File Updates Under Random/Arbitrary Insertions and Deletions

Qiwen Wang, Sidharth Jaggi, Muriel Médard, *Fellow, IEEE,* Viveck R. Cadambe, *Member, IEEE,* and Moshe Schwartz, *Senior Member, IEEE*

***Abstract*— The problem of one-way file synchronization, henceforth called "file updates", is studied in this paper. Specifically, a client edits a file, where the edits are modeled by insertions and deletions *(InDels)*. An old copy of the file is stored remotely at a data-centre, and is also available to the client. We consider the problem of throughput- and computationally-efficient communication from the client to the data-centre, to enable the data-centre to update its old copy to the newly edited file. Two models for the source files and edit patterns are studied: the random pre-edit sequence left-to-right random InDel (RPES-LtRRID) process, and the arbitrary pre-edit sequence arbitrary InDel (APES-AID) process. In both models, we consider the regime, in which the number of insertions and deletions is a small (but constant) fraction of the length of the original file. For both models, information-theoretic lower bounds on the best possible compression rates that enable file updates are derived (up to first order terms). Conversely, a simple compression algorithm using dynamic programming (DP) and entropy coding (EC), henceforth called DP-EC algorithm, achieves rates that are within constant additive gap (which diminishes as the alphabet size increases) to information-theoretic lower bounds for both models. For the RPES-LtRRID model, a dynamic-programming-run-length-compression (DP-RLC) algorithm is proposed, which achieves a compression rate matching the information-theoretic lower bound up to first order terms. Therefore, when the insertion and deletion probabilities are small (such that first order terms dominate), the achievable rate by DP-RLC is nearly optimal for the RPES-LtRRID model.**

***Index Terms*— Synchronization, insertions, and deletions.**

I. Introduction

AS THE paradigm of cloud storage becomes pervasive, storing and transmitting files and their edited versions consumes a huge amount of resources (storage, bandwidth, computation *etc.*) in client-datacentre communication, and intra-datacentre traffic. If a file is "lightly edited", storing and transmitting the entire new file from clients to servers wastes a significant amount of storage space and bandwidth. Scenarios in which the number of edits is a small fraction of the file length are very common in real-life editing behavior. For example, data-backup systems such as Dropbox and Time Machine keep regular snapshots of users' files. In revision-control software such as CVS, Git and Mercurial, users (programmers) are likely to periodically commit and store their code after a small number of edits. Currently, many online-backup services use *delta encoding* (also known as *delta compression*), and only upload the edited pieces of files [1]–[3]. However, to the best of our knowledge, no existing techniques provide information-theoretically optimal compression guarantees, and this is the principle contribution of our work.

There are potentially many other types of edits besides symbol insertions and deletions, for instance block insertions and deletions, substitutions, transpositions, copy-paste, crop, *etc.* – these and other edit models have been considered in, among other works [4]–[9]. Since these other edit models are in general a combination of symbol insertions and deletions, we focus on the "base case" of symbol insertions and deletions.[1]

A. Our Work/Contributions

In this work, we study the problem of one-way communication of file updates to a data-centre. The client (henceforth called the encoder) has a file $\mathbf{X}$ (henceforth called the pre-edit source sequence *PreESS*) drawn from some distribution, and modifies it through some edits – we shortly describe both the source sequence and the edit patterns in more detail – to generate the new file $\mathbf{Y}$. The encoder has both the old file $\mathbf{X}$ and the edited version of the file $\mathbf{Y}$.[2] The encoder transmits a function of $(\mathbf{X}, \mathbf{Y})$ to the data-centre (henceforth called the decoder). The pre-edit source sequence $\mathbf{X}$ is available at the decoder as side-information. The goal of communication is for the decoder to reconstruct $\mathbf{Y}$. A "good" communication scheme manages to achieve this while requiring minimal communication from the encoder to the decoder.

Manuscript received June 8, 2016; revised April 18, 2017; accepted April 30, 2017. Date of publication May 17, 2017; date of current version September 13, 2017. Q. Wang was supported by the Knut and Alice Wallenberg Foundation. This paper was presented at the 2015 IEEE Information Theory Workshop, and the 2016 IEEE Information Theory Workshop.

Q. Wang is with the Department of Information Science and Engineering, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden (e-mail: qiwenw@kth.se).

S. Jaggi is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: jaggi@ie.cuhk.edu.hk).

M. Médard is with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: medard@mit.edu).

V. R. Cadambe is with the Department of Electrical Engineering, State College, Pennsylvania State University, PA 16802 USA (e-mail: viveck@engr.psu.edu).

M. Schwartz is with the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel (e-mail: schwartz@ee.bgu.ac.il).

Communicated by J. Chen, Associate Editor for Shannon Theory.

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

Digital Object Identifier 10.1109/TIT.2017.2705100

[1] As is common in the literature, we characterize the compression performance of our file update scheme in terms of the number of symbols inserted and deleted. However, explicitly modeling other common user operations can lead to different schemes and possibly better compression performance in other setting.

[2] The encoder may actually ALSO have access to the actual edit pattern, but as we shall see this doesn't necessarily help in our problem.

We now discuss the pre-edit source sequence, and edit patterns. There are many possible combinations of different pre-edit source sequences and edit patterns. Some of those that have been studied in the literature include: arbitrary input sequences [8], [10], random input sequences [9], [11]–[13], (partial) permutations [4], duplications [14]; uniformly random edits [8]–[12], Markov edit processes [13].

In this work, we consider two models. In the Random Pre-Edit Sequence Left-to-Right Random InDel (RPES-LtRRID) model, a file is modeled as a sequence of i.i.d symbols drawn uniformly at random from an alphabet $\mathcal{A}$. The new file is obtained from the old file through a left-to-right random InDel pattern, which is generated from a Markov chain with three states: the "insert symbol" state, the "delete symbol" state, and the "no-operation" state. Briefly speaking, these three states correspond to the cursor moving "from left to right", and at each point, either a uniformly random symbol is inserted, or the symbol at the cursor is deleted, or the cursor jumps ahead without making any change. This model attempts to capture a "one-pass/streaming" edit behavior. There are potentially many ways to model stochastic InDels. Some other stochastic InDel models are discussed in Section V-A. Our results should in general translate over to those models in the regime with small (but constant) fractions of insertions and deletions.

We also study an Arbitrary Pre-Edit Sequence Arbitrary InDel (APES-AID) model. In this model, the old file is modeled as an arbitrary sequence of symbols from the source alphabet $\mathcal{A}$. The new file is generated from the old file through an arbitrary/worst-case InDel sequence, meaning that the number of edit operations is at most a small but possibly constant fraction of the file length n. The edits including insertions and deletions occur in arbitrary positions, and the insertions insert arbitrary symbols from the alphabet $\mathcal{A}$. Both these models are described formally in Section II-B.

In both models, we consider arbitrary alphabet sizes. We first derive information-theoretic lower bounds (in first order terms) on the compression rates needed for the decoder to reconstruct $\mathbf{Y}$ for both models. To do so, for the RPES-LtRRID model we build non-trivially on [15] about the capacity of deletion channels (see Theorem 8), and for the APES-AID model we provide a combinatorial argument (see Theorem 9). We then design computationally-efficient achievability schemes based on dynamic programming and entropy coding (see Theorem 11, Theorem 12 and Theorem 13). The compression rates achieved by the dynamic-programming-entropy-coding (DP-EC) algorithm is within an explicitly computable additive gap to the lower bounds (in first order terms) for almost all alphabet-sizes,[3] and number of edits. In the regime where the number of edits is a small but possibly constant fraction of the length of $\mathbf{X}$ and the alphabet size is large, the gap is small and converges to zero as the alphabet size increases (details in Section IV-B). For the RPES-LtRRID model, the dynamic-programming-run-length-compression (DP-RLC) algorithm achieves a compression rate that matches the lower bound up to first order terms. In the regime when the insertion and deletion probabilities are small such that first order terms dominate, the achievable rate by DP-RLC algorithm for RPES-LtRRID model is nearly optimal.

[3]In the RPES-LtRRID model, we actually have no restriction on the alphabet-size; in the APES-AID model, for technical reasons, our lower bound holds only for alphabets of size at least 3.

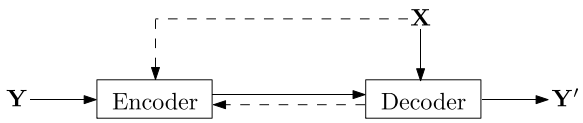

Fig. 1. Synchronization model: To reconstruct the edited version $\mathbf{Y}$ at the decoder using the original version $\mathbf{X}$ as side-information. The transmission between the encoder and the decoder can be either one-way communication or interactive communication, with or without the dashed line from the decoder to the encoder in the figure. The original version $\mathbf{X}$ might be also available at the encoder, shown by the dashed line from $\mathbf{X}$ to the encoder.

B. Related Work

There are two lines of related work – file synchronization, and InDel channels. The problem considered in our work follows the file synchronization problem. However, the techniques for InDel channels and InDel-correcting codes are usually helpful for file synchronization problems.

The general communication model for the synchronization problem is as shown in Fig. 1. The encoder knows the new file $\mathbf{Y}$ and may also know the old file $\mathbf{X}$. The decoder knows the old file $\mathbf{X}$. The purpose is to let the decoder learn $\mathbf{Y}$ (the encoder may or may not learn $\mathbf{X}$) through communication, either one-way or interactively. The one-way synchronization problem, without the dashed line from the decoder to the encoder in Fig. 1, can be interpreted as a problem of source coding with decoder side-information. It is well-known that source coding with decoder side-information is closely related to channel coding [16], [17], and indeed our work on one-way synchronization develops techniques from some prior works in the channel coding literature [15], [18], [19].

Various models of synchronization with edits problems have been considered in the literature – see Table I for a summary. In the early 1990s, Orlitsky had a series of works [20]–[24] on interactive communication for the synchronization of two random variables, providing information-theoretic bounds on the amount of communication needed when different numbers of rounds of communication are allowed. Specifically, in [22], [24] the synchronization of *correlated files* is considered, where the files (modeled by binary sequences) are within a small edit distance from each other. Later, a widely-used file synchronization algorithm called *rsync* was firstly released by Andrew Tridgell and Paul Mackerras in 1996, and was discussed in detail in Tridgell's PhD thesis [25]. In 2001, Orlitsky and Viswanathan [6] considered both communication and computation efficiency of the file synchronization problem. They derived information-theoretic *upper bound* on the number of bits needed to be transmitted when the edit distance between two files is unknown *a priori*. They also proposed a computationally efficient protocol requiring number of bits with about a $2\log n$ multiplicative factor from their upper bound, where n denotes the file length. At the

TABLE I

Related works on file synchronization. The Content of Each Column Is as Follows – [1] Two Aspects of Each Communication Model Are Shown Here. The First Aspect Concerns What Information Is Available to Which Party. Depending on the Specific Model Considered, Either the Original File (the Pre-Edit Source Sequence) $\mathbf{X}$, or the New File (the Post-Edit Source Sequence) $\mathbf{Y}$, or Both may be Available at the Encoder And The Decoder. The Second Aspect Considered Is Whether Interactive/Two-Way Transmissions Between The Encoder and Decoder Are Allowed, or only the Encoder Is Allowed to Transmit (One-Way Communication). [2]The Size of the Source Alphabet – 2 Denotes a Binary Source Alphabet, and $|\mathcal{A}|$ Denotes a General Alphabet. [3]Pre-Edit Source Sequence – 'Arb' Represents an Arbitrary ("worst-case") Pre-Edit Source Sequence; 'Ran' Represents the Pre-Edit Sequences Drawn i.i.d. From The Alphabet. [4]'Arb' Represents the Positions and Contents of the Edits Being Arbitrary; 'Ran' Represents Random Positions and Contents of Edits; 'Markov' Represents the Edit Process Being a Markov Chain. [5]Here 'Ins','Del' and 'Sub' Respectively Represent Insertion, Deletion and Substitution Edit Operations. Reference [6] Proposed Practical Protocol for Various Edit Operations Including Insertion, Deletion, Replacement of a Character, Transposition, Block Deletion and Block Replication. [6]Upper bounds on the Number of Edits in Each Work, as a Function of n (Length of the Pre-Edit Source Sequence $\mathbf{X}$). [7]Whether an Explicit Information-Theoretical Lower Bound Is Presented, Where 'Y' and 'N' Stands for 'Yes' and 'No' Respectively, and '-' for the Case Where the Number of Edits Is $o(n)$ or Within a Factor of Order-Optimal Lower Bounds in Some Two-Way Communication Models. [8]Whether the Algorithm Is Deterministic ('D') or Random ('R'). [9]The Complexity of the Algorithm, as a Function of n (Length of the Pre-Edit Source Sequence $\mathbf{X}$). [10]Whether the Algorithm Has "small" Error – ε-Error, or Zero Error. [11] The Number of Bits Transmitted. In Our Notation, ε Stands for the Fraction (of n) of Insertions, and δ for the Fraction of Deletions. In [8], [10]–[12], the Fractions of Insertions and Deletions Vanish With n, Hence the Corresponding Variables Are Denoted as ϵ_n and δ_n. [12] This Column has Additional Remarks on Specific Works. Reference [24] Investigates Cases Where Edits do Not Introduce New Runs Or Destroy Existing Runs, Where a Run Is a Maximal Block of Contiguous Identical Symbols

Ref	[1]Prob Description	[2] $\mathcal{A}$ size	[3]Pre-ESS	[4]Edits	[5]Edit Ops	[6]#(Edits)	[7] Explicit Info Theo LB	[8]Algo	[9]Comp	[10] P_e	[11] #(bits) transmitted	[12] Remarks
[24]O93	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Arb	Ins,Del	$\mathcal{O}(n)$	Y	R	$\mathcal{O}(e^n)$	0	$(1+\epsilon+\delta)H(\frac{\epsilon+\delta}{1+\epsilon+\delta})n+\mathcal{O}(\log n)$	Upper bound on total # of ins & del, $(\mathbf{X},\mathbf{Y})$ balanced pair *only for edits not changing runs
	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$							D	-		$(\epsilon+\delta)n\log(1+\epsilon+\delta)n+o(n\log n)$*	
	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$							D	$\mathcal{O}(n^2)$		$(\epsilon+\delta)n[\log((1+\epsilon+\delta)n)+2]$	
[6]OV01	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Arb	Ins,Del,etc.	$\mathcal{O}(n)$	N	-	$\mathcal{O}(n\log n)$	ε	$2(\epsilon+\delta)n\log n(2\log n+$ $\log\log n+\log(\epsilon+\delta)-\log P_e)$	Theoretical upper bd $(\epsilon+\delta)n\log n$
[5]CPC+00	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	$\lvert\mathcal{A}\rvert$	Arb	Arb	Ins,Del,Sub	$\mathcal{O}(n)$	Y	R	$\mathcal{O}(n\log n)$	ε	$\Theta((\epsilon+\delta)n\log^2 n)$	LZ distance, block edits
[10]VZR10	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Ran	Ins,Del	$o(\frac{n}{\log n})$	-	D	$\mathcal{O}(n)$	ε	$\mathcal{O}((\epsilon_n+\delta_n)n\log n)$	build on VT code [26]
		$\lvert\mathcal{A}\rvert$									$\mathcal{O}((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)n\log n)$	
[8]VSR13	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Ran	Ins,Del,Sub	$o(n)$	-	D	$\mathcal{O}(n)$	ε	$\Theta((\epsilon_n+\delta_n)^{2/3}n\log n)$	
		$\lvert\mathcal{A}\rvert$									$\Theta((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)^{2/3}n\log n)$	
[11]YD12	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Ran	Ran	Del	$\mathcal{O}(n)$	-	D	$\mathcal{O}(n^4)$	ε	$\mathcal{O}((-\delta\log\delta)n)$	
[12]BD13	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	$\lvert\mathcal{A}\rvert$	Ran	Ran	Ins,Del	$\mathcal{O}(n)$	-	D	-	ε	$\mathcal{O}((-(\epsilon+\delta)\log(\epsilon+\delta))n)$	$\mathcal{A}$ can be non-uniform
[13]MRT11	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$	2	Ran	Markov	Del	$\mathcal{O}(n)$	Y	-	-	ε	$(-\delta\log\delta+\delta(\log 2e-1.29)$ $+\mathcal{O}(\delta^{2-\tau}))n$	In [4] Ran is a special case of Markov
[9]MRT12	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$	2	Ran	Ran	Ins,Del,Sub	$\mathcal{O}(n)$	N	D	$\mathcal{O}(n^2)$	0	$(\lim_{n\to\infty} H(\mathbf{X}\lvert\mathbf{Y})/n+$ $\mathcal{O}(\max(\epsilon,\delta)^{2-\tau}))n$	
This work	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{X}$ $Enc \rightarrow Dec$	$\lvert\mathcal{A}\rvert$	Arb	Arb	Ins,Del	$\mathcal{O}(n)$	Y	D	$\mathcal{O}(n^2)$	0	$(\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $+\log e\cdot\epsilon^2+\mathcal{O}(\epsilon^4))n$	
			Ran	Ran	Ins,Del	$\mathcal{O}(n)$	Y	D	$\mathcal{O}(n^2)$	ε	$(\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $-(\delta+\epsilon)C_{\lvert\mathcal{A}\rvert}+\mathcal{O}(\max(\epsilon,\delta)^{2-\tau})n$	

same time, Cormode *et al.* [5] also proposed computationally efficient schemes to synchronize documents, and showed that the number of bits needed to transmit is of the same order as [6]. They considered different types of metrics for measuring the distance between two files: Hamming distance, edit distance (Levenshtein distance), and Lempel-Ziv (LZ) distance proposed in their work motivated by the Lempel-Ziv data-compression algorithm which also takes block edits into account.

The above-mentioned works consider file synchronization with arbitrary file sequences and arbitrarily distributed edits. Some recent works consider randomly distributed edits performed on arbitrary files or randomly distributed edits on random files, see column 3 and column 4 in Table I. In [8], an interactive synchronization algorithm was introduced which corrects $o(n)$ random insertions, deletions and substitutions under binary source alphabets. Reference [8] quantified the trade-off between the number of rounds and the transmission rate of communication. Its algorithm can deal with bursty insertions and deletions, and can also differentiate substitutions from InDels. This is generalized from the previous work by the authors of [8] which is able to correct $o(n/\log n)$ insertions and deletions [10]. An algorithm was designed in [10] that splits the source sequence into pieces with only a single insertion/deletion or a single burst of insertions/deletions, then uses VT codes, a code by Varshamov and Tenengoltz [26] which is able to correct a single insertion or deletion, to correct edits in those pieces. That algorithm was used as a component in [11] where the synchronization algorithm is able to correct a small constant fraction of deletions over the binary alphabet. In [12] the authors designed synchronization algorithms for insertions and deletions for non-binary non-uniform sources. A one-way file synchronization model was studied in [13] with Markov deletions over the binary alphabet, in which a first-order (in deletion probability) approximation of the optimal rate was derived by using an information-spectrum method. Later, in [9], by allowing both files to be available at the encoder, the authors designed a one-way file synchronization algorithm that can synchronize random insertions, deletions and substitutions over binary alphabets, with communication rate matching the implicit information-theoretic bound $\lim_{n\to\infty}\frac{1}{n}H(\mathbf{Y}|\mathbf{X})$ up to first-order terms.

