $z_i^{(t)}$ *does not contain* 1*s that belong to a* 1*-buffer that was not identified by the algorithm and that the buffers that were identified before and after* $z_i^{(t)}$ *are indeed two adjacent* 1*-buffers. Then, the probability that the* 0*-buffer inside* $z_i^{(t)}$ *not identified is at most* $\exp(-\Omega_{\varepsilon,\mu}(n_R))$.

Proof. The proof is identical to the proof of Claim 4. □

Our next goal is to bound the probability of detecting "spurious" 0-buffers inside a $z_i^{(t)}$. That is, we upper bound the probability that the channel deletes many consecutive 1s in such a way that a long run of 0s is created where all of its bits belong to an inner codeword and the algorithm mistakenly thinks that it is a 0-buffer.

Claim 6. *Let* $z_i^{(t)}$ *be a string obtained after Step (1a). Assume that* $z_i^{(t)}$ *does not contain* 1*s that belong to a* 1*-buffer that was not identified by the algorithm and the buffers that were identified before and after* $z_i^{(t)}$ *are indeed two adjacent buffers. Then, the probability that the algorithm identifies a* 0*-buffer such that all of the bits of that buffer are bits of an inner codeword is at most* $\Omega_{\varepsilon,\mu,\gamma}(n_R))$. [6]

Proof. The computation is similar to the one done in the previous section, yet a bit more delicate. Since the 1-buffers surrounding $z_i^{(t)}$ were identified and correspond to adjacent 1-buffers in the original codeword, $z_i^{(t)}$ contains bits that correspond to $a_{i'} \circ 0^B \circ b_{i'}$ for some i'. We write $z_i^{(t)}$ as $z_i^{(t)} = \tilde{a}_{i'} \circ 00\cdots0 \circ \tilde{b}_{i'}$ where all the bits of $\tilde{a}_{i'}, \tilde{b}_{i'}$ correspond to $a_{i'}, b_{i'}$, respectively, and the run of zeros in between contains 0 bits that correspond to the 0-buffer in the original codeword.

In order to create a run of zeros (in which all the zeros correspond to inner codewords) of length at least $(1-d(M))B/2$, the channel must delete all 1s from at least an interval of this size. Note here that the length of $\tilde{b}_{i'}$ is $\leq \log n_R$ which is less than $(1-d(M))B/2$, the threshold from which a buffer is considered (for large enough n_R). Thus, a spurious 0-buffer can be created only inside $\tilde{a}_{i'}$.

[6] Note that if some of the bits belong to a 0-buffer, then the 0s that originally belonged to the inner codeword are merged to the 0-buffer. Thus, a spurious 0 buffer occurs only when all its 0 bits belong to an inner codeword.

By the properties of our inner codes and by recalling that $\zeta = (1-d(M))\nu/4$, any interval of length $\zeta n_R = (1-d(M))B/4$ must contain at least $\gamma\zeta n_R = \gamma(1-d(M))B/4$ ones. Let $a_i, i \in [M-1]$ be the number of ones in this interval that belong to runs of length i and denote by a_M the number of bits that belong to runs of length $\geq M$. Deleting all these ones happens with probability

$$\begin{aligned} d(1)^{a_1} d(2)^{a_2} \cdots d(M)^{a_M} &\leq \left(\frac{4 \cdot \sum_{i=1}^{M} a_i d(i)}{\gamma(1-d(M))B} \right)^{\frac{\gamma(1-d(M))B}{4}} \\ &\leq (1-\mu)^{\gamma\nu n_R/8} \\ &\leq \exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R)) \end{aligned}$$

where the first inequality follows from the weighted AM-GM inequality and the second inequality is due to the assumption that $d(1) \leq d(2) \leq \cdots \leq d(M) \leq 1-\mu$ and that $\sum_{i=1}^{M} a_i = \gamma(1-d(M))B/4$.

Now, union bounding over all such intervals, the probability that there exists a spurious buffer is at most $n_R \cdot \exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R)) = \exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R))$. □

The final claim regarding the buffers bounds the probability that a spurious 1-buffer is created inside the output of channel in an encoded pair (a_i, b_i).

Claim 7. *Assume that $a_i \circ 0^B \circ b_i$ is transmitted through the channel and let $\hat{z}$ be the received output. Then, the probability that a spurious 1-buffer is identified in $\hat{z}$ is at most $\exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R))$. Moreover, the expected number of spurious 1-buffers in $\hat{z}$ is at most $\exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R))$ and the maximal number of spurious buffers in $\hat{z}$ is at most $32\nu^{-1}$.*

Proof. The claim about the probability to have a spurious 1-buffer in $\hat{z}$ (and the bound on the expected number of spurious 1-buffers) follows by the same argument as in Claim 6. The size of a buffer is at least $(1-d(M))B/2$. Thus, the number of spurious buffers is at most $\frac{2n_R}{(1-d(M))B} \leq \frac{32}{\nu}$. □

Next, we define a pair $(x_i^{(t)}, y_i^{(t)})$ (these pairs are obtained in Step 1b) to be a *good pair* if there exists a $j \in [n_{\text{out}}]$ such that all the bits of $x_i^{(t)}$ belong to a_j and all the bits of $y_i^{(t)}$ belong to b_j. Equivalently, this means that the 1-buffers that are to the left and to the right of $a_j \circ 0^B \circ b_j$ were identified (and there were no spurious 1-buffers inside) and the 0-buffer between a_j and b_j was identified (and there were no spurious 0-buffers).

Definition 16. *Let $z^{(t)}$ be the t-th trace and let $(x_1^{(t)}, y_1^{(t)}), \ldots, (x_{n_t'}^{(t)}, y_{n_t'}^{(t)})$ be the output of Step (1b). We call a pair $(x_i^{(t)}, y_i^{(t)})$ a* good pair *if there exists $j \in [n_{\text{out}}]$ such that all the bits of $x_i^{(t)}$ are bits of a_j and all the bits of $y_i^{(t)}$ are bits of b_j.*

Our next proposition states that, with high probability, after Step (1b), there are at least $(1-\xi) \cdot n_{\text{out}}$ good pairs. Formally, we have the following.

Proposition 2. *Let $z^{(t)}$ be the t-th trace. Let $(x_1^{(t)}, y_1^{(t)}), \ldots, (x_{n_t'}^{(t)}, y_{n_t'}^{(t)})$ be the output of Step (1b). Then, with probability $1 - \exp(-\Omega(n_{\text{out}}))$, at least $n_{\text{out}}(1-\xi)$ of them are good pairs.*

Proof. Let $z_1^{(t)}, \ldots, z_{n_t}^{(t)}$ be the strings obtained in Step 1a after identifying 1-buffers. We start by counting how many $z_i^{(t)}$s are such that the 1-buffers to the right and to the left of the $z_i^{(t)}$ are two adjacent *genuine* 1-buffers (by *genuine* we mean that bits of the identified buffer are indeed bits of a 1-buffer and not bits of run in an inner codeword). We shall call such a $z_i^{(t)}$ a *good* $z_i^{(t)}$.