Since our work follows those works on file synchronization, here we address the differences between our work and some

prior works mentioned in Table I, and we show that our work differs from each of the prior works in significant ways. For instance, in our model the encoder knows both files, hence we design one-way communication protocols, rather than the multi-round protocols required in the models where the encoder and the decoder each has one version of the file as in [5], [6], [8], [10]–[12]. In those works, an information-theoretic lower bound, which is called the genie-aided lower bound in [10] (because it assumes that a genie tells the encoder the old file), is shown to be $\log \binom{n}{s}$ where n is length of the file and s is the total number of insertions and deletions. This genie-aided lower bound is usually used as a benchmark to be compared with in those works with interactive synchronization models, and schemes with achievable rates in the same order as the genie-aided lower bound are considered good (see column 11 of Table I). In our work, we improve the lower bound to be $\log \binom{n}{s} - C_{|\mathcal{A}|} \cdot s$ (see Theorem 8) where $C_{|\mathcal{A}|}$ is a constant that depends only on the alphabet size $|\mathcal{A}|$. This improvement is due to the generalization from the observation in [13] that, one does not need to know the exact locations of the deletions that happened in a block of identical symbols to update the sequence. The one-way communication model studied in [9] and [13] is the closest to the RPES-LtRRID model in our work (Section II-B.1). The main difference is that in [13] an information-theoretic lower bound for a random model with only deletions is derived, while our work derives a lower bound for the case with both deletions and insertions. Allowing insertions at the same time with deletions adds significant difficulty to the analysis. Moreover, our strategies differ. In [13], the authors use an information-spectrum approach. In this work, the strategy is to consider "typical" edits and align the original sequence with the output sequence after applying only those "typical" edits. Besides, our work considers arbitrary alphabet instead of binary alphabet, and also considers an arbitrary model where both file sequences and edits are arbitrary. Furthermore, our DP-EC scheme is "universal" for both random and arbitrary models (Section IV-A). Our DP-RLC algorithm for the RPES-LtRRID model shares the same essense with the algorithm in [9], that is, to encode edits according to the lengths of the runs where they occur. However, our model differs from the one in [9], and we calculate explicitly the achievable rate of DP-RLC scheme, which matches our lower bound in all first order terms.

To address the relation between the InDel channel problem and the file synchronization problem, the InDels in the channel can be modeled the same way as the InDels in the file synchronization problem. The purposes of these two problems are different. In InDel channels, one need to choose the input distribution to maximize the channel capacity $\lim_{n\to\infty} \max_{\Pr(\mathbf{X})} \frac{1}{n} I(\mathbf{X};\mathbf{Y}) = \lim_{n\to\infty} \max_{\Pr(\mathbf{X})} \frac{1}{n} \left(H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})\right)$. In the file synchronization problem, the input distribution is given, e.g., arbitrary input sequences or random input sequences *etc*. The purpose is to find the minimum amount of information the encoder needs to send to the decoder $\min_{\Pr(\mathbf{Y}|\mathbf{X})} H(\mathbf{Y}|\mathbf{X})$, where the probability $\Pr(\mathbf{Y}|\mathbf{X})$ is predetermined by the InDel model.

II. Model

A. Notational Convention

The notational conventions in this work are as follows. Uppercase nonboldface symbols such as X are used to denote random variables; and lowercase nonboldface symbols such as x are used to denote sample values of those random variables. Sequences of random variables or their sample values are denoted by boldface symbols, for example, $\mathbf{X}$ and $\mathbf{x}$ are sequences of random variable X and its sample values x respectively. The length of a sequence $\mathbf{X}$ is denoted by $l(\mathbf{X})$. The length of a subsequence of $\mathbf{X}$ is denoted by $l_{\mathbf{X}}$, to specify that the subsequence comes from $\mathbf{X}$. Sets are denoted by calligraphic symbols, such as $\mathcal{S}$. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. We use $H(\cdot)$ for entropy and conditional entropy of random variables. We denote standard binary entropy by $\mathcal{H}(\cdot)$, that is, $\mathcal{H}(p) = -p\log p - (1-p)\log(1-p)$. We also use $\mathcal{H}(\cdot)$ for entropy of generalized Bernoulli distribution, *e.g.,* $\mathcal{H}(p_1, p_2, 1-p_1-p_2) = -p_1\log p_1 - p_2\log p_2 - (1-p_1-p_2)\log(1-p_1-p_2)$. All logarithms are binary.

B. Source Sequences and Edit Sequences

1) Random Pre-Edit Sequence Left-to-Right Random InDel (RPES-LtRRID) Model: As noted in the introduction, many different stochastic models for source sequences and edits have been considered in the literature. In this work, we are interested in a model where both source sequence and edits are i.i.d. distributed as described below.

- *Pre-Edit Source Sequence (PreESS):* The source initially has a pre-edit source sequence $\bar{\mathbf{X}} = (\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n)$, a length-$n$ sequence of symbols drawn i.i.d. uniformly at random from the source alphabet $\mathcal{A} = \{0, \ldots, a-1\}$. We artificially append an *end of file* symbol $\bar{X}_{n+1} = \text{eof}$ to the end of $\bar{\mathbf{X}}$, to set a stopping rule for the InDel process.
- *InDel Sequences:* the left-to-right edit process is modeled as an i.i.d. sequence drawn from the following three edit operations,
  - insertion $\bar{\imath}$: insert (write) a symbol uniformly drawn from $\mathcal{A}$;
  - deletion $\bar{\Delta}$: read one symbol rightwards in the pre-edit source sequence $\bar{\mathbf{X}}$, and delete the symbol;
  - no-operation $\bar{\eta}$: read one symbol rightwards in the pre-edit source sequence $\bar{\mathbf{X}}$, and do nothing.

  The edit process ends when it reaches the end of file $\bar{X}_{n+1} = \text{eof}$. Let $\bar{O}$ denote a random edit operation, with probability distribution $P(\bar{O} = \bar{\imath}) = \epsilon$, $P(\bar{O} = \bar{\Delta}) = \delta$, and $P(\bar{O} = \bar{\eta}) = 1 - \epsilon - \delta$. The edit operation sequence $\bar{O}^{n+K_\imath}$ is a sequence of i.i.d. edit operations with variable length $n + K_\imath$, where $K_\imath$ denotes the number of insertions. The contents of insertions are drawn uniformly i.i.d. from the source alphabet $\mathcal{A}$, denoted by $\bar{C}^{K_\imath}$. We denote an edit pattern under this model by $\bar{\mathbf{E}} = (\bar{O}^{n+K_\imath}, \bar{C}^{K_\imath})$, and name it a random (ϵ, δ)-InDel sequence.

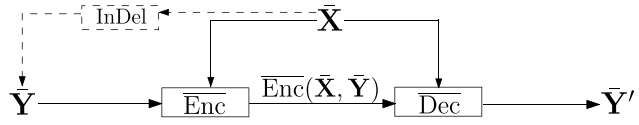

Fig. 2. Communication model: The source has both $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. The sequence $\bar{\mathbf{Y}}$ is obtained from $\bar{\mathbf{X}}$ through random (ϵ, δ)-InDel sequences discussed in Section II-B.1. The source encodes the source sequences $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ into a transmission $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and sends it to the decoder through a noiseless channel. The PreESS $\bar{\mathbf{X}}$ is available at the decoder as side-information. The decoder receives $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, and reconstructs the PosESS $\bar{\mathbf{Y}}'$ from $(\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}})$. Here the bar superscript is used to denote the fact that the source sequences and edit sequences are as described in Section II-B.1 rather than Section II-B.2. The communication model for the APES-AID model discussed in Section II-B.2 is similar, except that $\{\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{Y}}'\}$ is replaced with $\{\mathbf{X}, \mathbf{Y}, \text{Enc}(\mathbf{X}, \mathbf{Y}), \mathbf{Y}'\}$.

- *Post-Edit Source Sequence (PosESS):* The post-edit source sequence $\bar{\mathbf{Y}} = \bar{\mathbf{Y}}(\bar{\mathbf{X}}, \bar{\mathbf{E}})$ is a sequence obtained from $\bar{\mathbf{X}}$ by applying the InDel pattern $\bar{\mathbf{E}} = (\bar{O}^{n+K_\iota}, \bar{C}^{K_\iota})$.
- *Runs:* We define a *run* as a maximal block of consecutively identical symbols [27]. To avoid confusion, we occasionally use $\mathbf{X}$-*run* to emphasize the run is from sequence $\mathbf{X}$.

2) Arbitrary Pre-Edit Sequence Arbitrary InDel (APES-AID) Model:

- *Pre-Edit Source Sequence (PreESS):* The source initially has a pre-edit source sequence $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, an arbitrary length-$n$ sequence over the source alphabet $\mathcal{A}$.
- *InDel Sequences:* The arbitrary (ϵ, δ)-InDel model allows at most ϵn insertions and δn deletions, happening in an arbitrary order. Each insertion inserts an arbitrary symbol from the source alphabet. Let $K_\iota \leq \epsilon n$ and $K_\Delta \leq \delta n$ denote the numbers of insertions and deletions respectively. The edit operation sequence O^{n+K_ι} is a length-$(n + K_\iota)$ sequence with K_ι insertions, K_Δ deletions and $n - K_\Delta$ no-operations. The insertion contents are denoted by a length-K_ι sequence C^{K_ι} of arbitrary symbols from $\mathcal{A}$. We denote an edit pattern under this model by $\mathbf{E} = (O^{n+K_\iota}, C^{K_\iota})$.
- *Post-Edit Source Sequence (PosESS):* A post-edit source sequence, denoted by $\mathbf{Y} = \mathbf{Y}(\mathbf{X}, \mathbf{E})$, is the sequence obtained from $\mathbf{X}$ by applying an arbitrary InDel sequence $\mathbf{E} = (O^{n+K_\iota}, C^{K_\iota})$.

C. Communication Model

The communication system is as shown in Fig. 2. It is a problem of source compression with decoder side information. We define the communication model for both RPES-LtRRID model and APES-AID model. For clarity, we state the communication system for the RPES-LtRRID model, and repeat for the APES-AID model using notation without bars.

In RPES-LtRRID model, the source has both the PreESS $\bar{\mathbf{X}}$ and the PosESS $\bar{\mathbf{Y}}$. The PosESS $\bar{\mathbf{Y}}$ is obtained from the PreESS $\bar{\mathbf{X}}$ through a random (ϵ, δ)-InDel sequence. The edit sequence is not available to the source. The PreESS $\bar{\mathbf{X}}$ and PosESS $\bar{\mathbf{Y}}$ are encoded into a transmission $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ using an *encoder* $\overline{\text{Enc}}$. Taking as inputs the transmission $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and the PreESS $\bar{\mathbf{X}}$, the *decoder* $\overline{\text{Dec}}$ reconstructs the PosESS $\bar{\mathbf{Y}}$ as $\bar{\mathbf{Y}}'$. The code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ comprises the encoder-decoder pair $(\overline{\text{Enc}}, \overline{\text{Dec}})$. The *average rate* $\bar{R}$ of the code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is the average number of bits per source symbol transmitted by the encoder, defined as $\frac{1}{n}\sum_{\bar{\mathbf{X}},\bar{\mathbf{Y}}} p(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \log |\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$. A code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is "$(1 - P_e)$-good" if the *average probability of error*, defined as $\Pr_{\bar{\mathbf{X}},\bar{\mathbf{Y}}}\{(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) : \overline{\text{Dec}}\,(\overline{\text{Enc}}\,(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}}) \neq \bar{\mathbf{Y}}\}$, is less than P_e. A rate $\bar{R}_{\epsilon,\delta}$ is said to be *achievable on average* if for any $P_e > 0$, for all sufficiently large n, there is a code with average rate $\bar{R}_{\epsilon,\delta}$ that is $(1 - P_e)$-good. The infimum (over all such $\bar{\mathcal{C}}_n^{\epsilon,\delta}$) of all achievable rates is called the *optimal average compression rate*, and is denoted $\bar{R}^*_{\epsilon,\delta}$.

In APES-AID model, the source has both the PreESS $\mathbf{X}$ and the PosESS $\mathbf{Y}$. The PosESS $\mathbf{Y}$ is obtained from the PreESS $\mathbf{X}$ through an arbitrary (ϵ, δ)-InDel sequence. Again, the edit sequence is not available to the source. The PreESS $\mathbf{X}$ and PosESS $\mathbf{Y}$ are encoded using an *encoder* Enc into a *transmission* $\text{Enc}(\mathbf{X}, \mathbf{Y})$ from the set $\{1, 2, \ldots, 2^{nR}\}$, where R denotes the *rate* of the encoder Enc. Taking as inputs the transmission $\text{Enc}(\mathbf{X}, \mathbf{Y})$ and the PreESS $\mathbf{X}$, the *decoder* Dec reconstructs the PosESS $\mathbf{Y}$ as $\mathbf{Y}'$. The code $\mathcal{C}_n^{\epsilon,\delta}$ comprises the encoder-decoder pair (Enc, Dec). A code $\mathcal{C}_n^{\epsilon,\delta}$ is said to be "good" if for every pair of $(\mathbf{X}, \mathbf{Y})$, the decoder outputs the correct PosESS, *i.e.* $\mathbf{Y}' = \mathbf{Y}$. A rate $R_{\epsilon,\delta}$ is said to be *achievable* if for all sufficiently large n, there exists a good code with rate at most $R_{\epsilon,\delta}$. The infimum (over all such $\mathcal{C}_n^{\epsilon,\delta}$) of all achievable rates is called the *optimal compression rate*, and is denoted $R^*_{\epsilon,\delta}$.

Remark: For the RPES-LtRRID model, we allow small probability of error, because there are some atypical sequences and edit patterns which can be neglected. But there is no such notion in the APES-AID model. Hence, we require zero-error decodability for the APES-AID model.

III. Lower Bound

A. RPES-LtRRID Model: Proof Roadmap

Since the decoder already has access to the PreESS $\bar{\mathbf{X}}$, the entropy of $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ merely needs to equal $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$, the conditional entropy of the entire PosESS given the PreESS (see the details in Lemma 2). The challenge is to characterize this conditional entropy in single-letter/computable form, rather than as a function dependent on the blocklength n – indeed the same challenge is faced in providing information-theoretic converses for many problems in which information is processed and/or communicated. For scenarios when the relationship from $\bar{\mathbf{X}}$ to $\bar{\mathbf{Y}}$ corresponds to a *memoryless* channel, standard techniques often apply – unfortunately, this is not the case in our file updates problem. We follow the lead of [15], which noted that for InDel processes that are independent of the sequence being edited (as in our case), characterizing $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ is equivalent to characterizing $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. (Recall that $\bar{\mathbf{E}}$ denotes the random variable corresponding to the edit pattern.) In fact $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ can be written as $H(\bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. This is because of the aforementioned independence between $\bar{\mathbf{E}}$ and $\bar{\mathbf{X}}$, and the fact that $\bar{\mathbf{Y}}$ is a deterministic function of $\bar{\mathbf{X}}$ and $\bar{\mathbf{E}}$. We argue this formally in Lemma 3. The entropy of the edit patterns $H(\bar{\mathbf{E}})$ equals exactly to the entropy of specifying the locations of deletions,

the locations of insertions and their contents (this is argued formally in Lemma 4).[4] Since multiple edit patterns can take a PreESS $\bar{\mathbf{X}}$ to the same PosESS $\bar{\mathbf{Y}}$, the term $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ corresponds to the uncertainty in the edit pattern given both $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. The intuition is that disambiguating this uncertainty is useless for the problem of file updating, hence this quantity is called "nature's secret" in [13]. For instance, given $\bar{\mathbf{X}} = 00000$ and $\bar{\mathbf{Y}} = 000$, the decoder does not know, nor does it need to know, which specific pattern of two deletions converted $\bar{\mathbf{X}}$ to $\bar{\mathbf{Y}}$; all the encoder needs to communicate to the decoder is that there were two deletions. In general, if a symbol is deleted from a run or the same symbol generating a run is inserted in the run (edits that shorten or lengthen runs in $\bar{\mathbf{X}}$), the encoder does not need to specify to the decoder the exact locations of deletions or insertions in $\bar{\mathbf{X}}$-runs.

However, characterizing $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ is still a non-trivial task, since it corresponds to an entropic quantity of variables with long dependance. One challenge is that it is hard to align $\bar{\mathbf{X}}$-runs and $\bar{\mathbf{Y}}$-runs. In other words, it is in general difficult to tell which run(s) in $\bar{\mathbf{X}}$ lead to a specific run in $\bar{\mathbf{Y}}$. We call this run(s) in $\bar{\mathbf{X}}$ the *parent run(s)* of the run in $\bar{\mathbf{Y}}$ [15]. We develop the approach in [15]:

- We first carefully "perturb" the original edit pattern $\bar{\mathbf{E}}$ to a *typicalized edit pattern* $\hat{\mathbf{E}}$ (described in details in Definition 1).
- We compute the *typicalized PosESS* $\hat{\mathbf{Y}}$ corresponding to operating the typicalized edit pattern $\hat{\mathbf{E}}$ on the PreESS $\bar{\mathbf{X}}$ (see Definition 2 for details).
- We show via non-trivial case analysis and Lemma 5 that with a "small amount" ($\mathcal{O}(\max(\epsilon, \delta)^2\ n)$ bits) of additional information, $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ can be aligned.
- We show two implications of the above alignment: Lemma 6 provides a bound on $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$, named *typicalized nature's secret*, which is the uncertainty of the typicalized edit pattern given PreESS $\bar{\mathbf{X}}$ and typicalized PosESS $\hat{\mathbf{Y}}$ (Definition 2); Lemma 7 shows that $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ is close to $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$.

Pulling together the implications of the steps above enables us to characterize $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$, up to first order in ϵ and δ. We summarize the steps of our proof in Fig. 3.

One major difference between our work and the analysis in [15] is that since we consider both insertions and deletions, our case-analysis is significantly more intricate. Another difference is that we explicitly characterize our bounds for sequences over all (finite) alphabet sizes, whereas [15] concerned itself only with binary sequences. Also, besides the difference in models and techniques, the underlying motivation differs. The authors of [15] focused on characterizing the capacity of deletion channels, and hence they need to optimize over all statistics of the channel input. On the other hand we focus on the file updates problem, and hence we have no channel input but source sequences drawn according to source statistics.

[4]Recall in our left-to-right InDel model a symbol that is inserted will not be deleted. Even in other models, the reduction in the entropy of $\bar{\mathbf{E}}$ due to interaction of insertions and deletions would be a multiplicative factor of $\epsilon \times \delta$, which is a higher-order term we do not focus on in this work in the regime of small ϵ,δ.

B. Converse (Lemmas 2-4)

Recall in the InDel model (described in Section II-B.1), the total number of deletions and no-operations equals n, with probability of an edit to be a deletion and to be a no-operation (conditioning on that the edit is not an insertion) equals $\frac{\delta}{1-\epsilon}$ and $\frac{1-\epsilon-\delta}{1-\epsilon}$ respectively. Hence, the total number of deletions K_Δ follows a binomial distribution $B(n, \frac{\delta}{1-\epsilon})$ with mean $\frac{\delta}{1-\epsilon}n$. Recall that in our model we allow insertions in front of the first symbol and after the last symbol – this is the reason why the index of number of insertions K_ι is parametrized by $(n+1)$ rather than n in the following. The distribution of the number of insertions in the beginning of the InDel process and after each deletion or no-operation is $\text{Geo}_0(1-\epsilon)$, the geometric distribution on the support of $\{0, 1, 2, \dots\}$ with parameter $(1-\epsilon)$ [28]. The InDel process stops when the total number of deletions and no-operations is n. Hence, K_ι is the sum of $n+1$ i.i.d. random variables whose distributions follow $\text{Geo}_0(1-\epsilon)$. On the other hand, K_ι is the number of insertions with probability ϵ until $n+1$ deletions/no-operations occur, which follows a negative binomial distribution $\text{NB}(n+1; \epsilon)$ with mean $(n+1)\frac{\epsilon}{1-\epsilon}$ [28].

Throughout this section, because we deal with sequences with random lengths, we use Theorem 3 in [29] multiple times. We restate the theorem here as a preliminary for our later proofs.

Theorem 1 [29]: [Theorem 3 (Determined Stopping Time)] A stopping time N is said to be a determined stopping time for the i.i.d. sequence $X_1, X_2, \dots$ if the event $\{N = n\} \in \sigma(X_1, X_2, \dots, X_n)$ for all $n = 1, 2, \dots$, where $\sigma(X_1, X_2, \dots, X_n)$ is the σ-field generated by $X_1, X_2, \dots, X_n$. Then, for a determined stopping time N,

$$H(X^N) = E[N]H(X_1),$$

where $X^N \in \mathcal{A}^*$ denotes the randomly stopped sequence.

Lemma 2 (Converse): For the RPES-LtRRID model, any achievable rate $\bar{R}_{\epsilon,\delta}$ is bounded from below by $\lim_{n\to\infty} \frac{1}{n} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$.

Proof: We first prove a modified version of the conventional Fano's inequality $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}') \le 1 + P_e \log|\bar{\mathcal{Y}}|$, where $\bar{\mathcal{Y}}$ denotes the support of $\bar{\mathbf{Y}}$. Because in our model with unbounded number of insertions, the length of $\bar{\mathbf{Y}}$ can be arbitrarily large as the block-length n grows without bound. Hence, the upper bound on the term $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}', \bar{\mathbf{Y}}' \neq \bar{\mathbf{Y}}) \le \log|\bar{\mathcal{Y}}|$ for proving the conventional Fano's inequality does not work in our problem. We modify Fano's inequality by bounding the term with $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}', \bar{\mathbf{Y}}' \neq \bar{\mathbf{Y}}) \le H(\bar{\mathbf{Y}})$. The PosESS $\bar{\mathbf{Y}}$ is a sequence of symbols drawn uniformly i.i.d. from $\mathcal{A}$, where its length $(n - K_\Delta + K_\iota)$ is a determined stopping time for $\bar{\mathbf{Y}}$. Hence by Theorem 1, $H(\bar{\mathbf{Y}}) = (n - E[K_\Delta] + E[K_\iota])\log|\mathcal{A}| = \left(\frac{1-\delta}{1-\epsilon}n + \frac{\epsilon}{1-\epsilon}\right)\log|\mathcal{A}|$. Hence, our modified Fano's inequality follows as in (1) below. There exists a σ_n such that $\sigma_n \to 0$ as $n \to \infty$, and

$$H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \le 1 + P_e\left(\frac{1-\delta}{1-\epsilon}n + \frac{\epsilon}{1-\epsilon}\right)\log|\mathcal{A}| \le n\sigma_n, \tag{1}$$

where P_e is the average probability of error (recall

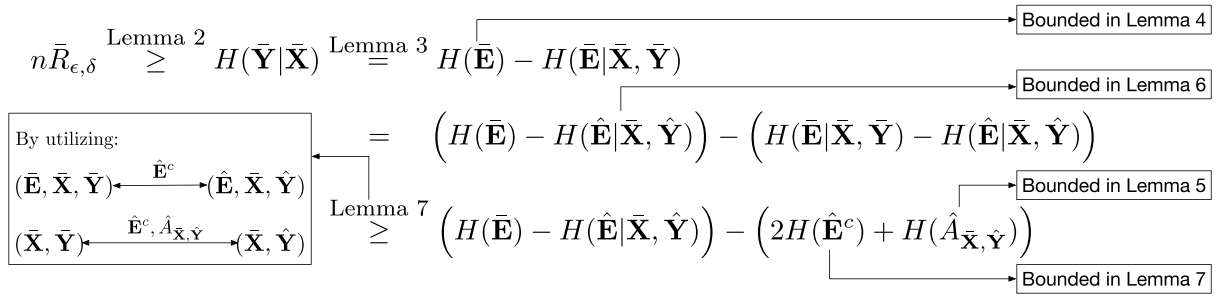

Fig. 3. Flowchart of the proof: The natural lower bound of the amount of information that the encoder needs to send to the decoder is given by the conditional entropy $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$, which we show in Lemma 3 equals to the amount of information to describe the edit pattern $H(\bar{\mathbf{E}})$ subtracted by an amount called "nature's secret" $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. We characterize $H(\bar{\mathbf{E}})$ in Lemma 4. These are included in Section III-B. To characterize nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, we perturb the edit pattern $\bar{\mathbf{E}}$ to a typicalized edit pattern $\hat{\mathbf{E}}$, and introduce some relating concepts in Section III-C. We show in Lemma 7 that nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ is within at most an order $\mathcal{O}(\max(\epsilon, \delta)^2)n$ gap from the typicalized nature's secret $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$, which we characterize in Lemma 6. Both Lemma 6 and Lemma 7 involve a term $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$, which we introduce in Section III-C and quantify its uncertainty in Lemma 5. The lower bound hence follows directly. These are included in Section III-D.

Section II-C). Because we require P_e goes to zero as $n \to \infty$, σ_n should also go to zero as $n \to \infty$.

We have the following chain of inequalities,

$$\begin{aligned} n\bar{R}_{\epsilon,\delta} &\geq H(\overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\geq H(\overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|\bar{\mathbf{X}}) \\ &= H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) + H(\overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\overset{(a)}{=} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) - H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\overset{(b)}{\geq} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) - n\sigma_n, \end{aligned} \tag{2}$$

where equality (a) holds since $\overline{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ is a deterministic function of $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. Inequality (b) follows from our modified Fano's inequality (1).

Dividing both sides of (2) by n deduces our converse. □

Lemma 3: The conditional entropy $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ equals the entropy of the edit pattern $H(\bar{\mathbf{E}})$, less "nature's secret" $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, i.e., $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) = H(\bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$.

Proof:

$$\begin{aligned} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) &\overset{(a)}{=} H(\bar{\mathbf{E}}|\bar{\mathbf{X}}) + H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \\ &\overset{(b)}{=} H(\bar{\mathbf{E}}) + H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \\ &\overset{(c)}{=} H(\bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \end{aligned}$$

where equality (a) is by the Chain Rule; (b) is because the edit pattern $\bar{\mathbf{E}}$ is independent of the PreESS $\bar{\mathbf{X}}$; and (c) is because the PosESS $\bar{\mathbf{Y}}$ is a deterministic function of $(\bar{\mathbf{X}}, \bar{\mathbf{E}})$. □

Lemma 4: The entropy of the edit pattern equals the entropy of specifying the locations of deletions, the locations of insertions and the contents of insertions, specifically, $\lim_{n\to\infty} \frac{1}{n} H(\bar{\mathbf{E}}) \geq \mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon \log|\mathcal{A}| + 2\min(\epsilon, \delta)^{2-\tau} + \mathcal{O}(\max(\epsilon, \delta)^2)$, for some $\tau \in (0, 1)$.

Proof: The probabilistic model of the edit process is well-defined, hence the entropy of it can be calculated. We use Taylor series expansion to obtain the result up to second order in ϵ and δ. The details are in Appendix A. □

C. Typicalized Edit Patterns, Local and Global Alignments

As discussed in Section III-A and Fig. 3, the next quantity we need to bound is the nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. However, this quantity is in general difficult to calculate because $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ are difficult to align. Hence, we perturb the edit pattern $\bar{\mathbf{E}}$ to a typicalized edit pattern $\hat{\mathbf{E}}$, for which an analogue of nature's secret $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ can be calculated (see Lemma 6 for details). We now formally define the typicalized edit pattern $\hat{\mathbf{E}}$ and some sequences that depend on $\hat{\mathbf{E}}$:

Definition 1 (Typicalized Edit Pattern): The typicalized edit pattern $\hat{\mathbf{E}}$ is determined from $(\bar{\mathbf{X}}, \bar{\mathbf{E}})$ by choosing a subset of edits in the original edit pattern $\bar{\mathbf{E}}$ in the following way. The extended run [15] of a run in $\bar{\mathbf{X}}$ includes the run and its two neighbouring symbols, one on each side. Given $(\bar{\mathbf{X}}, \bar{\mathbf{E}})$, for all $\bar{\mathbf{X}}$-runs, count the number of edits per extended run.[5] *If there is no more than one edit in the extended run, the edit pattern in this run is set to be the same in the typicalized edit pattern. If there is more than one edit in the extended run, the typicalized edit pattern $\hat{\mathbf{E}}$ has no edits in that run, that is, the $\bar{\mathbf{X}}$-run and the corresponding $\hat{\mathbf{Y}}$-run are identical.*

Remark: Whether to eliminate the deletions of neighbouring symbols or not is decided by checking the extended runs of the runs they belong to. For example, for $\bar{\mathbf{E}}$: 0/111/223, there are two edits in the extended run 01112 of the second run 111, hence the edit in the second run – the deletion of the left-most 1 – is eliminated in $\hat{\mathbf{E}}$. The right-neighbour 2 of the run 111 belongs to the third run 22, whose extended run 1223 contains only one edit. Hence, the deletion of the right-neighbour 2 of the run 111 is not eliminated in $\hat{\mathbf{E}}$. The typicalized edit pattern in this example is $\hat{\mathbf{E}}$: 0111/223.

Denote the numbers of insertions and deletions in $\hat{\mathbf{E}}$ by $\hat{K}_\iota$ and $\hat{K}_\Delta$ respectively. Since in our model the way we define edit patterns ensures that the sum of the number of deletions and no-operations in any edit pattern (including typicalized edit patterns) always equals exactly n, the length of $\hat{\mathbf{E}}$ equals $n + \hat{K}_\iota$.

[5]Deletion of any symbol in the extended run (including deletion of either of the two symbols neighbouring the $\bar{\mathbf{X}}$-run) adds one to the count. Insertion of a symbol adds one to the count only if the insertion happens to the right of the left-neighbour of the $\bar{\mathbf{X}}$-run, and to the left of the right-neighbour of the $\bar{\mathbf{X}}$-run.

														length
$\bar{\mathbf{X}}$	0	0	0	1	1	1	1	2	2	3	2	3	3	$n = 13$
actual edits	0	$0^{\downarrow 1}$	0	1	$\not{1}^{\downarrow 4}$	$\not{1}$	1	$\not{2}$	2	3	2	$3^{\downarrow 3}$	3	
$\bar{\mathbf{E}}$	$\bar{\eta}$	$\bar{\eta}\,\bar{\iota}_1$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\Delta}\,\bar{\iota}_4$	$\bar{\Delta}$	$\bar{\eta}$	$\bar{\Delta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}\,\bar{\iota}_3$	$\bar{\eta}$	$n + K_\iota = 16$
$\bar{\mathbf{Y}}$	0	0 1	0	1	4		1		2	3	2	3 3	3	$n - K_\Delta + K_\iota = 13$
$\hat{\mathbf{E}}^c$	$-$	$-\,-$	$-$	$-$	$\underline{\bar{\Delta}}\,\underline{\bar{\iota}}_4$	$\underline{\bar{\Delta}}$	$-$	$-$	$-$	$-$	$-$	$-\,-$	$-$	$n + K_\iota = 16$
$\hat{\mathbf{E}}$	$\bar{\eta}$	$\bar{\eta}\,\bar{\iota}_1$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\Delta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}$	$\bar{\eta}\,\bar{\iota}_3$	$\bar{\eta}$	$n + \hat{K}_\iota = 15$
$\hat{\mathbf{Y}}$	0	0 1	0	1	1	1	1		2	3	2	3 3	3	$n - \hat{K}_\Delta + \hat{K}_\iota = 14$

Fig. 4. An example of source sequences, edit sequences and typicalized sequences.

Definition 2 (Typicalized PosESS): *The typicalized PosESS* $\hat{\mathbf{Y}}$ *is the post-edit source sequence obtained by operating the typicalized edit pattern* $\hat{\mathbf{E}}$ *on the PreESS* $\bar{\mathbf{X}}$*. The length of* $\hat{\mathbf{Y}}$ *equals* $n - \hat{K}_\Delta + \hat{K}_\iota$*.*

Definition 3 (Complement of typicalized edit pattern): *The complement of typicalized edit pattern describes the eliminated edits. Let '$-$' denote no elimination, and underlined characters '$\underline{\bar{\iota}}$' and '$\underline{\bar{\Delta}}$' denote eliminations of insertions and deletions respectively. A length-$(n + K_\iota - \hat{K}_\iota)$ sequence $\underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}$ over $\{-, \underline{\bar{\iota}}, \underline{\bar{\Delta}}\}$ specifies the eliminated edits. A length-$(K_\iota - \hat{K}_\iota)$ sequence $\underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)}$ over the source alphabet $\mathcal{A}$ specifies the contents of insertions that are eliminated. Let $\hat{\mathbf{E}}^c = (\underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}, \underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)})$ denote the complement of typicalized edit pattern.*

In Fig. 4 we show an example of all the sequences we define. In the example, the first row shows a length $n = 13$ PreESS $\bar{\mathbf{X}}$ sequence over the alphabet $\{0, 1, 2, 3, 4, 5\}$. The second row shows in shorthand edits performed on $\bar{\mathbf{X}}$, the notation of which is more visual and appears throughout the section in examples. A superscript of a downarrow together with a number denotes an insertion of the corresponding source symbol, *e.g.,* $^{\downarrow 1}$ denotes an insertion of 1. A line through a symbol denotes a deletion of that symbol, *e.g.,* $\not{1}$ denotes a deletion of 1. The third row shows the corresponding edit pattern $\bar{\mathbf{E}}$. As defined in the model section, insertions are represented by $\bar{\iota}$, deletions by $\bar{\Delta}$, and no operations by $\bar{\eta}$. Here, for the sake of brevity we abuse notation by representing the contents of insertions as subscripts to the corresponding $\bar{\iota}$, rather than as a separate $\bar{C}^{K_\iota}$. For instance in the example in this figure, the operation of inserting a 4 after the fifth symbol is represented by $\bar{\iota}_4$. Since there are three insertions in the edit sequence, the length of the edit sequence $\bar{\mathbf{E}}$ equals $n + 3 = 16$. The resulting PosESS $\bar{\mathbf{Y}}$ is shown in the fourth row. Note that $\bar{\mathbf{X}}$ has 6 runs – 000, 1111, 22, 3, 2 and 33 (single symbols distinct from their neighbors also count as runs). The corresponding extended runs are respectively 0001, 011112, 1223, 232, 323, and 233. The number of edits in each of these runs is therefore respectively 1, 3, 1, 0, 0, 1, and in the corresponding extended runs is 1, 4, 1, 0, 1, 1. Hence the only edits eliminated from $\bar{\mathbf{E}}$ to get $\hat{\mathbf{E}}$ are the three edits in the second $\bar{\mathbf{X}}$-run (since the corresponding extended $\bar{\mathbf{X}}$-run has 4 edits and by our definition typicalized edit patterns may only have at most one edit per extended run). The complement' of typicalized edit pattern therefore has blanks '$-$' everywhere except in the locations corresponding to the three edits in the second run of $\bar{\mathbf{X}}$, as shown in the fifth row. The sixth row shows the typicalized edit process with all the edit operations present in $\bar{\mathbf{E}}$, except those corresponding to the three in the second run of $\bar{\mathbf{X}}$. Finally, the last row shows the resulting typicalized PosESS $\hat{\mathbf{Y}}$ resulting from operating $\hat{\mathbf{E}}$ on $\bar{\mathbf{X}}$. We shall reuse this example later multiple times to explain different concepts.

In Fig. 5, we represent the dependencies of all the sequences we define above, and some internal random variables we define and use in later proofs.

We recall the definition of parent run and introduce the term *local alignment*, both will appear many times throughout this section.

- Parent run: parent run(s) of a $\bar{\mathbf{Y}}$-run are the run(s) in $\bar{\mathbf{X}}$ which lead to the $\bar{\mathbf{Y}}$-run.
- Local alignment: it refers to identifying the parent $\bar{\mathbf{X}}$-run(s) for a specific $\hat{\mathbf{Y}}$-run.

We first show that $\hat{\mathbf{Y}}$-runs can be mostly aligned to the parent run(s) in $\bar{\mathbf{X}}$. The intuition is that, since $\bar{\mathbf{X}}$-runs undergo at most one edit in the typicalized edit pattern, for any $\hat{\mathbf{Y}}$-run, there are only a few possible cases for its parent run(s), and the corresponding length(s). There are only two events where the cases of the parent run-length intersect, which are named *ambiguous local alignment events* and defined formally in Definition 4. The local alignment might be ambiguous. However, the ambiguity might be resolved later. Because it is possible that for one of the ambiguous local alignments, no typicalized edit pattern (roughly speaking, with no more than one edit per run) can convert $\bar{\mathbf{X}}$ to $\hat{\mathbf{Y}}$ for the remaining runs. Otherwise, both local alignments are possible and results in different *global alignments*, defined formally in Definition 5. Hence, one can align $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ in a left-to-right manner by checking the lengths of $\hat{\mathbf{Y}}$-runs and $\bar{\mathbf{X}}$-runs, with the aid of some extra information indicating which global alignment it is. The extra information $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ is defined in Definition 5. Its entropy is bounded from above in Lemma 5. Example 1 is shown to illustrate both the case where an ambiguous local alignment is resolved by aligning further runs, and the case where an ambiguous local alignment is not resolved and hence leads to two possible global alignments. Once $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ are aligned, the uncertainty of the typicalized edit pattern $\hat{\mathbf{E}}$ only lies in the uncertainty of positions of deletions, and positions of insertions that lengthen runs (inserting the same symbol as

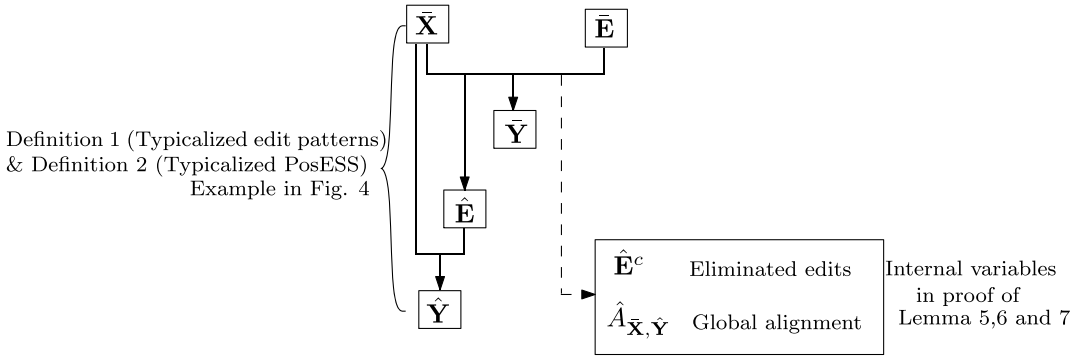

Fig. 5. The dependency of all the sequences and internal random variables for the proofs.

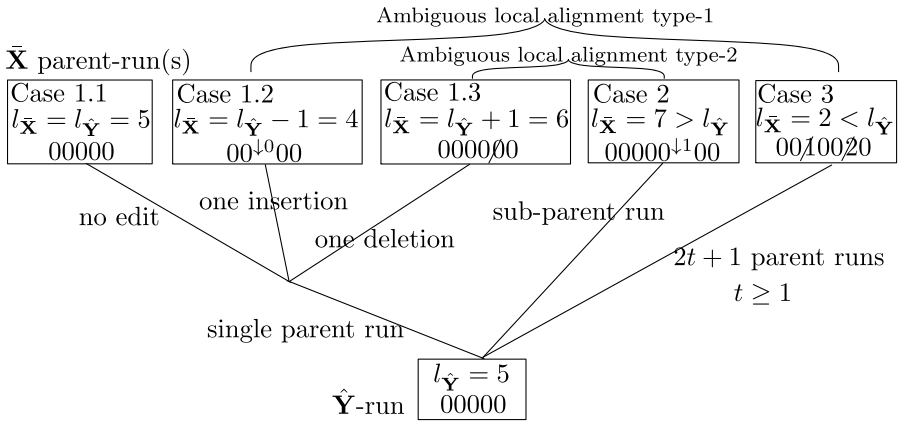

Fig. 6. All possible cases of the parent run(s) in $\bar{\mathbf{X}}$ for a length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run.

those constructing the run). We characterize this uncertainty in Lemma 6.