By Claim 4, the probability that a specific 1-buffer is deleted is at most $\exp(-\Omega_{\varepsilon,\mu}(n_R)) < \xi/10$ where the inequality holds for large enough n_R (recall that $\xi = \varepsilon^4/T$ and that n_R will be a constant that will be chosen in the end). Thus, the probability that more than $\xi n_{\text{out}}/9$ genuine 1-buffers are not identified is at most $\exp(-\Omega(n_{\text{out}}))$. By Claim 7, the expected number of spurious 1-buffers between two adjacent genuine 1-buffers is at most $\exp(-\Omega_{\nu,\mu,\gamma}(n_R)) < \xi/10$ and the maximal number of spurious buffers in the channel output on $a_j \circ 0^B \circ b_j$ is at most $32\nu^{-1}$. Thus, we can apply Hoeffding's bound (Lemma 3), and get that the probability of having more than $\xi n_{\text{out}}/9$ spurious 1-buffers in the whole received string is at most $\exp(-\Omega(n_{\text{out}}))$. Thus, with probability $1 - \exp(-\Omega(n_{\text{out}}))$ at least $n_{\text{out}}(1 - \xi/9)$ genuine 1-buffers were identified and at most $\xi n_{\text{out}}/9$ spurious buffers were identified. Consequently, it is easy to observe that there must be at least $n_{\text{out}}(1 - 5\xi/9)$ $z_i^{(t)}$s that are good $z_i^{(t)}$s. Indeed, every deleted genuine buffer in the worst case transforms two potential good z_is into one bad z_i, and every spurious 1-buffer transforms one potential good z_i into two bad z_is.

We now turn to look inside the good $z_i^{(t)}$s. A good $z_i^{(t)}$ will be transformed into a good pair (see Definition 16) if the 0-buffer is correctly identified and there are no spurious 0-buffers inside. According to Claim 5, the 0-buffer is deleted with probability $\exp(-\Omega_{\varepsilon,\mu}(n_R)) < \xi/10$, and according to Claim 6 the probability to have a spurious 0-buffer in $z_i^{(t)}$ is at most $\exp(-\Omega_{\varepsilon,\mu,\gamma}(n_R)) < \xi/10$. Thus, with probability at least $1 - \xi/5$, a good $z_i^{(t)}$ is transformed into a good pair $(x_i^{(t)}, y_i^{(t)})$. Note that by the behavior of the channel, the event that $z_i^{(t)}$ is transformed into a good pair is independent of all other $z_i^{(t)}$. Thus, applying a Chernoff bound we get that with probability $1 - \exp(-\Omega(n_{\text{out}}))$, there are at least $(1 - \xi/4) \cdot n_{\text{out}}(1 - 5\xi/9) \geq n_{\text{out}}(1 - \xi)$ good pairs. □

Remark 10. We emphasize that in our construction n_R is a constant (with respect to n_{out}) that can be made as large as we wish. Specifically, we need n_R to be such that all the inequalities that involve it in the proof of Proposition 2 will hold. Also, recall from the construction that n_R is also at least $2^{n(\varepsilon_S,\zeta)}$ for some function n. This implies that there exists $n_{R,0} = n_{R,0}(\varepsilon, \varepsilon_S, \mu, \gamma, \zeta)$ such that for every $n_R > n_{R,0}$, all the inequalities are true.

Our next claim shows that, with high probability, the synchronization symbol is successfully decoded in almost all of the good pairs.

Claim 8. *With probability at least* $1 - \exp(-\Omega(n_{\text{out}}))$*, there are at least* $(1 - 2\varepsilon_S) \cdot n_{\text{out}}(1 - \xi)$ *good pairs* $(x_i^{(t)}, y_i^{(t)})$ *for which* $y_i^{(t)}$ *was correctly decoded.*

Proof. $y_i^{(t)}$ is the output of a 01-trimming BDC-RL-Bounded(d, μ, M) on some input x. According to the inner code that we chose for the synchronization symbols, we have the assumption that the probability of decoding error is at most ε_S. Thus, by the independentness induced by the channel, we can apply a Chernoff bound (Lemma 2) and get the claimed result. □

Now, we prove the correctness of our decoding algorithm.

Proposition 3. *Fix the parameters as described in Section 6.2 and Table 1. Let* $m \in \Sigma_{\text{out}}^{(1-\varepsilon/4)n_{\text{out}}}$ *be a message that is encoded to a codeword* c *using our encoding algorithm. Let* $z^{(1)}, \ldots, z^{(T)}$ *be* T *traces of* c *under the channel* BDC-RL-Bounded(d, μ, M)*. Then, with probability* $1 - \exp(-\Omega(n_{\text{out}}))$*, the decoding algorithm on input* $z^{(1)}, \ldots, z^{(T)}$ *outputs the correct message* m*.*

Proof. For a fixed trace, say the t-th trace, by Proposition 2 and Claim 8, with probability at least $1 - \exp(-\Omega(n_{\text{out}}))$ there are at least $(1 - 2\varepsilon_S)(1 - \xi)n_{\text{out}}$ of pairs (x_i, y_i) for which the decoding of y_i resulted in the correct synchronization symbol. Also, observe that the number of total pairs

(good and bad), denoted by n'_t in Step (1b) can be upper bounded by $n'_t \leq n_t \leq (1+2\xi/9)n_{\text{out}}$. Indeed, n_t is the number of z_is in Step (1a), and clearly in Step (1b) this number can only decrease. Furthermore, note that $n_{\text{out}} - n_t \leq 2 \cdot \#\text{spurious buffers}$, and so, by the proof of Proposition 2, with probability $1-\exp(-\Omega(n))$ the number of spurious buffers is at most $\xi n_{\text{out}}/9$.

This implies that the symbols $\tilde{s}_1^{(t)}, \ldots, \tilde{s}_{n'_t}^{(t)}$ obtained in Step (1c) can be obtained from $S = s_1 \cdots s_n$ by performing at most $(2\varepsilon_S + \xi)n_{\text{out}}$ deletions and at most

$$n'_t - (1-2\varepsilon_S-\xi)n_{\text{out}} \leq (1+2\xi/9)n_{\text{out}} - (1-2\varepsilon_S+\xi)n_{\text{out}} \leq (2\varepsilon_S+2\xi)n_{\text{out}}$$

insertions. In total, there are at most $(4\varepsilon_S+3\xi)n_{\text{out}}$ insdel errors between S and $\tilde{s}_1^{(t)}, \ldots, \tilde{s}_{n'_t}^{(t)}$. By Lemma 17, the number of correctly matched synchronization symbols is at least $n_{\text{out}}(1-4\varepsilon_S - 3\xi - 3\sqrt{\eta})$. Thus, after Step (1e), we have $(\tilde{a}_1^{(t)}, \ldots, \tilde{a}_{n_{\text{out}}}^{(t)})$ where at least $n_{\text{out}}(1-4\varepsilon_S-3\xi-3\sqrt{\eta})$ indices $j \in [n_{\text{out}}]$ are such that $\tilde{a}_j^{(t)}$ is a trace of a 10-trimming BDC-RL-Bounded(d,μ,M) channel applied to a_j. In this case, we say that the *trace-alignment procedure succeeded* and it happens with probability $1-\exp(-\Omega(n_{\text{out}}))$. Union bounding over all T traces, we have that with probability at least $1 - T\cdot\exp(-\Omega(n_{\text{out}})) = 1-\exp(-\Omega(n_{\text{out}}))$, the trace-alignment procedure succeeds for all traces.

Now, given that the trace-alignment procedure succeeded for all traces, we turn to count for how many indices $j \in [n_{\text{out}}]$ the vector $(\tilde{a}_j^{(1)}, \ldots, \tilde{a}_j^{(T)})$ represents indeed T traces of a_j. This happens if for all traces the pair (a_j, b_j), after being encoded and transmitted through the channel complies with the following conditions for the t-th trace, for all $t \in \{1, \ldots, T\}$:

- After Step (1a), there exists an index i such that *all* the bits in $z_i^{(t)}$ correspond to $a_j \circ 0^B \circ b_j$.
- After Step (1b), the pair $(x_i^{(t)}, y_i^{(t)})$ is a good pair. In particular, all the bits of $x_i^{(t)}$ correspond to a_j and all the bits of $y_i^{(t)}$ correspond to b_j.
- After Step (1c), we have $\tilde{s}_i^{(t)} = s_j$, namely, the decoding of $y_i^{(t)}$ to the synchronization symbol succeeded.
- After Step (1d), we get that the positioning of $\tilde{s}_i^{(t)}$ in the synchronization string $s_1 \cdots s_n$ is correct.