For a $\hat{\mathbf{Y}}$-run with length denoted by $l_{\hat{\mathbf{Y}}}$, its possible parent run(s) are categorized into the following cases, as shown in Fig. 6 (we give examples corresponding to the $\hat{\mathbf{Y}}$-run being 00000):

- **Case 1 (single parent-run):** The parent $\bar{\mathbf{X}}$-run is a single run with length denoted by $l_{\bar{\mathbf{X}}}$.
  - **Case 1.1 (1-parent-0-edit):** No edit in the parent run, hence $l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}}$. Eg: $00000 \to 00000$.
  - **Case 1.2 (1-parent-1-ins):** One insertion in the parent run, hence $l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1$. Eg: $00^{\downarrow 0}00 \to 00000$.
  - **Case 1.3 (1-parent-1-del):** One deletion in the parent run, hence $l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$. Eg: $000\not{0}0\not{0} \to 00000$.
- **Case 2 (sub-parent):** The parent run is a sub-run of a length-$l_{\bar{\mathbf{X}}}$ run, that is, an insertion of a different symbol within a parent run breaks it into two runs. In this case, $l_{\bar{\mathbf{X}}} > l_{\hat{\mathbf{Y}}}$. Eg: $00000^{\downarrow 1}00 \to 00000100$. The next two runs (1 and 00 in this example) in $\hat{\mathbf{Y}}$ are also aligned to this $\bar{\mathbf{X}}$-run.
- **Case 3 (multi-parent):** There are multiple parent runs. The number of parent runs is odd, denoted by $2t+1$ ($t \geq 1$). Among these $\bar{\mathbf{X}}$-runs, $t+1$ runs (the odd-numbered ones) comprise the same symbol (0, in this example) as the corresponding $\hat{\mathbf{Y}}$-run, with lengths $l_1, \ldots, l_{t+1}$ respectively. Interleaved among these are the even-numbered $\bar{\mathbf{X}}$-runs, comprising of just one symbol each, which must be different from the symbols that comprise $\hat{\mathbf{Y}}$. In this case, all the length-1 even-numbered $\bar{\mathbf{X}}$-runs get deleted and there is no edit in the other $t+1$ odd-numbered $\bar{\mathbf{X}}$-runs (of the same symbol as in the $\hat{\mathbf{Y}}$-run). Hence $l_{\hat{\mathbf{Y}}} = \sum_{j=1}^{t+1} l_j$ and $l_{\bar{\mathbf{X}}} = l_1 < l_{\hat{\mathbf{Y}}}$. Eg: $00\not{1}00\not{2}0 \to 00000$.

Noting the parent run-lengths in all the above cases and examining the run-lengths of $\hat{\mathbf{Y}}$ and $\bar{\mathbf{X}}$ in a left-to-right manner, $\hat{\mathbf{Y}}$-runs can be mostly aligned to $\bar{\mathbf{X}}$-runs, except for the following two ambiguous local alignment events. We show later that with the help of some small amount additional information, $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ can be aligned.

Definition 4 (Ambiguous Local Alignment): In the following two types of events, simply comparing the run-lengths $l_{\bar{\mathbf{X}}}$ and $l_{\hat{\mathbf{Y}}}$ cannot resolve which case the edit is.

- *Ambiguous local alignment type-1 Γ^1 ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1$): Recall in Case 3 ($l_{\bar{\mathbf{X}}} < l_{\hat{\mathbf{Y}}}$), when $t = 1$ and $l_{\bar{\mathbf{X}}} = l_1 = l_{\hat{\mathbf{Y}}} - 1$, $l_2 = 1$, the length of the parent run is the same as in Case 1.2 ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1$). Hence, for a pair of $\bar{\mathbf{X}}$-run and $\hat{\mathbf{Y}}$-run to be aligned, if the length of $\hat{\mathbf{Y}}$-run is $l_{\hat{\mathbf{Y}}}$ and the length of $\bar{\mathbf{X}}$-run equals $l_{\hat{\mathbf{Y}}} - 1$, one cannot tell immediately whether it is Case 1.2 or Case 3.*
- *Ambiguous local alignment type-2 Γ^2 ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$): Recall in Case 2 ($l_{\bar{\mathbf{X}}} > l_{\hat{\mathbf{Y}}}$), when $l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$ and an*

$\bar{\mathbf{X}}$ 0 0 0 1 1 1 1 2 2 3 2 3 3

$\hat{\mathbf{Y}}$ 0 0 1 0 1 1 1 1 2 3 2 3 3 3

Alignment 1: $\not{0}$ 0 0 $1^{\downarrow 0}$ 1 1 1 2 $Error!$

Alignment 2: 0 $0^{\downarrow 1}$ 0 1 1 1 1 $\not{2}$ 2 3 2 $3^{\downarrow 3}$ 3

Alignment 3: 0 $0^{\downarrow 1}$ 0 1 1 1 1 $2^{\downarrow 3}$ 2 3 $\not{2}$ 3 3

Fig. 7. An example with cases where both ambiguous local alignment resolved and unresolved: with Alignment 1, there is no typicalized edit pattern that leads to $\hat{\mathbf{Y}}$, hence the ambiguity in the first half is resolved; both Alignment 2 and Alignment 3 lead to $\hat{\mathbf{Y}}$, hence the ambiguity in the second half remains.

insertion of a different symbol occurs in front[6] *of the last symbol of the* $\bar{\mathbf{X}}$*-run, leading to a length-*$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$*-run, the length of the* $\bar{\mathbf{X}}$*-run is the same as in Case 1.3 (*$l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$*). Hence, for a pair of* $\bar{\mathbf{X}}$*-run and* $\hat{\mathbf{Y}}$*-run to be aligned, if the length of* $\hat{\mathbf{Y}}$*-run is* $l_{\hat{\mathbf{Y}}}$ *and the length of* $\bar{\mathbf{X}}$*-run equals* $l_{\hat{\mathbf{Y}}} + 1$*, one cannot tell immediately whether it is Case 1.3 or Case 2.*

Example 1: The ambiguous local alignments might be resolved when aligning further $\bar{\mathbf{X}}$*-runs and* $\hat{\mathbf{Y}}$*-runs. Not all local ambiguous alignments lead to different global alignments (different alignments of all* $\bar{\mathbf{X}}$*-runs and* $\hat{\mathbf{Y}}$*-runs, see Definition 5 for details). In Fig. 7, there is an ambiguous local alignment type-2 event* Γ^2 *(*$l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$*) in aligning the first* $\bar{\mathbf{X}}$*-run and* $\hat{\mathbf{Y}}$*-run. The first* $\hat{\mathbf{Y}}$*-run (*00*) is of length* 2*, and the first* $\bar{\mathbf{X}}$*-run (*000*) to be aligned with the* $\hat{\mathbf{Y}}$*-run is of length* 3 *– they comprise the same symbol* 0*. The edit in the first* $\bar{\mathbf{X}}$*-run may be Case 1.3 (single-deletion) or Case 2 (single-insertion breaking the* $\bar{\mathbf{X}}$*-run). We therefore examine the next symbols in* $\bar{\mathbf{X}}$ *and* $\hat{\mathbf{Y}}$*. First of all, we examine the next one or two symbols in* $\bar{\mathbf{X}}$ *and* $\hat{\mathbf{Y}}$*, the local ambiguity is still not resolved. Specifically, the symbol after the first* $\hat{\mathbf{Y}}$*-run (*00*) is a* 1*, the same as the symbol after the first* $\bar{\mathbf{X}}$*-run (*000*), which means Case 1.3 (single-deletion) is possible. The second symbol after the* $\hat{\mathbf{Y}}$*-run (*00*) is a* 0*, the same as the symbol the first* $\hat{\mathbf{Y}}$*-run (*00*) comprise, which means Case 2 (single-insertion breaking the* $\bar{\mathbf{X}}$*-run) is possible. However, ambiguity is resolved when aligning the second* $\bar{\mathbf{X}}$*-run to* $\hat{\mathbf{Y}}$*:*

- *Alignment 1: This must mean that a* 0 *was inserted after the first* 1 *in the second* $\bar{\mathbf{X}}$*-run (*1111*), breaking it into two runs of* 1*'s in* $\hat{\mathbf{Y}}$ *separated by a* 0 *(respectively the third to the eighth symbols in* $\hat{\mathbf{Y}}$*). This scenario is shown in the third line of the figure. Since the second* $\bar{\mathbf{X}}$*-run has four* 1*'s, the resulting* $\hat{\mathbf{Y}}$*-run has three more* 1*'s, with no more edits (since it is a typicalized* $\hat{\mathbf{Y}}$*-run). However, there are four* 1*'s in* $\hat{\mathbf{Y}}$ *after the inserted* 0*. Hence, alignment 1 is not possible.*
- *Alignment 2: The first three runs in* $\hat{\mathbf{Y}}$ *(*0010*) are aligned to the first* $\bar{\mathbf{X}}$*-run. The next* $\bar{\mathbf{X}}$*-run and* $\hat{\mathbf{Y}}$*-run to align both have four* 1*'s, hence can be aligned correctly and unambiguously.*

[6]Hereinafter, by an edit occurs in front of a particular symbol in a sequence, it means that the edit occurred immediately after the previous symbol and before this particular symbol.

Starting from the third $\bar{\mathbf{X}}$*-run, the edits in both alignment 2 and alignment 3 convert* $\bar{\mathbf{X}}$ *to* $\hat{\mathbf{Y}}$*. Hence, the local ambiguous alignment is unresolved. The challenge therefore is to characterize the probability of such local ambiguity being globally unresolvable. This is the thrust of Lemma 5.* ■

We formally define the global alignment (we sometimes call it alignment for short) of a pair of PreESS and typicalized PosESS $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$, and also the partial alignment of their subsequences.

Definition 5 (Global Alignment): Denote the number of runs in a typicalized PosESS $\hat{\mathbf{Y}}$ *by* $\rho_{\hat{\mathbf{Y}}}$*. The typicalized PosESS* $\hat{\mathbf{Y}}$ *can then be decomposed into* $\hat{\mathbf{Y}}$*-runs as*

$$\hat{\mathbf{Y}} = \hat{Y}(1)\hat{Y}(2)\dots\hat{Y}(\rho_{\hat{\mathbf{Y}}}).$$

We then divide $\bar{\mathbf{X}}$ *into "segments that leads to corresponding* $\hat{\mathbf{Y}}$*-runs". Denote* $\bar{X}_{\hat{\mathbf{Y}}}(i)$ *to be the segment in* $\bar{\mathbf{X}}$ *that leads to the* i*th* $\hat{\mathbf{Y}}$*-run, for all* $i = 1, 2, \dots, \rho_{\hat{\mathbf{Y}}}$*, hence,*

$$\bar{\mathbf{X}} = \bar{X}_{\hat{\mathbf{Y}}}(1)\bar{X}_{\hat{\mathbf{Y}}}(2)\dots\bar{X}_{\hat{\mathbf{Y}}}(\rho_{\hat{\mathbf{Y}}}).$$

Note that $\bar{X}_{\hat{\mathbf{Y}}}(i)$*'s are in general not runs of* $\bar{\mathbf{X}}$*. Recall in Fig. 6 that the parent run(s) can be one* $\bar{\mathbf{X}}$*-run, a part of* $\bar{\mathbf{X}}$*-run, or multiple* $\bar{\mathbf{X}}$*-runs. To eliminate uncertainty in the way of dividing the segments, for any* $\hat{Y}(i)$ *that is created by insertions, set the corresponding* $\bar{X}_{\hat{\mathbf{Y}}}(i)$ *to be an empty run* ϕ *with length* 0*. For any* $\bar{\mathbf{X}}$*-run that is deleted and the two neighbouring runs of it on both sides are comprised of different symbols, we force the deleted run to join the segment of its right neighbouring run. The alignment of* $\bar{\mathbf{X}}$ *and* $\hat{\mathbf{Y}}$ *is defined by the vector of the lengths of the segments* $\bar{X}_{\hat{\mathbf{Y}}}(i)$*'s,*

$$\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}} \triangleq \left(|\bar{X}_{\hat{\mathbf{Y}}}(1)|, |\bar{X}_{\hat{\mathbf{Y}}}(2)|, \dots, |\bar{X}_{\hat{\mathbf{Y}}}(\rho_{\hat{\mathbf{Y}}})|\right).$$

Definition 6 (Partial Alignment): For the subsequence of a typicalized PosESS $\hat{\mathbf{Y}}$ *consisting of the first* $i_{\hat{\mathbf{Y}}}$ *runs* $\hat{Y}(1)\hat{Y}(2)\dots\hat{Y}(i_{\hat{\mathbf{Y}}})$ *where* $i_{\hat{\mathbf{Y}}} \le \rho_{\hat{\mathbf{Y}}}$*, suppose the segments of* $\bar{\mathbf{X}}$ *that lead to the* $\hat{\mathbf{Y}}$*-runs are* $\bar{X}_{\hat{\mathbf{Y}}}(1)\bar{X}_{\hat{\mathbf{Y}}}(2)\dots\bar{X}_{\hat{\mathbf{Y}}}(i_{\hat{\mathbf{Y}}})$*. The partial alignment of* $\bar{\mathbf{X}}$ *and* $\hat{\mathbf{Y}}$ *up to depth* $i_{\hat{\mathbf{Y}}}$ *is defined by the vector of the lengths of the segments* $\bar{X}_{\hat{\mathbf{Y}}}(i)$*'s,*

$$\hat{A}^{i_{\hat{\mathbf{Y}}}}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}} \triangleq \left(|\bar{X}_{\hat{\mathbf{Y}}}(1)|, |\bar{X}_{\hat{\mathbf{Y}}}(2)|, \dots, |\bar{X}_{\hat{\mathbf{Y}}}(i_{\hat{\mathbf{Y}}})|\right).$$

Definition 7 (Alignment Tree): The alignment tree of a pair of PreESS and typicalized PosESS $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ *is a binary tree with depth* $\rho_{\hat{\mathbf{Y}}}$*. Each path* $P_{\hat{A}}$ *of the tree corresponds to a sample value of the global alignment* $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$*, with the* i*th entry of* $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ *at the node in depth-*i*. Every split (node with two children) in the alignment tree represents an unresolved ambiguous local alignment. Denote the number of splits on a path* $P_{\hat{A}}$ *by* $N_s(P_{\hat{A}})$*. If a particular PreESS* $\bar{\mathbf{x}}$ *and an edit pattern* $\bar{\mathbf{e}}$ *are given, they determine the typicalized edit pattern* $\hat{\mathbf{e}}$*, hence determine the typicalized PosESS* $\hat{\mathbf{y}}$ *and also the global alignment. Hence, we use notation* $N_s(P_{\hat{A}}(\bar{\mathbf{x}}, \bar{\mathbf{e}}))$ *when* $\bar{\mathbf{x}}$ *and* $\bar{\mathbf{e}}$ *are known.*

Remark: Global alignment is a way to classify typicalized edit patterns, hence also original edit patterns. For any pair $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$, denote the set of all typicalized edit patterns $\hat{\mathbf{e}}$ which

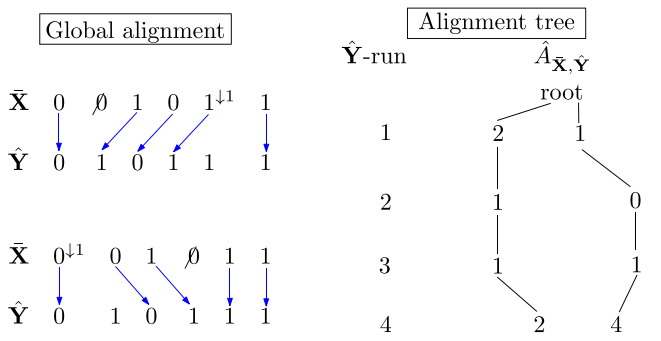

Fig. 8. Example of global alignments and alignment tree.

takes $\bar{\mathbf{x}}$ to $\hat{\mathbf{y}}$ by $\{\hat{\mathbf{e}} : (\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}\}$, where $(\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}$ means that processing $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$ results in $\hat{\mathbf{y}}$. The global alignment $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ is a random variable in a vector form, of lengths of segments in $\bar{\mathbf{x}}$ which lead to the $\hat{\mathbf{y}}$-runs. Let $\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})$ denode the sample space of $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$, and $|\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})|$ denote its size. The set $\{\hat{\mathbf{e}} : (\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}\}$ are classified into $|\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})|$ groups, denoted by $\{\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}\}$ based on the global alignment, where $\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}$ denotes the set of $\hat{\mathbf{e}}$ associated with the global alignment $\vec{a} \in \Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})$. This concept is useful in our later proofs.

Example 2: An example of global alignments and alignment tree is shown in Fig. 8. In this example, there are two global alignments of $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ – $(2, 1, 1, 2)$ *and* $(1, 0, 1, 4)$. *Corresponding to the alignment* $(2, 1, 1, 2)$, *besides the typicalized edit pattern* $0\emptyset 101^{\downarrow 1}1$ *shown in the figure, there are five other typicalized edit patterns. Because deletion of either the first or the second* 0, *and an insertion of* 1 *in front of, between or after the last two* 1*'s result in the same* $\hat{\mathbf{Y}}$. *There is only one typicalized edit pattern associated with alignment* $(1, 0, 1, 4)$, *which is* $0^{\downarrow 1}01\emptyset 11$ *as shown in the figure. The example illustrates how global alignments classify typicalized edit patterns into different groups. The alignment tree is a structure of describing global alignments which helps illustrating the align module in Fig. 15 in Appendix B.* ■

D. RPES-LtRRID Model: Lower Bound

Recall that nature's secret is the uncertainty of the edit pattern given PreESS and PosESS. We now bound the typicalized nature's secret $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ from above by $H(\hat{\mathbf{E}}, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$. We further bound the latter quantity from above by the sum of two terms: the uncertainty $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ of the global alignment, and the uncertainty $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ of the typicalized edit pattern given the global alignment.

Lemma 5: The asymptotic entropy rate of the global alignment of $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ *is bounded from above by* $\lim_{n\to\infty} \frac{1}{n} H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \le \mathcal{O}(\max(\epsilon, \delta)^2)$.

Proof: See Appendix B. □

Lemma 6 below characterizes the entropy rate of the typicalized nature's secret.

Lemma 6: The asymptotic rate of the typicalized nature's secret is bounded from above by $\lim_{n\to\infty} \frac{1}{n} H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}) \le C_{|\mathcal{A}|}(\delta + \epsilon) + \mathcal{O}(\max(\epsilon, \delta)^2)$, *where* $C_{|\mathcal{A}|} = \sum_{l=1}^{\infty} \left(\frac{1}{|\mathcal{A}|}\right)^{l-1} \left(1 - \frac{1}{|\mathcal{A}|}\right)^2 l \log l$ *is a constant that depends only on the alphabet size* $|\mathcal{A}|$.

Proof: Knowing the global alignment of a pair of PreESS and PosESS $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$, the uncertainty in the typicalized edit pattern lies only in the uncertainty of locations of single-deletions and single-insertions of the same symbol (as those composing the run) within the $\bar{\mathbf{X}}$-runs. Recall that we denote the number of runs in $\bar{\mathbf{x}}$ by $\rho_{\bar{\mathbf{x}}}$, and the run lengths by $\{l_1, l_2, \ldots, l_{\rho_{\bar{\mathbf{x}}}}\}$. From Definition 1 of typicalized edit pattern, an $\bar{\mathbf{X}}$-run undergoes at most one edit. Hence, we derive the probability of insertions and deletions in typicalized edit pattern from both *symbol-perspective* and *run-perspective*, which is helpful in calculating $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$.

- *Symbol-perspective typicalized insertion/deletion probabilities:* For any $j = 1, 2, \ldots, \rho_{\bar{\mathbf{x}}}$, let $\hat{\delta}_j$ denote the probability that any specific symbol in the jth $\bar{\mathbf{x}}$-run is deleted,
$$\hat{\delta}_j = \delta(1 - \epsilon - \delta)^{l_j+1} \in \Big(\delta - (l_j + 1)(\delta^2 + \epsilon\delta), \delta\Big). \quad (3)$$
Similarly, let $\hat{\epsilon}_j$ denote the probability that there is an insertion between two specific symbols in the extended run of the jth $\bar{\mathbf{x}}$-run,
$$\hat{\epsilon}_j = \epsilon(1 - \epsilon - \delta)^{l_j+2} \in (\epsilon - (l_j + 2)(\epsilon^2 + \epsilon\delta), \epsilon). \quad (4)$$
In fact, we only need inequalities $\hat{\delta}_j \le \delta$ and $\hat{\epsilon}_j \le \epsilon$ for bounding the nature's secret of the typicalized edit process from above.[7] The intuition of $\hat{\delta}_j \le \delta$ and $\hat{\epsilon}_j \le \epsilon$ is that, because we eliminate some edits when typicalizing $\bar{\mathbf{E}}$ to $\hat{\mathbf{E}}$, the probability of insertions and deletions are smaller in typicalized edit patterns than the original insertion/deletion probability.
- *Run-perspective typicalized insertion/deletion probabilities:* Recall in Definition 5 that the global alignment $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ is a random variable in a vector form, of lengths of segments in $\bar{\mathbf{x}}$ which lead to the $\hat{\mathbf{y}}$-runs. Recall that $|\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})|$ denote the size of the sample space $\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})$ of $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$. The set of all typicalized edit patterns $\hat{\mathbf{e}}$ which takes $\bar{\mathbf{x}}$ to $\hat{\mathbf{y}}$ is denoted by $\{\hat{\mathbf{e}} : (\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}\}$. Given PreESS $\bar{\mathbf{x}}$, the probability that the typicalized PosESS is $\hat{\mathbf{y}}$ equals the sum of the probabilities over all $\hat{\mathbf{e}}$ such that processing $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$ results in $\hat{\mathbf{y}}$, *i.e.,* $p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) = \sum_{\{\hat{\mathbf{e}}:(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}\}} p(\hat{\mathbf{e}})$. The set $\{\hat{\mathbf{e}} : (\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}\}$ are classified into $|\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})|$ groups $\{\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}\}$ based on the global alignments, where $\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}$ denotes the set of $\hat{\mathbf{e}}$ associated with global alignment $\vec{a}$ of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$. Hence, for any $\vec{a} \in \Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})$,
$$p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) = \left(\sum_{\hat{\mathbf{e}}\in\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}} p(\hat{\mathbf{e}})\right) \Big/ \left(\sum_{\{\hat{\mathbf{e}}:(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}\}} p(\hat{\mathbf{e}})\right) = \left(\sum_{\hat{\mathbf{e}}\in\hat{\mathbf{E}}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}_{(\vec{a})}} p(\hat{\mathbf{e}})\right) \Big/ p(\hat{\mathbf{y}}|\bar{\mathbf{x}}).$$

[7] The specific distribution of the typicalized edit process might be of interest for future research on the capacity of InDel channels.

For any global alignment $\vec{a} \in \mathbf{\Omega}(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})$, let $D_{(\vec{a})}^{\rho_{\bar{\mathbf{x}}}} \in \{0,1\}^{\rho_{\bar{\mathbf{x}}}}$ denote the *run-perspective single-deletion pattern*, where $D_{(\vec{a}),j} = 1$ indicates that there is one deletion in the jth $\bar{\mathbf{x}}$-run when the global alignment of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$ is $\vec{a}$. Similarly, denote $I_{\text{same}(\vec{a})}^{\rho_{\bar{\mathbf{x}}}} \in \{0,1\}^{\rho_{\bar{\mathbf{x}}}}$ to be the *run-perspective single-same-symbol-insertion pattern*, where $I_{\text{same}(\vec{a}),j} = 1$ indicates that there is one insertion of the same symbol (insertion that extends the run) in the jth $\bar{\mathbf{x}}$-run when the global alignment of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$ is $\vec{a}$. Note that when typicalized edit pattern $\hat{\mathbf{e}}$ is given, the corresponding global alignment $\vec{a}$ is fixed. Hence, the probability that there is one deletion in the jth $\bar{\mathbf{x}}$-run averaging over all typicalized edit patterns is

$$\begin{aligned}
&\textstyle\sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{\vec{a}\in\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) p(D_{(\vec{a}),j} = 1) \\
&\quad = \textstyle\sum_{\hat{\mathbf{y}}} \sum_{\vec{a}\in\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} \sum_{\hat{\mathbf{e}}\in\hat{\mathbf{E}}_{(\vec{a})}^{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(\hat{\mathbf{e}}) p(D_{(\vec{a}),j} = 1) \\
&\quad = \textstyle\sum_{\hat{\mathbf{e}}} p(\hat{\mathbf{e}}) p(D_{(\vec{a}),j} = 1) \\
&\quad = l_j \hat{\delta}_j. \qquad (5)
\end{aligned}$$

Similarly, the probability that there is an insertion of the same symbol in the jth $\bar{\mathbf{x}}$-run averaging over all typicalized edit patterns is

$$\begin{aligned}
&\textstyle\sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{\vec{a}\in\Omega(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) p(I_{\text{same}(\vec{a}),j} = 1) \\
&\quad = \textstyle\sum_{\hat{\mathbf{e}}} p(\hat{\mathbf{e}}) p(I_{\text{same}(\vec{a}),j} = 1) \\
&\quad = \frac{1}{|\mathcal{A}|}(l_j + 1)\hat{\epsilon}_j. \qquad (6)
\end{aligned}$$

Given the global alignment of $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$, the uncertainty of the typicalized edit pattern is

$$\begin{aligned}
&H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \\
&= \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}}} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}) \sum_{\vec{a}\in\mathbf{\Omega}(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) H(\hat{\mathbf{E}}|\bar{\mathbf{x}}, \hat{\mathbf{y}}, \vec{a}) \\
&\overset{(a)}{=} \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}}} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}) \sum_{\vec{a}\in\mathbf{\Omega}(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \big(D_{(\vec{a}),j} \log l_j + \\
&\quad I_{\text{same}(\vec{a}),j} \log(l_j + 1)\big) \\
&= \sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{\vec{a}\in\mathbf{\Omega}(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) \cdot \\
&\quad \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \big(D_{(\vec{a}),j} \log l_j + I_{\text{same}(\vec{a}),j} \log(l_j + 1)\big) \\
&= \sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{\vec{a}\in\mathbf{\Omega}(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})} p(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = \vec{a}) \cdot \\
&\quad \big(p(D_{(\vec{a}),j} = 1) \log l_j + p(I_{\text{same}(\vec{a}),j} = 1) \log(l_j + 1)\big) \\
&\overset{(b)}{=} \sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \left(\hat{\delta}_j l_j \log l_j + \frac{1}{|\mathcal{A}|}\hat{\epsilon}_j (l_j + 1) \log(l_j + 1)\right) \\
&\overset{(c)}{\le} \sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \left(\delta l_j \log l_j + \frac{1}{|\mathcal{A}|}\epsilon (l_j + 1) \log(l_j + 1)\right) \\
&\overset{(d)}{=} \delta n \sum_{l=1}^{\infty} \left(\frac{1}{|\mathcal{A}|}\right)^{l-1} \left(1 - \frac{1}{|\mathcal{A}|}\right)^2 l \log l + \\
&\quad \frac{1}{|\mathcal{A}|}\epsilon n \sum_{l=1}^{\infty} \left(\frac{1}{|\mathcal{A}|}\right)^{l-1} \left(1 - \frac{1}{|\mathcal{A}|}\right)^2 (l+1) \log(l+1) + o(n) \\
&\overset{(e)}{=} (\delta + \epsilon) n \sum_{l=1}^{\infty} \left(\frac{1}{|\mathcal{A}|}\right)^{l-1} \left(1 - \frac{1}{|\mathcal{A}|}\right)^2 l \log l + o(n),
\end{aligned}$$

where step (a) follows because when the global alignment of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$ is known, the uncertainty only lies in the locations of the edits within those $\bar{\mathbf{x}}$-runs which undergo single-deletion and single-same-symbol-insertion. Step (b) follows from equations (5) and (6). Inequality (c) is because we have shown in (3) and (4) that $\hat{\delta}_j \le \delta$ and $\hat{\epsilon}_j \le \epsilon$. Step (d) is because asymptotically in n, the average number of length-l runs in $\bar{\mathbf{X}}$ is $\frac{np(l)}{E[L]} + o(n)$ [30], where $p(l) = \left(\frac{1}{|\mathcal{A}|}\right)^{l-1}\left(1 - \frac{1}{|\mathcal{A}|}\right)$ is the run length distribution of $\bar{\mathbf{X}}$ and $E[L] = 1/\left(1 - \frac{1}{|\mathcal{A}|}\right)$ is the expectation of the run length. Hence, by taking the sum of $l \log l$ based on the number of length-l runs in $\bar{\mathbf{x}}$ then averaging over $\bar{\mathbf{x}}$, $\sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} l_j \log l_j = \sum_{l=1}^{\infty} \frac{np(l)}{E[L]} l \log l + o(n)$, Similar calculation follows for $\sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} (l_j + 1) \log(l_j + 1)$. Step (e) follows by changing the term $(l + 1)$ to l.

Finally, the asymptotic entropy rate of the typicalized nature's secret can be bounded from above by

$$\begin{aligned}
&\lim_{n\to\infty} \frac{1}{n} H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}) \\
&\le \lim_{n\to\infty} \frac{1}{n} H(\hat{\mathbf{E}}, \hat{A}_{\bar{\mathbf{X}},\bar{\mathbf{Y}}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}) \\
&= \lim_{n\to\infty} \frac{1}{n} \big[H(\hat{A}_{\bar{\mathbf{X}},\bar{\mathbf{Y}}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) + H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \hat{A}_{\bar{\mathbf{X}},\bar{\mathbf{Y}}})\big] \\
&= \lim_{n\to\infty} \frac{1}{n} \big[H(\hat{A}_{\bar{\mathbf{X}},\bar{\mathbf{Y}}}) + H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \hat{A}_{\bar{\mathbf{X}},\bar{\mathbf{Y}}})\big] \\
&\le (\delta + \epsilon) \sum_{l=1}^{\infty} \left(\frac{1}{|\mathcal{A}|}\right)^{l-1} \left(1 - \frac{1}{|\mathcal{A}|}\right)^2 l \log l + \mathcal{O}(\max(\epsilon, \delta)^2).
\end{aligned}$$

□

In Lemma 7 below, we show that nature's secret (of the original edit pattern) is close to typicalized nature's secret. We first reprise a useful fact from [27].

Fact 1 [27][Fact V.25]: Suppose U, $\hat{U}$, and V are random variables with the property that U is a deterministic function of $\hat{U}$ and V, and also $\hat{U}$ is a deterministic function of U and V. (Denote this property by $U \overset{V}{\leftrightarrow} \hat{U}$.) Then

$$|H(U) - H(\hat{U})| \le H(V).$$

We use Fact 1 to bound $|H(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})|$ by $H(\hat{\mathbf{E}}^c)$. To do so, we map $(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}})$ as U, $(\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})$ as $\hat{U}$, and $\hat{\mathbf{E}}^c$ as V in Fact 1, and further, show below that the conditions required in Fact 1 are satisfied. Similarly, by mapping $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ as U, $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ as $\hat{U}$, and $(\hat{\mathbf{E}}^c, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ as V in Fact 1, and showing below that the conditions required in Fact 1 are also satisfied, we can bound $|H(\bar{\mathbf{X}}, \hat{\mathbf{Y}}) - H(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$ by $H(\hat{\mathbf{E}}^c, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$. The two sets of mappings in the following hold:

- $(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}}) \overset{\hat{\mathbf{E}}^c}{\leftrightarrow} (\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})$
  - "$\rightarrow$": The typicalized edit pattern $\hat{\mathbf{E}}$ (Definition 1) is a deterministic function of $\bar{\mathbf{E}}$ and $\bar{\mathbf{X}}$. Given $\hat{\mathbf{E}}$

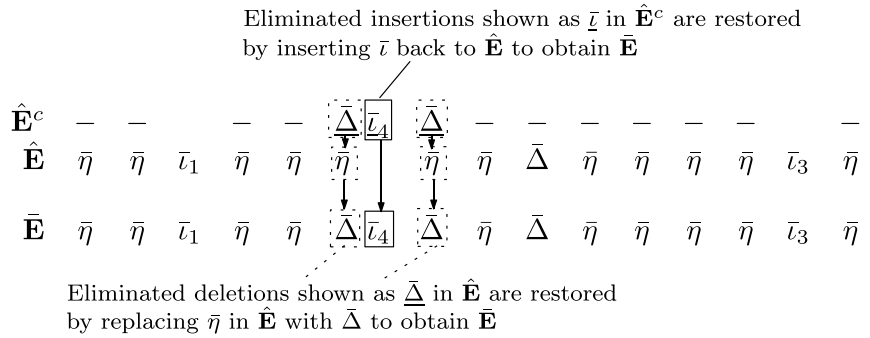

Fig. 9. Example of $\bar{\mathbf{E}} \xleftarrow{\hat{\mathbf{E}}^c} \hat{\mathbf{E}}$.

With $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$, segments of $\hat{\mathbf{Y}}$ and $\hat{\mathbf{E}}^c$ can be aligned to $\bar{\mathbf{X}}$-runs

$\bar{\mathbf{X}}$	0	0		0	1	1		1	1	2	2	3	2	3	3		
$\hat{\mathbf{Y}}$	0	0	1	0	1	1		1	1	2		3	2	3	3	3	
$\hat{\mathbf{E}}^c$	–	–		–	–	$\underline{\bar{\Delta}}$	$\underline{\bar{\iota}_4}$	$\underline{\bar{\Delta}}$	–	–	–	–	–	–	–		
$\bar{\mathbf{Y}}$	0	0	1	0	1		4		1	2		3	2	3	3	3	

- In the segments where $\hat{\mathbf{E}}^c$ are all '$-$'s, copy the segments of $\hat{\mathbf{Y}}$ to obtain $\bar{\mathbf{Y}}$;
- Otherwise, apply the eliminated edits as shown in $\hat{\mathbf{E}}^c$ back on $\bar{\mathbf{X}}$ to get $\bar{\mathbf{Y}}$.

Fig. 10. Example of $\bar{\mathbf{Y}} \xleftarrow{(\hat{\mathbf{E}}^c, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})} \hat{\mathbf{Y}}$.

and $\bar{\mathbf{X}}$, one can compute the typicalized PosESS $\hat{\mathbf{Y}}$ (Definition 2).

– "$\leftarrow$": The intuition of $(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}})$ being a deterministic function of $(\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})$ and $\hat{\mathbf{E}}^c$ is that, the original edit pattern $\bar{\mathbf{E}}$ is a union of typicalized edits $\hat{\mathbf{E}}$ and eliminated edits $\hat{\mathbf{E}}^c$. Specifically, one firstly aligns the no-operations '$\bar{\eta}$'s and the deletions '$\bar{\Delta}$'s in $\hat{\mathbf{E}}$ (in total n of them), to the no-eliminations '$-$'s and eliminated deletions '$\underline{\bar{\Delta}}$'s in $\hat{\mathbf{E}}^c$ (also n of them in total). Then one can obtain the original edit pattern $\bar{\mathbf{E}}$ from the typicalized edit pattern $\hat{\mathbf{E}}$ by changing the no-operations '$\bar{\eta}$'s to deletions '$\bar{\Delta}$'s where the symbols aligned with those no-operations '$\bar{\eta}$'s in $\hat{\mathbf{E}}^c$ is no-eliminations '$\underline{\bar{\Delta}}$'s, and placing insertions '$\bar{\iota}$'s back where eliminated insertions '$\underline{\bar{\iota}}$'s appear in $\hat{\mathbf{E}}^c$. A corresponding example is shown in Fig. 9. After determining $\bar{\mathbf{E}}$, $\bar{\mathbf{Y}}$ can be determined from $(\bar{\mathbf{X}}, \bar{\mathbf{E}})$.

- $(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \xleftrightarrow{(\hat{\mathbf{E}}^c, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})} (\bar{\mathbf{X}}, \hat{\mathbf{Y}})$

  – "$\rightarrow$": Firstly, by the definitions of $\bar{\mathbf{X}}$ and $\hat{\mathbf{E}}^c$, they can be aligned. The no-elimination sections of $\hat{\mathbf{E}}^c$ (comprises of only '$-$'s) align with those $\bar{\mathbf{X}}$-runs where no edits correspondingly in $\bar{\mathbf{E}}$ are eliminated. In other words, the edits $\bar{\mathbf{E}}$, hence PosESS $\bar{\mathbf{Y}}$ in the corresponding sections are typical (they don't change after typicalization), the alignment of which are specified by $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$. For sections in the eliminated edit pattern $\hat{\mathbf{E}}^c$ with some eliminated insertions '$\underline{\bar{\iota}}$' or deletions '$\underline{\bar{\Delta}}$', the corresponding sections in $\bar{\mathbf{Y}}$ can be verified by applying those eliminated edits to $\bar{\mathbf{X}}$. The corresponding sections in $\hat{\mathbf{Y}}$ should be the same as $\bar{\mathbf{X}}$, because the edits are all eliminated in those sections after typicalization.

  – "$\leftarrow$": With $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$, $\hat{\mathbf{Y}}$-runs can be aligned to parent run/runs in $\bar{\mathbf{X}}$ without any ambiguity. Also, the eliminated edits $\hat{\mathbf{E}}^c$ can be aligned to $\bar{\mathbf{X}}$. Hence, given typicalized PosESS $\hat{\mathbf{Y}}$ and eliminated edits $\hat{\mathbf{E}}^c$, one can reconstruct $\bar{\mathbf{Y}}$ as follows. If the corresponding section in $\hat{\mathbf{E}}^c$ for an $\bar{\mathbf{X}}$-run-$\hat{\mathbf{Y}}$-run match is "empty" (comprises of only '$-$'s), then we reconstruct the corresponding $\bar{\mathbf{Y}}$-run(s) the same as the $\hat{\mathbf{Y}}$-run(s). For sections in the eliminated edit pattern $\hat{\mathbf{E}}^c$ with some eliminated insertions '$\underline{\bar{\iota}}$' or deletions '$\underline{\bar{\Delta}}$', to reconstruct corresponding $\bar{\mathbf{Y}}$-runs, we only need to apply the eliminated edits stored in $\hat{\mathbf{E}}^c$ back to the corresponding $\bar{\mathbf{X}}$-runs. An example is shown in Fig. 10.

Lemma 7: The difference between the asymptotic entropy rates of nature's secret and typicalized nature's secret is bounded from above by $\lim_{n\to\infty} \frac{1}{n} |H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})| \leq 56 \max(\epsilon, \delta)^{2-\tau} + \mathcal{O}(\max(\epsilon, \delta)^2)$, *for some* $\tau \in (0, 1)$.