In each trace, there are at most $(4\varepsilon_S + 3\xi + 3\sqrt{\eta})n_{\text{out}}$ indices $j \in [n_{\text{out}}]$ for which one of the conditions does not hold. Thus, in total there are at most $T \cdot (4\varepsilon_S + 3\xi + 3\sqrt{\eta})n_{\text{out}}$ indices $j \in [n_{\text{out}}]$ for which there is a trace $t \in [T]$ such that one of the conditions above does not hold.

Now, recall that our code C_R is such that the decoding failure probability under the 10-trimming BDC-MT-RL-Bounded(d,T,μ,M) channel is ε_R. Thus, the probability that there are more than $(1-2\varepsilon_R)\cdot n_{\text{out}}(1-4T\varepsilon_S - 3T\xi - 3T\sqrt{\eta})$ failures when trace-decoding these symbols is at most $\exp(-\Omega(n_{\text{out}}))$. Recalling that $\xi = \varepsilon_R = \varepsilon_S = \varepsilon^4/T$ and $\eta = \varepsilon^8/T$, with probability $1-\exp(-\Omega(n))$ we get that for at least a

$$\left(1 - \frac{2\varepsilon^4}{T}\right) \cdot \left(1 - 10\varepsilon^4\right) \geq 1 - 12\varepsilon^4 \geq 1 - \frac{\varepsilon^3}{40}$$

fraction of indices $j \in [n_{\text{out}}]$ the trace-decoding procedure of C_R succeeds in Step 2. Therefore, with probability $1-\exp(-\Omega(n))$ the Hamming distance between $(r_1, \ldots, r_n)$ (the original outer codeword) and $\tilde{r}_1, \ldots, \tilde{r}_n$ is at most $\varepsilon^3 n_{\text{out}}/40$ and thus, by Proposition 1, the decoder of the outer code can decode it. □

We now prove Theorem 13.

Proof of Theorem 13. Note that the claim about the rate is given in Section 6.2, specifically, the rate is computed in Equation (17). Further, observe that the claim about the decoding failure probability is given by Proposition 3. Thus, we are left to justify the claim about the complexity of the encoding and decoding algorithms.

We start with the encoding algorithm. In the first step we run the encoding algorithm of the inner codes which runs in time $O(n_{\text{out}})$. Then, in the second step, each symbol in the outer codeword is encoded using the encoder of C_R and every symbol of the synchronization symbol is encoded using the encoder of C_S. Recall that the lengths of C_R is n_R (and the length of C_S is $\log n_R$) which is constant with respect to n_{out}. Therefore, the complexity of this step is $O(n_{\text{out}})$. Finally, placing the buffers in the third step also takes linear time. Thus, the encoder takes $O(n_{\text{out}})$ time.

Now, we analyze the complexity of the decoding algorithm. In the trace-alignment step, Steps 1a and 1b which search for buffers are performed in linear time. In Step 1c, we run the decoder of C_S on at most n_{out} symbols and thus this is done in $O(n_{\text{out}})$ time. In Steps 1d and 1e we run the matching algorithm from Lemma 17 which runs in time $O(n_{\text{out}}^2)$ and then simply reposition the output. Thus, the trace-alignment step takes $O(n_{\text{out}}^2)$ time. Then, Step 2 which decodes every trace using the decoding algorithm of C_R (again, recall that n_R is constant with respect to n_{out}), takes $O(n_{\text{out}})$ time. Finally, the decoding of the corrupted outer codeword in Step 3 takes $O(n_{\text{out}})$ according to Proposition 1. We conclude that the decoder takes $O(n_{\text{out}}^2)$ time. □

Remark 11. We emphasize here the order by which the parameters are set in order to make sure that there are no circular dependencies. First, the channel parameters are γ, M, T, and μ. We let ε be the desired gap to capacity we want to achieve. This sets the parameters $\varepsilon_S, \varepsilon_R, \xi, \nu, \eta, \delta_{\text{out}}$, and γ. Now, we choose large enough n_R so that all the constraints hold (those in the Proposition 2 and those imposed by the Corollary 2 and the construction).

We emphasize that all the parameters mentioned are independent of n_{out}.

7 Lower bounds on the capacity of a threshold deletion channel

In this section, we consider a simple example of a runlength-dependent channel and compute lower bounds on its capacity. The channel is defined next.

Definition 17. *Let* BDC-Thr(τ, d) *be a runlength-dependent deletion channel that acts as follows on every input run of length ℓ. If $\ell < \tau$, the channel leaves this runs as is. Otherwise (if $\ell \geq \tau$), every bit of this run is deleted independently with probability d.*

As discussed in Section 1, the motivation to define and study such a channel comes from error patterns observed in DNA-based storage systems. We provide two lower bounds on the capacity of BDC-Thr(τ, d), which we plot for the special cases $\tau = 2, 3$. We emphasize here that since this channel is a runlength-dependent deletion channel and complies with Definition 11, both lower bounds can be turned into explicit and efficient codes using Theorem 11.

7.1 First lower bound

Here we closely follow the approach of Diggavi and Grossglauser [DG06] and Drinea and Mitzenmacher [DM06] and provide a simple lower bound on the capacity of BDC-Thr(τ, d). We prove the following.

Theorem 15. *Let $d \in [0,1]$ and $\tau \in \mathbb{N}$ be such that $d \cdot \frac{\tau+1}{2^\tau} \leq 1/2$. The capacity of* BDC-Thr(τ, d) *is at least* $1 - h\big(d \cdot \frac{\tau+1}{2^\tau}\big)$.

The following proposition states that, with high probability, the fraction of bits that belong to runs of length $\geq \tau$ in a uniform random binary string is close to $\frac{\tau+1}{2^\tau}$.

Proposition 4. *Let X be a uniformly random binary string in $\{0,1\}^N$. Then, with probability $1 - \exp(-\Theta(N^{1/3}))$, the number of bits that belong to runs of length at least t is at most $(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3})) \cdot N$.*

Proof. We will view a random string X as being generated using a geometric distribution with parameter $1/2$ in the following way. Let $b_0 = \text{Ber}(1/2)$ (where Ber stands for the Bernoulli distribution). Then, for each $i = 1, 2, \ldots$ let r_i be sampled from Geom$(1/2)$ denote the length of the i-th run in X where the symbol of this run, b_i, is $\bar{b}_{i-1}$ (the opposite bit). We truncate this process when the resulting string is of length N (possibly trimming the last run). The expected number of runs is $N/\mathbb{E}[\text{Geom}(1/2)] = N/2$. Observe also that the expected number of runs of length i is $2^{-i} \cdot N/2$. We will denote by $r(X)$ the number of runs in X.

Now, the expected number of runs of length $< \tau$ is

$$\begin{aligned}
\frac{N}{2}\sum_{i=1}^{\tau-1}\frac{i}{2^i} &= \frac{N}{4}\sum_{i=1}^{\tau-1}\frac{i}{2^{i-1}} \\
&= \frac{N}{4}\left[\left(\sum_{i=1}^{\tau-1} x^i\right)'\right]_{x=\frac{1}{2}} \\
&= \frac{N}{4}\left[\left(\frac{1-x^\tau}{1-x}\right)'\right]_{x=\frac{1}{2}} \\
&= \frac{N}{4}\left[\frac{-\tau x^{\tau-1}(1-x)+(1-x^\tau)}{(1-x)^2}\right]_{x=\frac{1}{2}} \\
&= N\big(1-2^{-\tau}(\tau+1)\big)\,. \qquad (19)
\end{aligned}$$

Next, we will show that with probability $1-\exp(-\Theta(N^{1/3}))$, the length of Y, the received codeword, is at least $N(1-2^{-\tau}(\tau+1)-\varepsilon)$ for any constant ε. Indeed, first observe that with probability at least $1-\exp(-\Theta(N^{1/3})$ the number of runs in X is in the interval $[(1-\Theta(N^{-1/3}))N/2, (1+\Theta(N^{-1/3}))N/2]$. The proof of this claim can be found in [DM06, Proposition 1]. For simplicity, we will simply write $(1 \pm \Theta(N^{-1/3}))N/2$ to denote the respective interval.