Proof: With Fact 1 and the two sets of mappings as shown above, $|H(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})|$ can be bounded by $H(\hat{\mathbf{E}}^c)$, and $|H(\bar{\mathbf{X}}, \hat{\mathbf{Y}}) - H(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$ can be bounded by $H(\hat{\mathbf{E}}^c, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$. Hence,

$$
\begin{aligned}
&|H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})| \\
&= |(H(\bar{\mathbf{E}}, \bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}, \bar{\mathbf{X}}, \hat{\mathbf{Y}})) + (H(\bar{\mathbf{X}}, \hat{\mathbf{Y}}) - H(\bar{\mathbf{X}}, \bar{\mathbf{Y}}))| \\
&\leq H(\hat{\mathbf{E}}^c) + H(\hat{\mathbf{E}}^c, \hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \\
&\leq 2\, H(\hat{\mathbf{E}}^c) + H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}).
\end{aligned}
$$

In $\hat{\mathbf{E}}^c = (\underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}, \underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)})$, let $\zeta_j^{\bar{\Delta}}$ denote the probability that $\underline{\bar{O}}_j$ is an elimination of deletion $\underline{\bar{\Delta}}$, we have $\zeta_j^{\bar{\Delta}} = \delta - \delta(1-\epsilon-\delta)^{l_{(j)}+1} \leq (l_{(j)}+1)(\epsilon\delta+\delta^2)$, where $l_{(j)}$ is the length of the $\bar{\mathbf{X}}$-run where the eliminated deletion occurs. Averaging over $\bar{\mathbf{X}}$, $E_{\bar{\mathbf{X}}}[\zeta_j^{\bar{\Delta}}] \leq (E_{\bar{\mathbf{X}}}[l_{(j)}]+1)(\epsilon\delta+\delta^2)$. Note that the average length of an $\bar{\mathbf{X}}$-run is $E_{\bar{\mathbf{X}}}[l_{(j)}] = \frac{|\mathcal{A}|}{|\mathcal{A}|-1} \leq 2$, where equality holds when $|\mathcal{A}| = 2$. Hence, on average the probability that a deletion in $\bar{\mathbf{E}}$ gets eliminated is $\zeta^{\bar{\Delta}} \leq 3(\epsilon\delta+\delta^2) \leq 6\max(\epsilon,\delta)^2$. Similarly, there is an elimination of an insertion in $\hat{\mathbf{E}}^c$ with probability $\zeta_j^{\bar{\iota}} = \epsilon - \epsilon(1-\epsilon-\delta)^{l_{(j)}+2} \leq (l_{(j)}+2)(\epsilon\delta+\epsilon^2)$. Hence, averaging over $\bar{\mathbf{X}}$, an insertion in $\bar{\mathbf{E}}$ gets eliminated with probability $\zeta^{\bar{\iota}} \leq (E_{\bar{\mathbf{X}}}[l_{(j)}]+2)(\epsilon\delta+\epsilon^2) \leq 4(\epsilon\delta+\epsilon^2) \leq 8\max(\epsilon,\delta)^2$.

With similar calculations as in equation (8) in the proof of Lemma 4 in Appendix A,

$$\begin{aligned}
H(\underline{\bar{O}}_1) &= \mathcal{H}(\zeta^{\bar{\Delta}}, \zeta^{\bar{\iota}}, 1-\zeta^{\bar{\Delta}}-\zeta^{\bar{\iota}}) \\
&= \mathcal{H}(\zeta^{\bar{\Delta}}) + \mathcal{H}(\zeta^{\bar{\iota}}) - (\log e)\zeta^{\bar{\Delta}}\zeta^{\bar{\iota}} + \mathcal{O}(\max(\zeta^{\bar{\Delta}}, \zeta^{\bar{\iota}})^3) \\
&= -\zeta^{\bar{\Delta}}\log(\zeta^{\bar{\Delta}}) - (1-\zeta^{\bar{\Delta}})\log(1-\zeta^{\bar{\Delta}}) + H(\zeta^{\bar{\iota}}) \\
&\quad + \mathcal{O}(\max(\epsilon,\delta)^4) \\
&= -\zeta^{\bar{\Delta}}\log(\zeta^{\bar{\Delta}}) - (1-\zeta^{\bar{\Delta}})(\log e)(-\zeta^{\bar{\Delta}} + \mathcal{O}((\zeta^{\bar{\Delta}})^2)) \\
&\quad + H(\zeta^{\bar{\iota}}) + \mathcal{O}(\max(\epsilon,\delta)^4) \\
&= -\zeta^{\bar{\Delta}}\log(\zeta^{\bar{\Delta}}) + (\log e)\zeta^{\bar{\Delta}} - \zeta^{\bar{\iota}}\log(\zeta^{\bar{\iota}}) + (\log e)\zeta^{\bar{\iota}} \\
&\quad + \mathcal{O}(\max(\epsilon,\delta)^4) \\
&\leq 12\max(\epsilon,\delta)^{2-\tau} + 16\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2) \\
&= 28\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2).
\end{aligned}$$

Hence,

$$\begin{aligned}
&H(\hat{\mathbf{E}}^c) \\
&= H(\underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}, \underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)}) \\
&= H(\underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}) + H(\underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)} | \underline{\bar{O}}^{(n+K_\iota-\hat{K}_\iota)}) \\
&\overset{(a)}{=} (n + E[K_\iota] - E[\hat{K}_\iota]) H(\underline{\bar{O}}_1) + H(\underline{\bar{C}}^{(K_\iota-\hat{K}_\iota)} | (K_\iota - \hat{K}_\iota)) \\
&= (n + E[K_\iota] - E[\hat{K}_\iota]) H(\underline{\bar{O}}_1) + \left(E[K_\iota] - E[\hat{K}_\iota]\right)\log|\mathcal{A}| \\
&\leq \frac{n+\epsilon}{1-\epsilon} H(\underline{\bar{O}}_1) + \frac{n+\epsilon}{1-\epsilon}\zeta^{\bar{\iota}}\log|\mathcal{A}| \\
&\leq \frac{n+\epsilon}{1-\epsilon}\Big(28\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2) \\
&\quad + 8\max(\epsilon,\delta)^2\log|\mathcal{A}|\Big) \\
&= \frac{n+\epsilon}{1-\epsilon}\left(28\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)\right)
\end{aligned}$$

where step (a) is by Theorem 1.

Hence, we have from the beginning of the proof that $\lim_{n\to\infty}\frac{1}{n}|H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})| \leq \lim_{n\to\infty}\frac{1}{n}\left(2H(\hat{\mathbf{E}}^c) + H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})\right) \leq 56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)$ for some $\tau \in (0,1)$. (Recall in Lemma 5 we have shown that $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \leq \mathcal{O}(\max(\epsilon,\delta)^2)n$.) □

Remark: For our purpose of bounding the achievable rate from below, we only need one direction of the inequality, that is, $\lim_{n\to\infty}\frac{1}{n}(H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})) \geq -56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)$. Including the other direction of bounding in Lemma 7 might be useful for ongoing research on InDel channel capacity.

Theorem 8 below is our main result which characterizes the information-theoretic lower bound of the optimal compression rate for updating with RPES-LtRRID model.

Theorem 8: The optimal average transmission rate for updating with RPES-LtRRID process $\bar{R}^*_{\epsilon,\delta} = \lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) \geq \mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon\log|\mathcal{A}| - (\delta+\epsilon)C_{|\mathcal{A}|} - 56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)$, *for some* $\tau \in (0,1)$.

Proof: Combine Lemma 3, 4, 6, and 7, we have

$$\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) \\
&= \lim_{n\to\infty}\frac{1}{n}[H(\bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})] \\
&= \lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{E}}) - \lim_{n\to\infty}\frac{1}{n}H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}}) - \\
&\quad \lim_{n\to\infty}\frac{1}{n}(H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})) \\
&\geq \mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon\log|\mathcal{A}| + 2\min(\epsilon,\delta)^{2-\tau} - (\delta+\epsilon)C_{|\mathcal{A}|} \\
&\quad -56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2) \\
&\geq \mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon\log|\mathcal{A}| - (\delta+\epsilon)C_{|\mathcal{A}|} \\
&\quad -56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2),
\end{aligned}$$

for some $\tau \in (0,1)$. □

Remark: When $\epsilon = 0$ and $|\mathcal{A}| = 2$, our result matches with result in Corollary IV.5. for binary deletion channel in [15].

E. APES-AID Model: Lower Bound

Given an arbitrary length-n pre-edit source sequence $\mathbf{x}$ over $\mathcal{A}$, let the set $\mathcal{Y}_{\epsilon,\delta}(\mathbf{x})$, called the post-edit set, denote the set of all sequences over $\mathcal{A}$ that can be obtained from $\mathbf{x}$ via any arbitrary (ϵ,δ)-InDel pattern. For zero-error decodability, the encoder needs to send $\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{x})|$ bits to decoder. The larger the set $\mathcal{Y}_{\epsilon,\delta}(\mathbf{x})$, the larger the corresponding lower bound on the optimal achievable rate. Hence to find a good lower bound on the optimal achievable rate, one needs to find a pre-edit sequence $\mathbf{x}$ with a large post-edit set.

In two special cases, that is, an insertion model with arbitrary ϵn insertions, and an deletion model with arbitrary δn deletions, the sizes of post-edit sets have been well studied in the literature. We here reprise the results in [18] and [19], with our notation. For an arbitrary ϵ-insertion process, the size of any post-edit set $|\mathcal{Y}_{\epsilon,0}(\mathbf{X})| = \sum_{j=0}^{\epsilon n}\binom{n+\epsilon n}{j}(|\mathcal{A}|-1)^j \geq \binom{n+\epsilon n}{\epsilon n}(|\mathcal{A}|-1)^{\epsilon n}$, which is independent of the PreESS $\mathbf{X}$. For the arbitrary δ-deletion process, the size of the largest post-edit set $|\mathcal{Y}_{0,\delta}(\mathbf{X})| \geq \sum_{j=0}^{\delta n}\binom{n-\delta n}{j} \geq \binom{n-\delta n}{\delta n}$ depends on the PreESS $\mathbf{X}$. In the following, we give examples of the PreESSs and intuitions of the lower bounds for these two special cases.

For an arbitrary ϵ-insertion process, consider a PreESS denoted by $\mathbf{X}_\alpha$, which is a single length-n run consisting of only symbol $\alpha \in \mathcal{A}$. Consider insertions of the form that out of the $n+\epsilon n$ locations in the PosESS $\mathbf{Y}$, exactly ϵn locations correspond to insertions of symbols other than α. For such a PreESS $\mathbf{X}_\alpha$ and such insertion patterns, all the

possible resulting PosESS $\mathbf{Y}$'s are distinct. The number of such insertion patterns is bounded below by $\binom{n+\epsilon n}{\epsilon n}(|\mathcal{A}|-1)^{\epsilon n}$. Hence, a lower bound on the number of PosESS $|\mathcal{Y}_{\epsilon,0}(\mathbf{X}_\alpha)|$ is $\binom{n+\epsilon n}{\epsilon n}(|\mathcal{A}|-1)^{\epsilon n}$. The corresponding lower bound on the optimal achievable rate is $\frac{1}{n}\log|\mathcal{Y}_{\epsilon,0}(\mathbf{X}_\alpha)|$. As n goes to infinity, by Stirling's approximation [31], the lower bound converges to $(1+\epsilon)\mathcal{H}(\frac{\epsilon}{1+\epsilon})+\epsilon\log(|\mathcal{A}|-1)$.

For an arbitrary δ-deletion process, consider a PreESS denoted by $\mathbf{X}_{\text{diff}}$, where each symbol is different from its preceding symbol, in other words, $\mathbf{X}_{\text{diff}}$ consists of n length-1 runs. Consider the set of deletion patterns which delete an arbitrary subset of δn non-neighboring symbols from $\mathbf{X}_{\text{diff}}$, hence we require $\delta \leq 1/2$ here. Note that each such deletion pattern results in a distinct PosESS $\mathbf{Y}$. The number of these deletion patterns is $\binom{n-\delta n+1}{\delta n}$. The corresponding lower bound on the optimal achievable rate is $\frac{1}{n}\log|\mathcal{Y}_{0,\delta}(\mathbf{X}_{\text{diff}})|$. As n goes to infinity, by Stirling's approximation [31], the lower bound converges to $(1-\delta)\mathcal{H}(\frac{\delta}{1-\delta})$.

Theorem 9: When the source alphabet has size $|\mathcal{A}|\geq 3$ and the deletion fraction $\delta \leq 1/2$, the optimal transmission rate for updating with APES-AID process $R^*_{\epsilon,\delta} \geq \mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log(|\mathcal{A}|-2)-(\delta^2+\delta\epsilon-\epsilon^2)\log e+\mathcal{O}(\max(\epsilon,\delta)^3)$.

Proof: Consider a PreESS $\mathbf{X}_{\text{LB}}$ constructed by alternating two source symbols, *e.g.* $0101\ldots01$. This PreESS has largest number of runs – n runs, at the same time composes least number of symbols – two symbols, to form n runs. (If the sequence is composed of one source symbol, it must have only one run.)

We now describe a set of arbitrary (ϵ,δ)-InDel patterns that result in a large $\mathbf{X}_{\text{LB}}$-post-edit set. In this set of InDel patterns, we require that all δn deletions precede all ϵn insertions. Further, the deletions may delete any δn non-neighboring symbols (hence no contiguous deletions). This can happen because we require $\delta \leq 1/2$. The insertions can insert only symbols from $\{2,\ldots,|\mathcal{A}|-1\}$, *i.e.,* symbols different from those composing $\mathbf{X}_{\text{LB}}$. This can be satisfied given that $|\mathcal{A}|\geq 3$.

It can be verified that each such edit pattern results in a distinct PosESS $\mathbf{Y}$, by noting that given $\mathbf{X}_{\text{LB}}$ and $\mathbf{Y}$, one can reconstruct the edit pattern unambiguously. To do so, one first checks for the symbols different from those composing $\mathbf{X}_{\text{LB}}$ (those from the set $\{2,\ldots,|\mathcal{A}|-1\}$) to identify the insertion pattern uniquely. Then one takes out those inserted symbols, aligns the remaining sequence to $\mathbf{X}_{\text{LB}}$ and checks for the missing symbols (the 0's and 1's that are deleted) to identify the deletion pattern uniquely. Because no pairs of neighboring symbols get deleted, one can always identify the deleted 0's and 1's unambiguously by comparing $\mathbf{X}_{\text{LB}}$ and the remaining sequence after taking away the insertions. The overall InDel pattern is then the composition of the deletion pattern and the insertion pattern.

The number of such InDel patterns as described above is $\binom{n-\delta n+1}{\delta n}\binom{n-\delta n+\epsilon n}{\epsilon n}(|\mathcal{A}|-2)^{\epsilon n}$, hence is a lower bound on the number of PosESSs – $|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\text{LB}})|$. The corresponding lower bound on the optimal achievable rate $R^*_{\epsilon,\delta}$, given by $\frac{1}{n}\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\text{LB}})|$, is asymptotically $(1-\delta)\mathcal{H}\left(\frac{\delta}{1-\delta}\right)+(1-\delta+\epsilon)\mathcal{H}\left(\frac{\epsilon}{1-\delta+\epsilon}\right)+\epsilon\log(|\mathcal{A}|-2)$ by Stirling's approximation [31]. By expanding the binary entropy functions and taking Taylor series expansion, the optimal transmission rate is at least $\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log(|\mathcal{A}|-2)-(\delta^2+\delta\epsilon-\epsilon^2)\log e+\mathcal{O}(\max(\epsilon,\delta)^3)$. □

F. Summary

To summarize in this section, we derive information-theoretic lower bounds on optimal compression rates for both the RPES-LtRRID model and the APES-AID model. In Section III-B, we show that the optimal compression length for the RPES-LtRRID model is the description length of the edit sequence $H(\bar{\mathbf{E}})$, less the entropy of the nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$. We characterize the lower bound up to first order terms in ϵ and δ in Theorem 8. For the APES-AID model, we show in Section III-E that for certain type of PreESSs, the optimal compression rate is close to the entropy rate to describe the locations of edits, and the contents of insertions. We characterize the lower bound up to first order terms in ϵ and δ in Theorem 9 when the alphabet size is at least 3. In the next section, we propose algorithms for both models and show that our algorithms achieve compression rates that are close to the lower bounds in lower order terms.

IV. Algorithm and Performance

We first propose a unified *dynamic-programming-entropy-coding (DP-EC)* compression scheme for both RPES-LtRRID and APES-AID models in Section IV-A. The DP-EC scheme is a combination of dynamic programming and entropy coding. Note that using DP to find the edit distance between two sequences is well-known in the literature – the contribution here is to demonstrate that for large alphabet and small amount of edits, this algorithmic procedure results in expected description lengths that are within small constant additive gap to the information-theoretic lower bounds that we provide in Section III, which we show in Section IV-B. In Section IV-C, we present a *dynamic-programming-run-length-compression (DP-RLC)* scheme for the RPES-LtRRID model, which is modified slightly from the scheme in [9] to suit our model. The DP-RLC scheme has only one step different from the DP-EC scheme – after DP-encoder outputs an edit pattern, the DP-RLC scheme groups edits output by DP according to the lengths of the runs where those edits occur, and uses entropy codes to compress the grouped edit sequences. In Section IV-D, we show that the DP-RLC scheme achieves a compression rate matching the lower bound up to first order terms in ϵ and δ, including the first order term of the nature's secret, hence outperforms the DP-EC scheme for the RPES-LtRRID model.

A. Dynamic-Programming-Entropy-Coding (DP-EC) Scheme – Algorithm

For this section of a unified algorithm for APES-AID and PRES-LtRRID models, we unify presentation by using notation without bars. The encoding process is summarized in Algorithm 1.

Algorithm 1 Dynamic-Programming-Entropy-Coding Encoder

Input: The PreESS $\mathbf{X}$ and the PosESS $\mathbf{Y}$

Output: A transmission Enc$(\mathbf{X}, \mathbf{Y})$

1: **DP-enc:** Run a *dynamic program* on the input $(\mathbf{X}, \mathbf{Y})$ to output an edit pattern $\tilde{\mathbf{E}}$.
2: **Repre-enc:** Represent the edit pattern $\tilde{\mathbf{E}}$ as a pair of sequences $(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$.
3: **Entro-enc:** Use standard entropy codes [31] to compress sequences $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$.

Some additional information of the algorithm is as follows:

- For DP-enc, denote the number of insertions and deletions of $\tilde{\mathbf{E}}$ by $\tilde{\epsilon}n$ and $\tilde{\delta}n$ respectively, the edit pattern $\tilde{\mathbf{E}}$ satisfies the condition that $(\tilde{\epsilon}+\tilde{\delta})n$ is the minimum number of edits needed to convert $\mathbf{X}$ to $\mathbf{Y}$. Standard edit-distance algorithms [32] typically run in time that scales as $n_1\, n_2$ – the product of the lengths of the strings being compared. We reference here Ukkonen's work [33] since it gives an algorithm that is $\mathcal{O}(\min(n_1, n_2)\cdot s)$, where s refers to the *edit distance* between $\mathbf{X}$ and $\mathbf{Y}$ – the minimum number of edits needed to process on $\mathbf{X}$ to get $\mathbf{Y}$, and hence is faster.
- For Repre-enc, the edit operation sequence $\tilde{O}^{n+\tilde{\epsilon}n}$ is a length-$(n+\tilde{\epsilon}n)$ sequence over $\{\bar{\imath}, \bar{\Delta}, \bar{\eta}\}$ which specifies the edit operations of $\tilde{\mathbf{E}}$. The insertion content sequence $\tilde{C}^{\tilde{\epsilon}n}$ is a length-$\tilde{\epsilon}n$ sequence over the source alphabet $\mathcal{A}$ which specifies the content of insertions of $\tilde{\mathbf{E}}$.

The decoder of the DP-EC algorithm decodes $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$ by corresponding decoders of the entropy codes in Entro-enc in Algorithm 1, and reconstructs $\mathbf{Y}$ from $(\mathbf{X}, \tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$.