Now, we claim that, conditioned on $r(X) = (1 \pm \Theta(N^{1/3}))N/2$, with probability at least $1 - \exp(-\Theta(N^{1/3}))$ the number of runs of length i in X is at least $2^{-i}(1-\Theta(N^{-1/3}))^2 N/2$ for all $i \in [\tau-1]$. Indeed, let Z_i be a random variable that corresponds to the number of runs of length i in X. Then, conditioned on $r(X) = (1 \pm \Theta(N^{-1/3}))N/2$, we have

$$\begin{aligned}
\Pr\left[Z_i < 2^{-i}\cdot(1-\Theta(N^{-1/3}))^2\cdot\frac{N}{2}\right] &\leq \Pr\Big[Z_i < (1-\Theta(N^{-1/3})\mathbb{E}[Z_i])\Big] \\
&\leq \exp(-\Theta(N^{1/3})),
\end{aligned}$$

where the last inequality follows by a Chernoff bound. Now, as τ is constant with respect to N, we conclude that, conditioned on $r(X) = (1 \pm \Theta(N^{-1/3}))N/2$, the probability that for all $i \in [\tau-1]$

we have $Z_i \geq 2^{-i}N/2(1-\Theta(N^{-1/3}))^2$ is $1 - t \cdot \exp(-\Theta(N^{1/3})) = 1 - \exp(-\Theta(N^{1/3}))$. In this case, the number of bits in X that belong to runs of length $\geq t$ is at most

$$N - \frac{N}{2}(1-\Theta(N^{-1/3}))^2 \sum_{i=1}^{\tau} \frac{i}{2^i} = N\Big(1 - (1-\Theta(N^{-1/3}))^2(1-2^{-\tau}(\tau+1))\Big)$$
$$\leq N\left(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3})\right),$$

where the first equality follows by simply repeating the computation done above in Equation (19). We conclude that the probability that this does not happen is at most the sum of the following probabilities

- $r(X) \neq (1 \pm \Theta(N^{-1/3}))N/2$. As shown above, this occurs with probability $\exp(-\Theta(N^{1/3}))$.
- $r(X) = (1 \pm \Theta(N^{-1/3}))N/2$ and there exists an $i \in [\tau - 1]$ such that $Z_i < 2^{-i}N/2(1 - \Theta(N^{-1/3}))^2$. As shown above, this occurs with probability $\exp(-\Theta(N^{1/3}))$.

The proposition follows. □

To prove Theorem 15, we will need the following well-known result which gives the exact size of the insertion ball around a string.

Lemma 18 ([Lev01, Equation 24])**.** *Let $y \in \{0,1\}^{|y|}$ be a string. The number of strings $x \in \{0,1\}^n$ that contain y as a subsequence is $\sum_{i=1}^{n-|y|} \binom{n}{i}$.*

We are now ready to prove Theorem 15. The improvement over the bound of $1 - h(d)$ [DG06] for the binary deletion channel comes from the observation that a typical output Y in this channel is of length $\approx (1 - d \cdot \frac{\tau+1}{2^\tau})N$, which is greater than $\approx (1-d)N$. The proof will closely follow the proof of [DG06, Theorem 4.2] and incorporate the necessary modifications.

Proof of Theorem 15. Generate a random codebook with 2^{NR} i.i.d. codewords chosen uniformly from $\{0,1\}^N$. Let X be a transmitted codeword and Y be the output of the channel on X. By Proposition 4, with probability at least $1 - \exp(-\Theta(N^{-1/3}))$, the number of bits that can be deleted in X (i.e., the number of bits that belong to runs of length at least τ) is at most $N(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3}))$. On these bits, the channel acts in an i.i.d. fashion and deletes each bit with probability d. Thus, conditioned on X having at most $N(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3}))$ bits that belong to runs of length at least τ, the probability that more than $(d + \Theta(N^{-1/3})) \cdot N(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3}))$ bits are deleted from X is at most $\exp(-\Theta(N^{1/3}))$.

Thus, conditioned on X having at most $N(\frac{\tau+1}{2^\tau} + \Theta(N^{-1/3}))$ bits that belong to runs of length at least τ, we have that

$$|Y| \geq N - \left(d \cdot \frac{\tau+1}{2^\tau} N + \Theta(N^{2/3})\right) = N\left(1 - d \cdot \frac{\tau+1}{2^\tau} - \Theta(N^{-1/3})\right) \tag{20}$$

with probability at least $1 - \exp(-\Theta(N^{1/3}))$.

Our decoding algorithm is as follows. Upon receiving Y, if $|Y| < N\big(1 - d \cdot \frac{\tau+1}{2^\tau} - \Theta(N^{-1/3})\big)$, we declare "short" error, and denote this event by P_{short}. Otherwise, we check if Y is a subsequence of a single codeword in $\mathcal{C}$. If it is, then it must be X and we declare success. If Y is a subsequence of two or more codewords, we declare a collision error. Denote this event by P_{col}.

We turn to computing the collision error, assuming that the length of Y is at least m where m is the right-hand side of Equation (20). Let X_1 be a transmitted codeword, and let X_2 be a random

string. As in [DG06], we first upper bound the probability that the decoding algorithm declares collision error because of X_2 as

$$\begin{aligned}
P_{\text{col},X_2} &:= \sum_{Y,|Y|\geq m} \Pr[Y \text{ is subsequence of } X_2 \mid X_1] \Pr[Y|X_1] \\
&= \sum_{j=m}^{N} \left[\left(\sum_{i=1}^{N-j} \binom{N}{i} \cdot 2^{-N} \right) \cdot \left(\sum_{Y,|Y|=j} \Pr[Y|X_1] \right) \cdot \Pr[|Y| = j] \right] \\
&\leq 2^{-N} \sum_{j=m}^{N} \sum_{i=1}^{N-j} \binom{N}{i} \\
&\leq 2^{-N} \cdot (N-m+1)^2 \cdot \binom{N}{N-m} \\
&\leq 2^{-N(1-h(\frac{t+1}{2^t}\cdot d+o(1))-o(1))}
\end{aligned}$$

The second equality follows by Lemma 18 and taking into consideration that the insertion ball depends only on the length of Y. The first inequality follows since the sum of all probabilities (conditioned on Y having length at least j) equals 1. The second inequality is due to the assumption that $d \cdot \frac{\tau+1}{2^\tau} < \frac{1}{2}$ which implies that $N - m < N/2$. The last inequality follows by the bound $\binom{N}{\alpha N} < 2^{Nh(\alpha)}$ for $\alpha \leq 1/2$. Thus, we union bound over all $X_2 \in \mathcal{C}$ and get that the $P_{\text{col}} = 2^{NR} P_{\text{col},X_2}$. Thus,

$$\begin{aligned}
P_{\text{e}} &= P_{\text{col}} + P_{\text{short}} \\
&\leq 2^{N(R-(1-h(\frac{\tau+1}{2^\tau}\cdot d+o(1))-o(1)))} + \exp(-\Theta(N^{1/3})) \,.
\end{aligned}$$

As a result, for any constant $\varepsilon > 0$, for $R = 1 - h(\frac{\tau+1}{2^\tau} \cdot d) - \varepsilon$ it holds that $\lim_{N\to\infty} P_e \to 0$. □

7.2 Second lower bound

In this section we provide another lower bound on the capacity of BDC-Thr(τ, d). For this lower bound, we will greedily construct binary codes that can be reliably decoded against an adversary that is very restricted with the deletions that he can apply. We note that this construction is inspired by the binary inner code given in [CS22]. However, the codes we design here are more structured and tailored for this particular channel. For $v_1, \ldots, v_\tau$ such that $\sum_{i=1}^{\tau} v_i = 1$, we define $H((v_1, \ldots, v_\tau)) := \sum_{i=1}^{\tau} -v_i \log v_i$.

We prove the following.

Theorem 16. *Let $d \in (0,1)$, M and τ be positive integers, and $\beta_1, \ldots, \beta_\tau \in [0,1]$ be such that $\sum_{i=1}^{\tau} i\beta_i = 1$. Denote $\beta = \sum_{i=1}^{\tau} \beta_i$. Then, the capacity of* BDC-Thr(τ, d) *is at least*

$$\frac{\beta \cdot H\left(\frac{1}{\beta} \cdot (\beta_1, \ldots, \beta_\tau)\right) - (2\beta_\tau + g(d)) h\left(\frac{g(d)}{2\beta_\tau + g(d)}\right) - h(g(d))}{(1 - \beta_\tau \tau) + \beta_\tau M},$$

where

$$g(d) = 2\tau \cdot d^M + \sum_{i=1}^{\tau} (\tau - i) \binom{M}{i} (1-d)^i d^{M-i}.$$

We start by borrowing a lemma from the pioneering work of Levenshtein [Lev66] who first considered recovering from worst-case insertions and deletions. The lemma gives an upper bound on the deletion ball of a string that depends on the number of runs of the string. Formally, we have the following.

Lemma 19 ([Lev66])**.** *Let s be a string and denote by $r(s)$ the number of runs in s. There are at most*

$$\binom{r(s)+\ell-1}{\ell}$$

different subsequences of s of length $|s|-\ell$.

We start with defining a set of strings from which we shall pick our codewords.

Definition 18. *Let $\beta_1,\ldots,\beta_\tau \in [0,1]$ such that $\sum_{i=1}^{\tau} i\beta_i = 1$. Let $S_{\beta_1,\ldots,\beta_\tau} \subset \{0,1\}^N$ be a set that contains all strings that*

1. *Contain only runs of length $\leq \tau$.*
2. *The number of runs of length $i \in [\tau]$ is exactly $\beta_i m$.*

The next claim gives a lower bound on the size of $S_{\beta_1,\ldots,\beta_\tau}$.

Claim 9. *Given $S_{\beta_1,\ldots,\beta_\tau}$, denote $\beta = \sum_{i=1}^{\tau} \beta_i$. Then,*

$$\frac{1}{N}\log_2 |S_{\beta_1,\ldots,\beta_\tau}| \geq H\left(\frac{1}{\beta}(\beta_1,\ldots,\beta_\tau)\right) - o(1)\ .$$

Proof. Clearly, we have that

$$|S_{\beta_1,\ldots,\beta_\tau}| = \frac{(\beta N)!}{(\beta_1 N)!\cdots(\beta_\tau N)!}\ .$$

Then, the bound follows by using Stirling's approximation (e.g., [MU05, Lemma 7.3]). □

We will construct a code $C \subset S_{\beta_1,\ldots,\beta_\tau}$ that is robust against an adversary that performs a δ fraction of restricted deletions. More formally, we have the following definition.

Definition 19. *We call an adversary a (δ,τ)-*restricted adversary*, if given a string $s = r_1 \circ r_2 \circ \cdots \circ r_{\beta N}$, he is allowed to perform deletions only to runs r_i such that r_i or r_{i-1} are of length τ.*

Our next two claims will compute the size of a "deletion ball" and the size of an "insertion ball" under the (δ,τ)-restricted adversary.

Claim 10. *Let $s \in S_{\beta_1,\ldots,\beta_\tau}$. Then, (δ,τ)-restricted adversary defined in Definition 19 can produce at most*

$$\binom{2\beta_\tau N + \delta N}{\delta N}$$

different subsequences of s.

Proof. The number of runs of length τ is $\beta_\tau N$. Therefore, by Definition 19, the (δ,τ)-restricted adversary can apply deletions only to $2\beta_\tau N$ runs. As in the proof of Lemma 19 in [Lev66], we observe that each subsequence can be described by the number of deletions applied to each run. The proposition follows by the equivalence to the "stars and bars" paradigm. □

To bound the number of different strings $s \in S_{\beta_1,\ldots,\beta_\tau}$ for which the (δ,τ)-restricted adversary can turn s into s', we use a lemma from [GW17] that provides a general upper bound on the number of strings in $\{0,1\}^N$ that contain a particular string $s' \in \{0,1\}^{N-\delta N}$ as a subsequence.

Lemma 20 ([GW17, Lemma 7])**.** *Let $\delta \in (0,1/2)$ and let $s' \in \{0,1\}^{N-\delta N}$. The number of strings $s \in \{0,1\}^N$ that contain s' as a subsequence is at most $\delta N\binom{N}{\delta N}$.*

Since $S_{\beta_1,\ldots,\beta_\tau} \subseteq \{0,1\}^N$, we have the following claim

Claim 11. *Let $s' \in \{0,1\}^{N-\delta N}$ be a string that was obtained the (δ,τ)-restricted adversary that was given in a string in $S_{\beta_1,\ldots,\beta_\tau}$. Then, there are at most $\delta N\binom{N}{\delta N}$ strings $s \in S_{\beta_1,\ldots,\beta_\tau}$ that can be transformed to s' by the (δ,τ)-restricted adversary.*

Combining these two claims, we can claim the existence of a code $C \subseteq S_{\beta_1,\ldots,\beta_\tau}$ with a certain rate that is robust against the (δ,τ)-restricted adversary (defined in Definition 19).

Proposition 5. *Let $\delta \in (0,1)$. Let $\beta_1,\ldots,\beta_\tau \in [0,1]$ be such that $\sum_{i=1}^{\tau} i\beta_i = 1$ and denote $\beta = \sum_{i=1}^{\tau}\beta_i$. There is a code $C \subset S_{\beta_1,\ldots,\beta_\tau}$ with rate*

$$H\left(\frac{1}{\beta}(\beta_1,\ldots,\beta_\tau)\right) - (2\beta_\tau+\delta)h\left(\frac{\delta}{2\beta_\tau+\delta}\right) - h(\delta) - o(1)$$

that is robust against the (δ,τ)-restricted adversary.

Proof. We construct the code greedily. Each time we add a string c from $S_{\beta_1,\ldots,\beta_\tau}$ to our codebook C, we need to discard from $S_{\beta_1,\ldots,\beta_\tau}$ all the strings that are *confusable* with c. That is, all strings in $s \in S_{\beta_1,\ldots,\beta_\tau}$ for which the (δ,τ)-restricted adversary can transform s and c into the same string.

By Claim 10 and Claim 11, each string added to the code is confusable with at most

$$\delta N \cdot \binom{2\beta_2 N + \delta N}{\delta N} \cdot \binom{N}{\delta N}$$

strings in $S_{\beta_1,\ldots,\beta_\tau}$. Then, we have

$$\frac{1}{N}\log_2|C| \geq \frac{1}{N}\log_2|S_{\beta_1,\ldots,\beta_\tau}| - \frac{1}{N}\log_2\left|\delta N \cdot \binom{2\beta_2 N+\delta N}{\delta N}\cdot\binom{N}{\delta N}\right|$$

The proposition follows by Claim 9 and by applying the Stirling approximation. □

We now introduce our construction, which basically takes the code guaranteed by Proposition 5 and blows up the runs of length τ.