B. Dynamic-Programming-Entropy-Coding (DP-EC) Scheme – Performance

It is well known in the literature [32], [33] that dynamic programming finds the edit distance between two sequences, that is, DP minimizes $\tilde{\epsilon}n+\tilde{\delta}n$ – the total number of insertions and deletions. In fact, $\tilde{\epsilon}n$ and $\tilde{\delta}n$ are both minimized in the output of DP, because given $\mathbf{X}$ and $\mathbf{Y}$, the numbers of insertions and deletions satisfy $\tilde{\epsilon}n-\tilde{\delta}n = l(\mathbf{Y})-l(\mathbf{X})$ – the difference between the lengths of $\mathbf{Y}$ and $\mathbf{X}$. Hence, minimizing $\tilde{\epsilon}n+\tilde{\delta}n$ over all edit patterns that convert $\mathbf{X}$ to $\mathbf{Y}$ minimizes both $\tilde{\epsilon}n$ and $\tilde{\delta}n$. Hence, for both APES-AID and RPES-LtRRID models, denote $K_\imath$ the number of insertions and K_Δ the number of deletions in the actual edit pattern, we have $\tilde{\epsilon}n \le K_\imath$ and $\tilde{\delta}n \le K_\Delta$. We conclude in Fact 2 below.

Fact 2: The number of insertions $\tilde{\epsilon}n$ (respectively the number of deletions $\tilde{\delta}n$) of any edit pattern $\tilde{\mathbf{E}}$ output by DP-encoder is always no larger than the number of insertions $K_\imath$ (respectively the number of deletions K_Δ) of the actual edit pattern. Hence,

- *for APES-AID model, $\tilde{\epsilon}n \le K_\imath \le \epsilon n, \tilde{\delta}n \le K_\Delta \le \delta n$;*
- *for RPES-LtRRID model, $\tilde{\epsilon}n \le K_\imath, \tilde{\delta}n \le K_\Delta$.*

In the limit as the block length n goes to infinity, the compression rate of the above algorithm is $\lim_{n\to\infty}\frac{1}{n}\left(H(\tilde{O}^{n+\tilde{\epsilon}n})+H(\tilde{C}^{\tilde{\epsilon}n})\right)$. In Lemma 10 below, we characterize the asymptotic entropy rate of the edit operation sequence $\tilde{O}^{n+\tilde{\epsilon}n}$ output by DP with $\tilde{\epsilon}n$ insertions and $\tilde{\delta}n$ deletions. The proof is provided in Appendix C.

Lemma 10: The asymptotic entropy rate of $\tilde{O}^{n+\tilde{\epsilon}n}$ with $\tilde{\epsilon}n$ insertions and $\tilde{\delta}n$ deletions is $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}) = \mathcal{H}(\tilde{\delta})+\mathcal{H}(\tilde{\epsilon})+(\log e)\tilde{\epsilon}^2+\mathcal{O}(\tilde{\epsilon}^4)$.

In the following, we characterize upper bounds on the compression rates of DP-EC algorithm for both RPES-LtRRID and APES-AID models.

1) DP-EC Performance – RPES-LtRRID Model: In RPES-LtRRID model, the numbers of deletions and insertions may exceed their expectations $\frac{\delta}{1-\epsilon}n$ and $\frac{\epsilon}{1-\epsilon}(n+1)$, in which case more bits may need to be transmitted. Moreover, the number of insertions can be unbounded. In Theorem 11 below, we show that these events contribute a negligible amount to the compression rate as the block length n tends to infinity. Specifically, we use the Chernoff bound to show that the probability of the events, where the number of insertions (deletions) is much more than its expected value, is sub-exponentially small in the block length n.

Theorem 11: For RPES-LtRRID model, the DP-EC algorithm requires a compression rate of at most $\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|+\epsilon\delta^{1-\tau}+\epsilon^{2-\tau}+\epsilon^2\log|\mathcal{A}|-\frac{\delta^2+\epsilon^2}{2}\log e+\mathcal{O}(\max(\epsilon,\delta)^{3-\tau})$, *for some* $\tau\in(0,1)$.

Proof: The number of deletions K_Δ is the sum of n i.i.d. Bernoulli $\left(\frac{\delta}{1-\epsilon}\right)$ random variables. Hence, by the Chernoff bound $\Pr\left(K_\Delta \ge \frac{\delta}{1-\epsilon}n+\lambda_1\, n\right) \le e^{-2\lambda_1^2\, n}$. Taking $\lambda_1 = n^{-1/4}$, we have $\Pr\left(K_\Delta \ge \frac{\delta}{1-\epsilon}n+n^{3/4}\right) \le e^{-2\sqrt{n}}$.

The number of insertions $K_\imath$ is the sum of $n+1$ i.i.d. Geometric random variables denoted as $\text{Geo}_0(1-\epsilon)$, hence is not bounded from above. Standard concentration inequalities like Chernoff bounds or Bernstein's inequality *etc.* do not apply. We found that the bound in [34] suffices, which states that $\Pr\left(K_\imath \ge (1+\lambda_2)\frac{\epsilon}{1-\epsilon}(n+1)\right) \le \exp\left(\frac{-(1+\lambda_2)(n+1)\left(1-\frac{1}{1+\lambda_2}\right)^2}{2}\right)$. Taking $\lambda_2 = n^{-1/4}$, we have $\Pr\left(K_\imath \ge \frac{\epsilon}{1-\epsilon}(n+1)+\frac{\epsilon}{1-\epsilon}(n^{3/4}+n^{-1/4})\right) \le \exp\left(\frac{-(n+1)n^{-1/2}}{2(1+n^{-1/4})}\right) \le e^{-\sqrt{n}/4}$.

Case 1: From the above, with probability at least $1-e^{-2\sqrt{n}}-e^{-\sqrt{n}/4}$, by Fact 2 we have $\tilde{\delta} \le \frac{\delta}{1-\epsilon}+n^{-1/4}$ and $\tilde{\epsilon} \le \frac{\epsilon}{1-\epsilon}+\frac{\epsilon}{1-\epsilon}(n^{-1}+n^{-1/4}+n^{-5/4})$. By Lemma 10, the contribution to $\lim_{n\to\infty}\frac{1}{n}\left(H(\tilde{O}^{n+\tilde{\epsilon}n})+H(\tilde{C}^{\tilde{\epsilon}n})\right)$ in the regime that $K_\imath \le \frac{\epsilon}{1-\epsilon}(n+1)+\frac{\epsilon}{1-\epsilon}(n^{3/4}+n^{-1/4})$ and $K_\Delta \le \frac{\delta}{1-\epsilon}n+n^{3/4}$ is at most

$$\mathcal{H}\left(\frac{\delta}{1-\epsilon}\right)+\mathcal{H}\left(\frac{\epsilon}{1-\epsilon}\right)+\frac{\epsilon}{1-\epsilon}\log|\mathcal{A}| + \left(\frac{\epsilon}{1-\epsilon}\right)^2\log e+\mathcal{O}\left(\left(\frac{\epsilon}{1-\epsilon}\right)^4\right). \quad (7)$$

Case 2: Also, from the concentration inequalities above, with probability at most $e^{-2\sqrt{n}}+e^{-\sqrt{n}/4}$, $K_\Delta \in [\frac{\delta}{1-\epsilon}n+n^{3/4}, n]$ or $K_\imath \in [\frac{\epsilon}{1-\epsilon}(n+1)+\frac{\epsilon}{1-\epsilon}(n^{3/4}+n^{-1/4})), \infty)$. We first investigate the case that $K_\imath$ is also bounded above by n. The probability of this event ($K_\Delta \in [\frac{\delta}{1-\epsilon}n+n^{3/4}, n]$ or $K_\imath \in [\frac{\epsilon}{1-\epsilon}(n+1)+\frac{\epsilon}{1-\epsilon}(n^{3/4}+n^{-1/4})), n]$) is still bounded above by $e^{-2\sqrt{n}}+e^{-\sqrt{n}/4}$. The number of bits needed to specify

the edit pattern is at most linear in n (bounded from above by $2n \log |\mathcal{A}|$). However, the probability is sub-polynomial small in n. Hence, as the block length n goes to infinity, the contribution to $\lim_{n\to\infty} \left(H(\tilde{O}^{n+\tilde{\epsilon}n}) + H(\tilde{C}^{\tilde{\epsilon}n}) \right)$ by the event that $K_\Delta \in [\frac{\delta}{1-\epsilon} n + n^{3/4}, n]$ or $K_\imath \in [\frac{\epsilon}{1-\epsilon}(n+1) + \frac{\epsilon}{1-\epsilon}(n^{3/4} + n^{-1/4})), n]$ goes to zero.

Case 3: As mentioned above, the number of insertions $K_\imath$ can be unbounded. For the case $K_\imath > n$, when $K_\imath$ is linear in n ($K_\imath = \Theta(n)$), the number of bits needed to specify the edit pattern is linear in n, whereas the probability of this event is bounded above by $e^{-\sqrt{n}/4}$. When $K_\imath$ is $\Omega(n)$, recall that $K_\imath$ is negative binomial distributed, the pmf of which is $\Pr(K_\imath = k) = \binom{k+n}{k}(1-\epsilon)^{n+1}\epsilon^k$. Hence, in this case the number of bits needed to specify the edit pattern is linear in $K_\imath$, with probability of exponentially small in $K_\imath$. From both cases, as n goes to infinity, the contribution to $\lim_{n\to\infty} \frac{1}{n} \left(H(\tilde{O}^{n+\tilde{\epsilon}n}) + H(\tilde{C}^{\tilde{\epsilon}n}) \right)$ by the event that $K_\imath > n$ goes to zero.

From the above analysis, the entropy rate contributed to $\lim_{n\to\infty} \frac{1}{n} \left(H(\tilde{O}^{n+\tilde{\epsilon}n}) + H(\tilde{C}^{\tilde{\epsilon}n}) \right)$ by Case 2 and Case 3 goes to zero. Hence, from (7), we have $\lim_{n\to\infty} \frac{1}{n} \left(H(\tilde{O}^{n+\tilde{\epsilon}n}) + H(\tilde{C}^{\tilde{\epsilon}n}) \right) \leq \mathcal{H}(\frac{\delta}{1-\epsilon}) + \mathcal{H}(\frac{\epsilon}{1-\epsilon}) + \frac{\epsilon}{1-\epsilon} \log |\mathcal{A}| + (\log e)(\frac{\epsilon}{1-\epsilon})^2 + \mathcal{O}\left((\frac{\epsilon}{1-\epsilon})^4\right)$. We conclude our bound as a Taylor series expansion in ϵ and δ, by expanding the terms individually by Taylor series expansion.

The rate achieved by DP-EC algorithm is bounded from above by $\mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon \log |\mathcal{A}| + \epsilon\delta^{1-\tau} + \epsilon^{2-\tau} + \epsilon^2 \log |\mathcal{A}| - \frac{\delta^2+\epsilon^2}{2} \log e + \mathcal{O}(\max(\epsilon, \delta)^{3-\tau})$, for some $\tau \in (0, 1)$. □

2) DP-EC Performance – APES-AID Model: The performance analysis of DP-EC algorithm for APES-AID model is less complicated, because APES-AID model restricts the amount of insertions (deletions) to be at most ϵn (δn).

Theorem 12: For APES-AID model, the DP-EC algorithm requires a compression rate of at most $\mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon \log |\mathcal{A}| + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^4)$.

Proof: As we have mentioned above, the asymptotic compression rate of the DP-EC algorithm is given by $\lim_{n\to\infty} \frac{1}{n} \left(H(\tilde{O}^{n+\tilde{\epsilon}n}) + H(\tilde{C}^{\tilde{\epsilon}n}) \right)$. By Lemma 10, the entropy rate of $\tilde{O}^{n+\tilde{\epsilon}n}$ is given by $\lim_{n\to\infty} \frac{1}{n} H(\tilde{O}^{n+\tilde{\epsilon}n}) = \mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)$. The contents of insertions can be arbitrary symbols from $\mathcal{A}$, hence $\lim_{n\to\infty} \frac{1}{n} H(\tilde{C}^{\tilde{\epsilon}n}) = \lim_{n\to\infty} \frac{1}{n} \tilde{\epsilon} n \log |\mathcal{A}| = \tilde{\epsilon} \log |\mathcal{A}|$. So the compression rate of DP-EC algorithm for the APES-AID model is at most $\mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + \tilde{\epsilon} \log |\mathcal{A}| + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)$. By Fact 2, $\tilde{\epsilon} \leq \epsilon$ and $\tilde{\delta} \leq \delta$. Hence, the compression rate for APES-AID model is at most $\mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon \log |\mathcal{A}| + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^4)$. □

Remark: Note that the achievable rates in Theorem 11 and Theorem 12 are similar to Lemma 4. Although Lemma 4 is for RPES-LtRRID model only, the similarity indicates that directly compressing the output edit pattern of DP costs a description length about the same as the entropy of the edit sequences, *i.e.,* the number of bits to describe the locations of insertions and deletions, plus the contents of insertions. For APES-AID model, the achievable rate of DP-EC algorithm almost match the lower bound (Theorem 9) in lower order terms. However, for RPES-LtRRID model the lower bound (Theorem 8) indicates that one might need less bits (amounted by the nature's secret term) to enable updates. In next Section IV-C, we present an algorithm for RPES-LtRRID model which outperforms DP-EC algorithm.

C. Dynamic-Programming-Run-Length Compression (DP-RLC) Scheme – Algorithm

In this section, we present the DP-RLC compression scheme specifically for the RPES-LtRRID model. The scheme is slightly modified from the compression scheme in [9], because there are some differences between our model and the model in [9] in both source sequences and edit sequences. Another difference between our work and [9] is that we explicitly calculate the lower order terms of the achievable rate in Theorem 13.

Before going into details of the algorithm, we firstly classify insertions into two types:

- *Type-E insertions:* this includes insertions which insert the same symbol composing the runs where they occur, hence they extend the lengths of $\bar{\mathbf{X}}$-runs.
- *Type-O insertions:* this includes all the other insertions, hence they create new runs and/or break $\bar{\mathbf{X}}$-runs. Specifically, when they occur within runs, they break these runs and form new runs themselves; when they occur at the positions between two runs, if the source alphabet has size at least three, it is possible that they create new runs themselves.

The encoding process is summarized in Algorithm 2.

Algorithm 2 Dynamic-Programming-Run-Length Compression Encoder

Input: The PreESS $\bar{\mathbf{X}}$ and the PosESS $\bar{\mathbf{Y}}$
Output: A transmission $\overline{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$

1: **DP-enc:** Run a dynamic program on the input $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ to output an edit pattern.
2: **RL-Grouping-enc:** Group deletions and type-E insertions according to lengths of $\bar{\mathbf{X}}$-runs, denoted by $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$ and $\{\tilde{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}$. Type-O insertions are described in respect of the whole sequence of $\bar{\mathbf{Y}}$, and are denoted by $\tilde{\mathbf{I}}^{\mathrm{O}}$. // Details of this step is shown below.
3: **Entro-enc:** Use standard entropy codes [31] to compress the three sets of sequences $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\tilde{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}$ and $\tilde{\mathbf{I}}^{\mathrm{O}}$.

For RL-Grouping-enc, the encoder groups deletions and type-E insertions according to lengths of $\bar{\mathbf{X}}$-runs, as shown in Fig. 11. Specifically, denote the maximum length of $\bar{\mathbf{X}}$-runs by $l_{\max}$, and the number of $\bar{\mathbf{X}}$-runs with length l by $n(l)$ for all $l = 1, 2, \ldots, l_{\max}$.

- *Deletions* – Let $\tilde{D}_{l,i}$ denote the number of deletions in the ith length-l $\bar{\mathbf{X}}$-run. The sequence $\tilde{\mathbf{D}}_l = \tilde{D}_{l,1}\tilde{D}_{l,2}\ldots\tilde{D}_{l,n(l)}$ represents the numbers of deletions in all $\bar{\mathbf{X}}$-runs with length l. We use $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$ to denote the set of sequences $\{\tilde{\mathbf{D}}_1, \tilde{\mathbf{D}}_2, \ldots, \tilde{\mathbf{D}}_{l_{\max}}\}$.
- *Type-E insertions* – Denote the number of type-E insertions in the ith length-l $\bar{\mathbf{X}}$-run by $\tilde{I}_{l,i}^{\mathrm{E}}$. The sequence

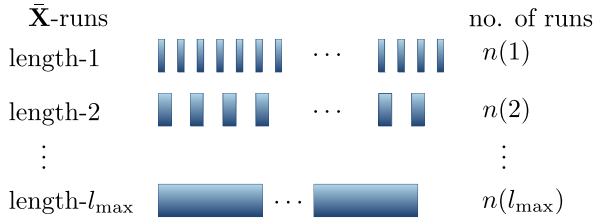

Fig. 11. Group $\bar{\mathbf{X}}$-runs by their lengths, and then count the number of deletions $\tilde{D}_{l,i}$ and the number of insertions that extend runs $\tilde{I}^{\mathrm{E}}_{l,i}$ for the ith length-l run, for all $l = 1, 2, \ldots, l_{\max}$ and $i = 1, 2, \ldots, n(l)$.

$\tilde{\mathbf{I}}^{\mathrm{E}}_l = \tilde{I}^{\mathrm{E}}_{l,1}\tilde{I}^{\mathrm{E}}_{l,2}\ldots\tilde{I}^{\mathrm{E}}_{l,n(l)}$ represents the numbers of type-E insertions in all $\bar{\mathbf{X}}$-runs with length l. We use $\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}$ to denote the set of sequences $\{\tilde{\mathbf{I}}^{\mathrm{E}}_1, \tilde{\mathbf{I}}^{\mathrm{E}}_2, \ldots, \tilde{\mathbf{I}}^{\mathrm{E}}_{l_{\max}}\}$. Note that for any ith length-l $\bar{\mathbf{X}}$-run, because DP-enc outputs edit pattern(s) with minimum number of edits, $\tilde{D}_{l,i}$ and $\tilde{I}^{\mathrm{E}}_{l,i}$ are not nonzero at the same time.

- *Type-O insertions* – Let $l(\bar{\mathbf{Y}})$ denote the length of $\bar{\mathbf{Y}}$, and i' denote the number of type-O insertions. Type-O insertions, denoted by $\tilde{\mathbf{I}}^{\mathrm{O}}$, comprises two sequences, *i.e.* $\tilde{\mathbf{I}}^{\mathrm{O}} = (\tilde{I}^{l(\bar{\mathbf{Y}})}_{\mathrm{O}}, \tilde{C}^{i'})$. The insertion pattern of type-O insertions is modeled by a length-$l(\bar{\mathbf{Y}})$ sequence $\tilde{I}^{l(\bar{\mathbf{Y}})}_{\mathrm{O}}$, where each $\tilde{I}_{\mathrm{O}}$ is either an insertion $\bar{\imath}$ or a no-operation $\bar{\eta}$. The contents of type-O insertions is modeled by a length-i' sequence $\tilde{C}^{i'}$ of symbols from the source alphabet $\mathcal{A}$.

The decoder of the DP-RLC algorithm decodes $\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}, \{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}, \tilde{\mathbf{I}}^{\mathrm{O}}\big)$ by corresponding decoders of the entropy codes in Entro-enc in Algorithm 2. Then, the decoder reconstructs $\bar{\mathbf{Y}}$ from $\big(\bar{\mathbf{X}}, \{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}, \{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}, \tilde{\mathbf{I}}^{\mathrm{O}}\big)$ by processing the InDels on $\bar{\mathbf{X}}$ in the same order as in the encoding procedure.

D. Dynamic-Programming-Run-Length Compression (DP-RLC) Scheme – Performance

Because we use entropy codes to compress the three sets of sequences $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}$ and $\tilde{\mathbf{I}}^{\mathrm{O}}$ in Step 3, the asymptotic compression rate of DP-RLC algorithm in Section IV-C is $\lim_{n\to\infty}\frac{1}{n}\Big(H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\tilde{\mathbf{I}}^{\mathrm{O}}\big)\Big)$. This entropy rate is characterized as an expansion in ϵ and δ, and the first order terms are computed explicitly and presented in Theorem 13 below.