Construction 1. *Let $M \geq \tau$ be an integer and let $\mathcal{C}'$ be the code guaranteed by Proposition 5. Define $\mathcal{C}$ to be the code obtained from $\mathcal{C}'$ by transforming each run of length τ to a run of length M. The rate of $\mathcal{C}$ is*

$$\frac{H\left(\frac{1}{\beta}(\beta_1,\ldots,\beta_\tau)\right) - (2\beta_\tau+\delta)h\left(\frac{\delta}{2\beta_\tau+\delta}\right) - h(\delta)}{1+(M-t)\beta_\tau} - o(1)\,.$$

Our next theorem states that this code is robust against the channel BDC-Thr(τ,d). We denote by $\mathrm{Bin}(n,p)$ the binomial distribution.

Theorem 17. *Let M and τ be integers where $M \geq \tau$. Let $\beta_1, \ldots, \beta_\tau \in [0,1]$ be such that $\sum_{i=1}^{\tau} i\beta_i = 1$ and denote $\beta = \sum_{i=1}^{\tau} \beta_i$. Let $\delta \in (0, 1/2)$ and denote by $\mathcal{C}$ the code from Construction 1 when given $\delta, \beta_1, \ldots, \beta_\tau$ and M. Let $Z \sim \mathrm{Bin}(M, 1-d)$ and for every nonnegative integer i, define*

$$P^{(\tau)\to(i)} := \Pr[Z = i]$$

and let

$$\alpha := \beta_\tau \cdot \left(2\tau P^{(\tau)\to(0)} + \sum_{i=1}^{\tau-1} (\tau - i) P^{(\tau)\to(i)} \right). \tag{21}$$

Let $c \in C$ be a codeword and let y be the output of the channel $\mathrm{BDC\text{-}Thr}(\tau, d)$ on c. If $\alpha < \delta$, there exists an algorithm that given y as input outputs c with probability $1 - \exp(-\Theta(n))$.

Proof. To prove the theorem, we will present a decoding algorithm and prove that it succeeds with probability $1 - \exp(-\Omega(N))$. Let C be the code given in Construction 1 and denote by C' the associated code given by Proposition 5. That is, for every $c' \in C'$ we obtain $c \in C$ by turning the τ-runs into M-runs. Assume that c was transmitted, assume that the channel output on c was $\tilde{c}$, and denote by c' the respective codeword in C'. Next, we give the decoding algorithm.

Decoding algorithm.

1. For every run in $\tilde{c}$, if its length is at least τ, decode it as a run of length τ. Denote the output of this step as $\tilde{c}'$ and note that $\tilde{c}'$ contains only runs of length $\leq \tau$.
2. Find a codeword in C such that the (δ, τ)-restricted adversary can produce $\tilde{c}$ from this codeword. Return this codeword.

Recall that the channel can apply deletions only to runs with length at least τ and that we have exactly $\beta_\tau N$ such runs in c (and in c'). Denote these runs in c' by $r_1, \ldots, r_{\beta_\tau N}$. We denote by $Z_i, i \in [\beta_\tau N]$ the random variables that correspond to the number of bits that survived after transmitting the blowed up version r_i, of length M. Clearly, $Z_i \sim \mathrm{Bin}(M, 1-d)$. Define the following random variable for $i \in [\beta_\tau m]$:

$$X_i := \begin{cases} 0, & \text{if } Z_i \geq \tau \\ \tau - Z_i, & \text{if } 0 < Z_i < \tau \\ 2\tau & \text{if } Z_i = 0 \end{cases}.$$

We claim that $\mathrm{ED}(c', \tilde{c}') \leq \sum_{i=1}^{\beta_\tau m} X_i$, and in particular it bounds the number of deletions that were performed to c' in order to obtain $\tilde{c}'$ (recall that c' is the codeword of C'). Indeed,

- If $Z_i \geq \tau$, then in Step 1, the algorithm decodes it as runs of length τ and no deletion occurred in the run r_i.
- If $0 < Z_i < \tau$, then in Step 1, the algorithm decodes it as runs of length Z_i and thus $\tau - Z_i$ deletions happened to r_i.
- If $Z_i = 0$, then this run is completely deleted by the channel. In this case, the run before and the run after merge to a single run of the same symbol. Then, in Step 1, the merged run is decoded to a single run. In this case, the channel caused τ deletions (the run r_i of length τ was completely deleted) and the decoding algorithm caused another at most $\leq \tau$ deletions. More formally, assume that we have the following sequence of runs $r_1 \circ r_2 \circ \cdots \circ r_{2\ell+1} \circ r_{2\ell+2}$ in c' and

that the runs $r_2, r_4, \ldots, r_{2\ell}$ were deleted by the channel and that r_1 and $r_{2\ell+1}$ and $r_{2\ell+2}$ were *not deleted* by the channel. The decoding algorithm sees a long run of the same symbol and decodes it as a single run. Then, in the worst case scenario, the decoding algorithm deletes all the bits that correspond to $r_3, \ldots, r_{2\ell+1}$. Indeed, if the $|r_1| < \tau$ then the length of the decoded merged run is at least r_1, and if $|r_1| = \tau$ the length of the decoded merged run is at least Z_1. In both cases, the number of additional deletions caused by the merge and the decoding algorithm is at most $|r_3| + |r_5| + \cdots + |r_{2\ell+1}| \leq \tau \cdot \ell$. Therefore, in total, we suffered at most $2\tau\ell$ deletions.

We have shown that $\mathrm{ED}(c', \tilde{c}') \leq \sum_{i=1}^{\beta_\tau m} X_i$. We also observe that $\tilde{c}'$ can be obtained from c' by the (δ, τ)-restricted adversary. Indeed, by the cases analyzed above and the definition of the X_is, every deletion is applied to a run of length τ or to the subsequent run. Our next goal is to provide an upper bound for the variable $X := \sum_{i=1}^{\beta_\tau N} X_i$ that holds with high probability. First, observe that

$$\mathbb{E}[X] \leq \left(2\tau P^{(\tau)\to(0)} + \sum_{i=1}^{\tau-1} (\tau - i) P^{(\tau)\to(i)} \right) \cdot \beta_\tau N \ . \tag{22}$$

On the other hand, since the event $0 < Z_i < \tau$ causes exactly $\tau - Z_i$ deletion and the event $Z_i = 0$ causes at least τ deletion, we can also lower bound $\mathbb{E}[X]$ by

$$\mathbb{E}[X] \geq \left(\tau P^{(\tau)\to(0)} + \sum_{i=1}^{\tau-1} (\tau - i) P^{(\tau)\to(i)} \right) \cdot \beta_\tau N \ . \tag{23}$$

Note that the X_is are independent random variables (by the nature of the channel). Also, $0 \leq X_i \leq 2\tau$ and they have finite first and second moments. Thus, we apply Lemma 3 and get that for any $\nu > 0$,

$$\Pr[X > (1+\nu)\mathbb{E}[X]] \leq \exp\left(-\frac{2\nu^2 (\mathbb{E}[X])^2}{\beta_\tau N \cdot 4\tau^2} \right) \leq \exp(-\Omega(N)) \ , \tag{24}$$

where the second inequality follows from the lower bound for $\mathbb{E}[X]$ in Equation (23). We have

$$\begin{aligned} \Pr[\mathrm{ED}(c, \tilde{c}) > \delta N] &\leq \Pr[X > \delta N] \\ &= \Pr\left[X > \left(1 + \frac{\delta - \alpha}{\alpha} \right) \alpha N \right] \\ &\leq \Pr\left[X > \left(1 + \frac{\delta - \alpha}{\alpha} \right) \mathbb{E}[X] \right] \\ &\leq \exp(-\Omega(N)) \ . \end{aligned}$$

The first inequality follows by the claim that X is an upper bound on $\mathrm{ED}(c', \tilde{c}')$. The second inequality follows by the assumption $\mathbb{E}[X] \leq \alpha N$ and the last inequality follows by the assumption that $\delta > \alpha$ and substituting $\nu = (\delta - \alpha)/\alpha$ in Equation (24). To conclude, the probability that $\mathrm{ED}(c', \tilde{c}') \leq \delta N$ is at least $1 - \exp(-\Omega(N))$. Now, since the code C' is robust against the (δ, τ)-restricted adversary, and since we showed that all the deletions performed by the channel or the algorithm are of this kind, the theorem follows. □

7.3 Achievable rates for $\tau = 2$ and $\tau = 3$

We provide plots of the lower bounds on the capacity of the BDC-Thr(τ, d) channel derived in previous sections, where we focused on $\tau = 2$ (Figure 1) and $\tau = 3$ (Figure 2). Observe that both

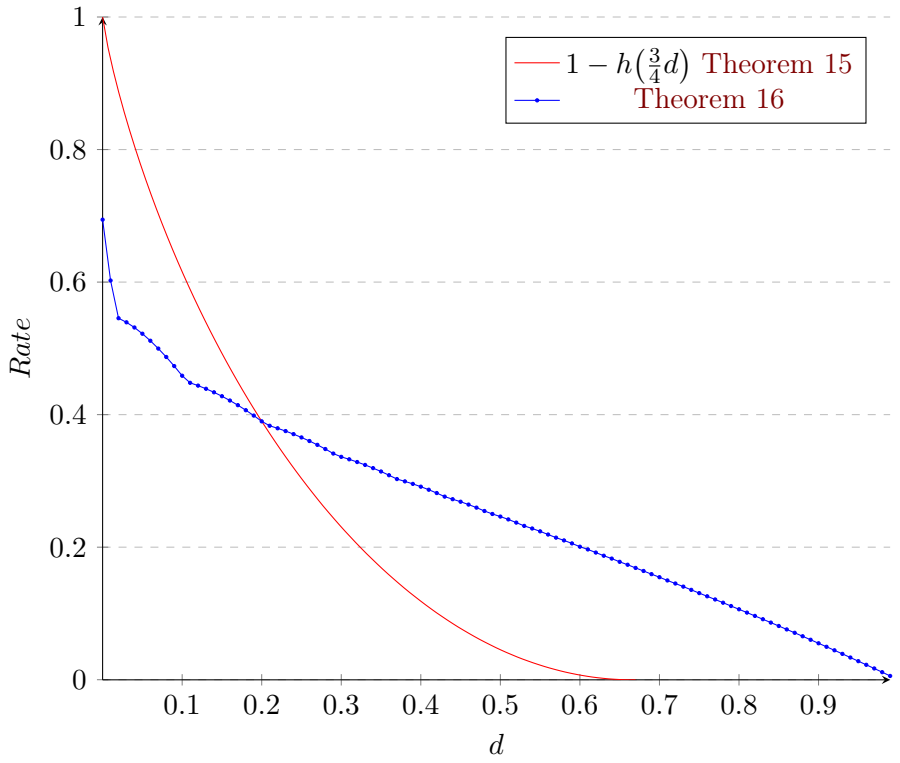

Figure 1: Lower bounds on the capacity of the BDC-Thr(τ, d) for $\tau = 2$.

our lower bounds given by Theorem 15 and Theorem 16 do not provide efficient coding schemes by default. Nevertheless, by invoking Theorem 11, these two lower bounds can be transformed into explicit and efficient codes for the BDC-Thr(τ, d) channel. Also, note that the lower bound implied by Theorem 16 contains several parameters $(\beta_1, \ldots, \beta_\tau, M)$. Therefore, we implemented a greedy search on these parameters to numerically maximize the lower bound and ran it for every deletion probability d, with two-digit precision.

We note that in the case of $\tau = 3$, the transmitter can send strings in $\{0,1\}^N$ that contain only runs of length 1 or runs of length 2 and the receiver receives those strings without any error. Thus, a baseline lower bound for the capacity of BDC-Thr(τ, d) when $\tau = 2$ is $\log_2((1+\sqrt{5})/2) - o(1) \leq 0.6943$. Indeed, the number of such strings of length n is given by the n-th Fibonacci number, defined by the recurrence $F(n) = F(n-1) + F(n-2)$. Our lower bounds improve on this baseline lower bound.

Acknowledgments

The authors thank Elena Grigorescu for many fruitful discussions on this problem.

R. Con was supported in part by the European Union (DiDAX, 101115134). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. J. Ribeiro was supported by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações.

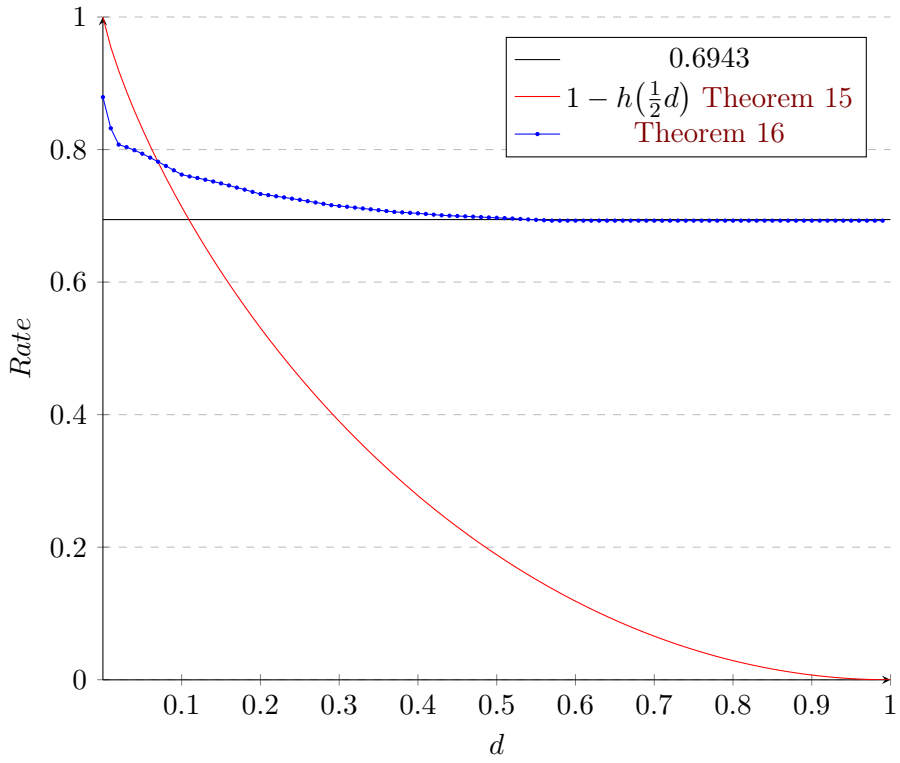

Figure 2: Lower bounds on the capacity of the BDC-Thr(τ, d) for $\tau = 3$.

References

[AT23] Dar Arava and Ido Tal. Stronger polarization for the deletion channel. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 1711–1716. IEEE, 2023.

[BKKM04] Tugkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. Reconstructing strings from random traces. In *SODA*, volume 4, pages 910–918, 2004.

[BLC+16] James Bornholt, Randolph Lopez, Douglas M. Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. A DNA-based archival storage system. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '16, page 637–649, New York, NY, USA, 2016. Association for Computing Machinery.

[BLS20] Joshua Brakensiek, Ray Li, and Bruce Spang. Coded trace reconstruction in a constant number of traces. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–493, 2020.

[CDL+22] Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, and Sandip Sinha. Near-optimal average-case approximate trace reconstruction from few traces. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 779–821. SIAM, 2022.

[CGMR20] Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and João Ribeiro. Coded trace reconstruction. *IEEE Transactions on Information Theory*, 66(10):6084–6103, 2020.

[Cha21] Zachary Chase. New lower bounds for trace reconstruction. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 57(2):627 – 643, 2021.

[CHL+19] Kuan Cheng, Bernhard Haeupler, Xin Li, Amirbehshad Shahrasbi, and Ke Wu. Synchronization strings: Highly efficient deterministic constructions over small alphabets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2185–2204. SIAM, 2019.

[CR21] Mahdi Cheraghchi and João Ribeiro. An overview of capacity results for synchronization channels. *IEEE Transactions on Information Theory*, 67(6):3207–3232, 2021.

[CS22] Roni Con and Amir Shpilka. Improved constructions of coding schemes for the binary deletion channel and the Poisson repeat channel. *IEEE Transactions on Information Theory*, 68(5):2920–2940, 2022.

[dBE52] Nicolaas Govert de Bruijn and Paul Erdős. Some linear and some quadratic recursion formulas. ii. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen: Series A: Mathematical Sciences*, 14:152–163, 1952.

[DG06] Suhas Diggavi and Matthias Grossglauser. On information transmission over a finite buffer channel. *IEEE Transactions on Information Theory*, 52(3):1226–1237, 2006.

[DM06] Eleni Drinea and Michael Mitzenmacher. On lower bounds for the capacity of deletion channels. *IEEE Transactions on Information Theory*, 52(10):4648–4657, 2006.

[Dob67] Roland L. Dobrushin. Shannon's theorems for channels with synchronization errors. *Problemy Peredachi Informatsii*, 3(4):18–36, 1967.

[EZ17] Yaniv Erlich and Dina Zielinski. DNA fountain enables a robust and efficient storage architecture. *Science*, 355(6328):950–954, 2017.

[Fei59] Amiel Feinstein. On the coding theorem and its converse for finite-memory channels. *Information and Control*, 2(1):25–44, 1959.

[FR20] Zoltán Füredi and Imre Z. Ruzsa. Nearly subadditive sequences. *Acta Mathematica Hungarica*, 161(2):401–411, 2020.

[GBC+13] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M LeProust, Botond Sipos, and Ewan Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435):77–80, 2013.

[GHP+15] Robert N Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J Stark. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[GI05] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[GL19] Venkatesan Guruswami and Ray Li. Polynomial time decodable codes for the binary deletion channel. *IEEE Transactions on Information Theory*, 65(4):2171–2178, 2019.

[GW17] Venkatesan Guruswami and Carol Wang. Deletion codes in the high-noise and high-rate regimes. *IEEE Transactions on Information Theory*, 63(4):1961–1970, 2017.

[HMG19] Reinhard Heckel, Gediminas Mikutis, and Robert N. Grass. A characterization of the DNA data storage channel. *Scientific reports*, 9(1):9663, 2019.

[HPP18] Nina Holden, Robin Pemantle, and Yuval Peres. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. In *Conference On Learning Theory*, pages 1799–1840. PMLR, 2018.

[HPPZ20] Nina Holden, Robin Pemantle, Yuval Peres, and Alex Zhai. Subpolynomial trace reconstruction for random strings and arbitrary deletion probability. *Mathematical Statistics and Learning*, 2(3):275–309, 2020.

[HRS19] Bernhard Haeupler, Aviad Rubinstein, and Amirbehshad Shahrasbi. Near-linear time insertion-deletion codes and $(1+\varepsilon)$-approximating edit distance via indexing. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 697–708, 2019.

[HS21a] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings and codes for insertions and deletions—a survey. *IEEE Transactions on Information Theory*, 67(6):3190–3206, 2021.

[HS21b] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: Codes for insertions and deletions approaching the Singleton bound. *J. ACM*, 68(5), September 2021.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on information theory*, 18(5):652–656, 1972.

[Lev66] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*, 1966.

[Lev01] Vladimir I. Levenshtein. Efficient reconstruction of sequences from their subsequences or supersequences. *Journal of Combinatorial Theory, Series A*, 93(2):310–332, 2001.

[LT21] Yonglong Li and Vincent Y. F. Tan. On the capacity of channels with deletions and states. *IEEE Transactions on Information Theory*, 67(5):2663–2679, 2021.

[MBT10] Hugues Mercier, Vijay K. Bhargava, and Vahid Tarokh. A survey of error-correcting codes for channels with symbol synchronization errors. *IEEE Communications Surveys Tutorials*, 12(1):87–96, First Quarter 2010.

[MD24] Ruslan Morozov and Tolga M. Duman. On the capacity of channels with Markov insertions, deletions and substitutions. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 3444–3449, 2024.

[MD25] Ruslan Morozov and Tolga M. Duman. Markov insertion/deletion channels: Information stability and capacity bounds, 2025. https://arxiv.org/abs/2401.16063.

[MDK18] Wei Mao, Suhas N. Diggavi, and Sreeram Kannan. Models and information-theoretic bounds for nanopore sequencing. *IEEE Transactions on Information Theory*, 64(4):3216–3236, 2018.

[Mit09] Michael Mitzenmacher. A survey of results for deletion channels and related synchronization channels. *Probability Surveys*, 6:1–33, 2009.

[MSV24] Brendon McBain, James Saunderson, and Emanuele Viterbo. On noisy duplication channels with Markov sources. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 3438–3443, 2024.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge university press, 2005.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[MVS24] Brendon McBain, Emanuele Viterbo, and James Saunderson. Information rates of the noisy nanopore channel. *IEEE Transactions on Information Theory*, 70(8):5640–5652, 2024.

[OAC+18] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. Random access in large-scale DNA data storage. *Nature biotechnology*, 36(3):242–248, 2018.

[PHJ+20] William H. Press, John A. Hawkins, Stephen K. Jones, Jeffrey M. Schaub, and Ilya J. Finkelstein. HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints. *Proceedings of the National Academy of Sciences*, 117(31):18489–18496, 2020.

[PLW22] Francisco Pernice, Ray Li, and Mary Wootters. Efficient capacity-achieving codes for general repeat channels. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 3097–3102, 2022.

[PT21] Henry D. Pfister and Ido Tal. Polar codes for channels with insertions, deletions, and substitutions. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2554–2559, 2021.

[PW24] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2024. Available at https://people.lids.mit.edu/yp/homepage/data/itbook-export.pdf.

[RRC+13] Michael G. Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J. Lennon, Ryan Hegarty, Chad Nusbaum, and David B. Jaffe. Characterizing and measuring bias in sequence data. *Genome biology*, 14:1–20, 2013.

[Rub22] Ittai Rubinstein. Explicit and efficient construction of nearly optimal rate codes for the binary deletion channel and the Poisson repeat channel. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, pages 105:1–105:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[Rub23] Ittai Rubinstein. Average-case to (shifted) worst-case reduction for the trace reconstruction problem. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 102:1–102:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[SGPY21] Sundara Rajan Srinivasavaradhan, Sivakanth Gopi, Henry D. Pfister, and Sergey Yekhanin. Trellis BMA: Coded trace reconstruction on IDS channels for DNA storage. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2453–2458, 2021.

[TPFV22] Ido Tal, Henry D. Pfister, Arman Fazeli, and Alexander Vardy. Polar codes for the deletion channel: Weak and strong polarization. *IEEE Transactions on Information Theory*, 68(4):2239–2265, 2022.

[TYYM+15] SM Hossein Tabatabaei Yazdi, Yongbo Yuan, Jian Ma, Huimin Zhao, and Olgica Milenkovic. A rewritable, random-access DNA-based storage system. *Scientific reports*, 5(1):14138, 2015.

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[YGM17] SM Hossein Tabatabaei Yazdi, Ryan Gabrys, and Olgica Milenkovic. Portable and error-free DNA-based data storage. *Scientific reports*, 7(1):5011, 2017.