Theorem 13: For the RPES-LtRRID model, the DP-RLC algorithm requires a compression rate of at most $\mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon\log|\mathcal{A}| - (\delta+\epsilon)C_{|\mathcal{A}|} + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)$, *for some* $\tau \in (0, 1)$.

To prove Theorem 13, we firstly show in Lemma 14 below that the entropy of sets of sequences $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}$ and $\tilde{\mathbf{I}}^{\mathrm{O}}$ is close to the entropy of three similar sets of sequences $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\bar{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}$, and $\bar{\mathbf{I}}^{\mathrm{O}}$ obtained from the original edit pattern $\bar{\mathbf{E}}$. We then compute the entropy rates of these three sets of sequences in Lemma 15-17 sequentially.

Remark: From Theorems 8 and 13, we observe that the lower bound and achievable rate match up to first order terms. Therefore, for small ϵ and δ, DP-RLC is asymptotically optimal.

Suppose we know the actual edit pattern $\bar{\mathbf{E}}$, we can group the edits in $\bar{\mathbf{E}}$ according to the lengths of $\bar{\mathbf{X}}$-runs with the same procedure in Step 2 of the DP-RLC algorithm, resulting in three similar groups of sequences, denoted by $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\bar{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}$, and $\bar{\mathbf{I}}^{\mathrm{O}}$. In Lemma 14 below, we prove that the description length required by the DP-RLC algorithm is close to the description length of these three groups of sequences corresponding to the actual edit pattern.

Lemma 14: The description length required by the DP-RLC algorithm $H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\tilde{\mathbf{I}}^{\mathrm{O}}\big)$ *is close to the description length of the groups of sequences* $H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\{\bar{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big) + H\big(\bar{\mathbf{I}}^{\mathrm{O}}\big)$ *corresponding to the actual edit pattern. Specifically,*

$$\begin{aligned}
\big|H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) - H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\big| &\le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2),\\
\big|H\big(\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big) - H\big(\{\bar{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big)\big| &\le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2),\\
\big|H\big(\tilde{\mathbf{I}}^{\mathrm{O}}\big) - H\big(\bar{\mathbf{I}}^{\mathrm{O}}\big)\big| &\le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2).
\end{aligned}$$

Proof: Note that for the same $\bar{\mathbf{X}}$ and for any l, $\bar{\mathbf{D}}_l$ has the same length as $\tilde{\mathbf{D}}_l$, because the algorithm groups edits according to the length of $\bar{\mathbf{X}}$-runs. We argue below that at most $\mathcal{O}(\max(\epsilon,\delta)^2)n$ terms are different in $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$ and $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$. For an $\bar{\mathbf{X}}$-run, the number of deletions in its output by DP may differ from the number of deletions in the original edit-pattern only if one of following two cases occurs:

Case 1: there is more than one edit in the extended run of this $\bar{\mathbf{X}}$-run;

Case 2: although there is only one edit in its extended run, the interaction of insertions and deletions in different runs leads to uncertainty in the edit pattern (recall Fig. 8 in Section III-C for an example).

Case 1 corresponds to the scenario where edits are atypical. Hence, it is straightforward that Case 1 occurs with probability at least in second order term, *i.e.,* $\mathcal{O}(\max(\epsilon,\delta)^2)$. Case 2 is more intricate. It corresponds to the ambiguous local alignment event (Definition 4). We proved in Lemma 5 that on average (over $\bar{\mathbf{X}}$ and $\bar{\mathbf{E}}$), Case 2 also occurs with probability at most $\mathcal{O}(\max(\epsilon,\delta)^2)$. From Case 1 and Case 2, on average at most $\mathcal{O}(\max(\epsilon,\delta)^2)n$ terms are different in $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$ and $\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$. Hence, the entropy of the component-wise difference is at most $H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) \le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2)$. To conclude,

$$\begin{aligned}
&\big|H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) - H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\big|\\
&\quad= \big|H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big) - H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\big|\\
&\quad= \big|H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\\
&\qquad - H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\big|\\
&\quad\le H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\\
&\qquad + H\big(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}|\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\\
&\quad\le 2\ H\big(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}} - \{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\big)\\
&\quad\le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2).
\end{aligned}$$

Similar arguments hold for type-E insertions. Hence, we also have $\big|H\big(\{\tilde{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big) - H\big(\{\bar{\mathbf{I}}^{\mathrm{E}}_l\}_{l=1}^{l_{\max}}\big)\big| \le n\cdot\mathcal{O}(\max(\epsilon,\delta)^2)$.

For type-O insertions, recall that $\tilde{\mathbf{I}}^{\mathrm{O}} = (\tilde{I}^{l(\mathbf{Y})}_{\mathrm{O}}, \tilde{C}^{i'})$ and $\bar{\mathbf{I}}^{\mathrm{O}} = (\bar{I}^{l(\mathbf{Y})}_{\mathrm{O}}, C^{i''})$, it is obvious that $\tilde{I}^{l(\bar{\mathbf{Y}})}_{\mathrm{O}}$ and $\bar{I}^{l(\bar{\mathbf{Y}})}_{\mathrm{O}}$ have

the same length. For similar reasons, they differ in at most $\mathcal{O}(\max(\epsilon,\delta)^2)n$ locations. Thus although the content of type-O insertions $\tilde{C}^{i'}$ and $C^{i''}$ may have different length, they differ by at most $\mathcal{O}(\max(\epsilon,\delta)^2)n$ entries. Therefore, we also have $|H(\tilde{\mathbf{I}}^{\mathrm{O}}) - H(\bar{\mathbf{I}}^{\mathrm{O}})| \le n \cdot \mathcal{O}(\max(\epsilon,\delta)^2)$. □

From Lemma 14, we can characterize (first order terms of) the achievable rate of DP-RLC by the entropy rates of sequences $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$, $\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}$, and $\bar{\mathbf{I}}^{\mathrm{O}}$ obtained from the original edit pattern, which we compute explicitly in the following.

Lemma 15: The asymptotic entropy rate of $\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}$ *is* $\lim_{n\to\infty}\frac{1}{n}H(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}) = \mathcal{H}(\delta) - \delta\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1} l\log l + \mathcal{O}(\max(\epsilon,\delta)^{2-\tau})$, *for some* $\tau \in (0,1)$.

Proof: The number of deletions in a length-l $\bar{\mathbf{X}}$-run is Binomial$(l, \frac{\delta}{1-\epsilon})$ distributed. Denote its probabilistic distribution by $\Pr_{\bar{\mathbf{D}}_l}$, *i.e.* $\Pr_{\bar{\mathbf{D}}_l}(d) = \Pr(\bar{\mathbf{D}}_l = d)$. We have $\Pr_{\bar{\mathbf{D}}_l}(d) = \binom{l}{d}\delta^d(1-\epsilon-\delta)^{l-d}/(1-\epsilon)^l$. Hence,

$$
\begin{aligned}
&H(\Pr_{\bar{\mathbf{D}}_l})\\
&= \sum_{d=0}^{l} -\binom{l}{d}\frac{\delta^d(1-\epsilon-\delta)^{l-d}}{(1-\epsilon)^l}\Big[\log\binom{l}{d} + d\log\delta\\
&\quad +(l-d)\log(1-\epsilon-\delta) - l\log(1-\epsilon)\Big]\\
&= -\frac{l}{1-\epsilon}\delta\log\delta - \frac{l}{1-\epsilon}(1-\epsilon-\delta)\log(1-\epsilon-\delta) +\\
&\quad l\log(1-\epsilon) + \sum_{d=0}^{l} -\binom{l}{d}\frac{\delta^d(1-\epsilon-\delta)^{l-d}}{(1-\epsilon)^l}\log\binom{l}{d}\\
&= \frac{l}{1-\epsilon}\left(\mathcal{H}(\delta) + \mathcal{O}\left(\max(\epsilon,\delta)^2\right)\right) - \frac{\delta(1-\epsilon-\delta)^{l-1}}{(1-\epsilon)^l} l\log l\\
&\quad + \sum_{d=2}^{l-1} -\binom{l}{d}\frac{\delta^d(1-\epsilon-\delta)^{l-d}}{(1-\epsilon)^l}\log\binom{l}{d}\\
&\le l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\\
&\quad + \sum_{d=2}^{l-1} -\binom{l}{d}\frac{\delta^d(1-\epsilon-\delta)^{l-d}}{(1-\epsilon)^l}\log\binom{l}{d}\\
&\quad l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\\
&\quad - \sum_{d=2}^{l-1}\binom{l}{d}\frac{\delta^d(1-\epsilon-\delta)^{l-d}}{(1-\epsilon)^l}\log l\\
&= l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\\
&\quad - \sum_{d=0}^{l-3}\binom{l}{d+2}\frac{\delta^{d+2}(1-\epsilon-\delta)^{l-d-2}}{(1-\epsilon)^l}\log l\\
&= l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\\
&\quad -\delta^2\log l\frac{(1-\epsilon-\delta)l\ (l-1)(l-2)}{(1-\epsilon)^3}\cdot\sum_{d=0}^{l-3}\binom{l-3}{d}\\
&\quad \frac{1}{(d+1)(d+2)(l-d-2)}\left(\frac{\delta}{1-\epsilon}\right)^d\left(1-\frac{\delta}{1-\epsilon}\right)^{l-d-3}\\
&\le l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\\
&\quad -\delta^2\log l\frac{(1-\epsilon-\delta)l\ (l-1)(l-2)}{(1-\epsilon)^3}\\
&\quad \cdot\sum_{d=0}^{l-3}\binom{l-3}{d}\frac{1}{(l-2)(l-1)l}\left(\frac{\delta}{1-\epsilon}\right)^d\left(1-\frac{\delta}{1-\epsilon}\right)^{l-d-3}\\
&= l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right) - \delta^2\log l\frac{(1-\epsilon-\delta)}{(1-\epsilon)^3}\\
&= l\mathcal{H}(\delta) - \delta l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right),
\end{aligned}
$$

for some $\tau \in (0,1)$. Recall that $n(l)$ denotes the number of $\bar{\mathbf{X}}$-runs with length l, we have $E[n(l)] = n\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}$,

$$
\begin{aligned}
&H(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}})\\
&= E\left[\sum_{l=1}^{l_{\max}} n(l)H(\Pr_{\bar{\mathbf{D}}_l})\right]\\
&= \sum_{l=1}^{\infty} E[n(l)]H(\Pr_{\bar{\mathbf{D}}_l})\\
&= \sum_{l=1}^{\infty} n\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}\big(l\mathcal{H}(\delta) - \delta l\log l\\
&\quad +\mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)\big)\\
&= n\mathcal{H}(\delta) - \delta n\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1} l\log l\\
&\quad +n\cdot\mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right).
\end{aligned}
$$

Hence, $\lim_{n\to\infty}\frac{1}{n}H(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}) = \mathcal{H}(\delta) - \delta\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1} l\log l + \mathcal{O}\left(\max(\epsilon,\delta)^{2-\tau}\right)$. □

Lemma 16: The asymptotic entropy rate of $\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}$ *is* $\lim_{n\to\infty}\frac{1}{n}H(\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}) = \left(\frac{2}{|\mathcal{A}|} - \frac{1}{|\mathcal{A}|^2}\right)(\mathcal{H}(\epsilon) + \epsilon\log|\mathcal{A}|) - \epsilon\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1} l\log l + \mathcal{O}(\epsilon^{2-\tau})$, *for some* $\tau \in (0,1)$.

Proof: The number of insertions in a length-l $\bar{\mathbf{X}}$-run is negative binomial NB$(l+1,\epsilon)$ distributed. Each insertion extends the run with probability $\frac{1}{|\mathcal{A}|}$. Denote the probabilistic distribution of number of insertions that extend runs in a length-l $\bar{\mathbf{X}}$-run by $\Pr_{\bar{\mathbf{I}}_l^{\mathrm{E}}}$, we have

$$\Pr_{\bar{\mathbf{I}}_l^{\mathrm{E}}}(i) = \sum_{k=i}^{\infty}\binom{k+l}{l}(1-\epsilon)^{l+1}\epsilon^k\binom{k}{i}\left(1-\frac{1}{|\mathcal{A}|}\right)^{k-i}\left(\frac{1}{|\mathcal{A}|}\right)^i.$$

With similar calculations as in Lemma 15, we have $H(\Pr_{\bar{\mathbf{I}}_l^{\mathrm{E}}}) \le \frac{(l+1)}{|\mathcal{A}|}\mathcal{H}(\epsilon) + (l+1)\frac{\epsilon}{|\mathcal{A}|}\log|\mathcal{A}| - \frac{\epsilon}{|\mathcal{A}|}(l+1)\log(l+1) + \mathcal{O}(\epsilon^{2-\tau})$ for some $\tau \in (0,1)$. Hence,

$$
\begin{aligned}
&H(\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}})\\
&= E\left[\sum_{l=1}^{l_{\max}} n(l)H(\Pr_{\bar{\mathbf{I}}_l^{\mathrm{E}}})\right]\\
&= \sum_{l=1}^{\infty} E[n(l)]H(\Pr_{\bar{\mathbf{I}}_l^{\mathrm{E}}})\\
&= \sum_{l=1}^{\infty} n\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}\Big(\frac{(l+1)}{|\mathcal{A}|}\mathcal{H}(\epsilon)\\
&\quad +\frac{(l+1)\epsilon}{|\mathcal{A}|}\log|\mathcal{A}| - \frac{\epsilon}{|\mathcal{A}|}(l+1)\log(l+1) + \mathcal{O}(\epsilon^{2-\tau})\Big)
\end{aligned}
$$

$$
\begin{aligned}
&= n\mathcal{H}(\epsilon)\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^l(l+1)\\
&\quad+n\epsilon\log|\mathcal{A}|\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^l(l+1)\\
&\quad-\epsilon n\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^l(l+1)\log(l+1)+\mathcal{O}(\epsilon^{2-\tau})n\\
&=\left(\frac{2}{|\mathcal{A}|}-\frac{1}{|\mathcal{A}|^2}\right)(n\mathcal{H}(\epsilon)+n\epsilon\log|\mathcal{A}|)\\
&\quad-\epsilon n\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}l\log l+\mathcal{O}(\epsilon^{2-\tau})n,
\end{aligned}
$$

hence, for type-E insertions, we have $\lim_{n\to\infty}\frac{1}{n}H\left(\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}\right)=\left(\frac{2}{|\mathcal{A}|}-\frac{1}{|\mathcal{A}|^2}\right)(\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|)-\epsilon\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}l\log l+\mathcal{O}(\epsilon^{2-\tau})$. □

The intuition of the coefficient $\frac{2}{|\mathcal{A}|}-\frac{1}{|\mathcal{A}|^2}$ in Lemma 16 is as follows. From the perspective of the whole sequence $\bar{\mathbf{X}}$, insertions within $\mathbf{X}$-runs extend runs with probability $\frac{1}{|\mathcal{A}|}$, and insertions between two $\bar{\mathbf{X}}$-runs extend runs with probability $\frac{2}{|\mathcal{A}|}$ (they may extend either the run on the left side or the run on the right side). On average, there are $n(1-\frac{1}{|\mathcal{A}|})$ $\bar{\mathbf{X}}$-runs. Hence, out of n possible positions for insertions, on average $n(1-\frac{1}{|\mathcal{A}|})$ of them are between two $\bar{\mathbf{X}}$-runs.[8] Hence, on average, there are $n(1-\frac{1}{|\mathcal{A}|})\cdot\epsilon\cdot\frac{2}{|\mathcal{A}|}+n\frac{1}{|\mathcal{A}|}\cdot\epsilon\cdot\frac{1}{|\mathcal{A}|}=n\epsilon\left(\frac{2}{|\mathcal{A}|}-\frac{1}{|\mathcal{A}|^2}\right)$ type-E insertions.

Based on the above observations, on average there are $n\epsilon\left(1-\frac{2}{|\mathcal{A}|}+\frac{1}{|\mathcal{A}|^2}\right)$ type-O insertions, which provides an intuition for Lemma 17 below. The proof is similar as Lemma 15 and Lemma 16 hence omitted here.

Lemma 17: The asymptotic entropy rate of $\bar{\mathbf{I}}^{\mathrm{O}}$ *is* $\lim_{n\to\infty}\frac{1}{n}H\left(\bar{\mathbf{I}}^{\mathrm{O}}\right)=\left(1-\frac{2}{|\mathcal{A}|}+\frac{1}{|\mathcal{A}|^2}\right)\mathcal{H}(\epsilon)+\left(1-\frac{2}{|\mathcal{A}|}+\frac{1}{|\mathcal{A}|^2}\right)\epsilon\log|\mathcal{A}|+\mathcal{O}(\max(\epsilon,\delta)^{2-\tau})$, *for some* $\tau\in(0,1)$.

Combining Lemma 14-17, we have that the asymptotic compression rate of DP-RLC algorithm is bounded by

$$
\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\left(H\left(\{\tilde{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\right)+H\left(\{\tilde{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}\right)+H\left(\tilde{\mathbf{I}}^{\mathrm{O}}\right)\right)\\
&\le\lim_{n\to\infty}\frac{1}{n}H\left(\{\bar{\mathbf{D}}_l\}_{l=1}^{l_{\max}}\right)+\lim_{n\to\infty}\frac{1}{n}H\left(\{\bar{\mathbf{I}}_l^{\mathrm{E}}\}_{l=1}^{l_{\max}}\right)\\
&\quad+\lim_{n\to\infty}\frac{1}{n}H\left(\bar{\mathbf{I}}^{\mathrm{O}}\right)+\mathcal{O}(\max(\epsilon,\delta)^2)\\
&\le\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|-\\
&\quad(\delta+\epsilon)\sum_{l=1}^{\infty}\left(1-\frac{1}{|\mathcal{A}|}\right)^2\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}l\log l+\mathcal{O}(\max(\epsilon,\delta)^{2-\tau}),
\end{aligned}
$$

for some $\tau\in(0,1)$, hence proved Theorem 13.

[8] In fact, there are $n+1$ possible positions for insertion, and on average $n(1-\frac{1}{|\mathcal{A}|})-1$ positions are between two $\mathbf{X}$-runs. However, asymptotically as n grows, these boundary effects are negligible.

V. Conclusion and Discussion

We investigate one-way file updates problem, specifically the communication complexity to enable file updates with small fractions of insertions and deletions (InDels). We study two models, one worstcase model with arbitrary source sequence and arbitrary InDels, the other stochastic model with random source sequence and random InDels. For the arbitrary model, we derive an information-theoretic lower bound on the communication complexity in Theorem 9. We show in Theorem 12 that a simple scheme combining dynamic programming and entropy coding (DP-EC) achieves a compression rate that is close to the lower bound up to first order terms. For the stochastic model, we show that the minimum communication complexity to enable file updates is the description length of the edit pattern, less a term called nature's secret, and characterize the optimal rate up to first order terms in Theorem 8. We show that the DP-EC algorithm achieves a compression rate which matches the description length of the edit sequence in Theorem 11. Hence the DP-EC algorithm performs well regardless of the latent model. We also provide a run-length compression (DP-RLC) scheme specifically for the stochastic model. We show in Theorem 13 that the DP-RLC scheme achieves a compression rate which matches the lower bound (Theorem 8) in all first order terms. Hence, DP-RLC scheme outperforms DP-EC scheme for the stochastic model.

There are potentially many ways to model stochastic InDels. Our results should in general translate over to those models in the regime with small fractions of insertions and deletions. In Section V-A below, we discuss some other stochastic InDel models in the literature and some potential models for further investigation.

A. Different Stochastic InDel Processes

A general left-to-right Markov InDel process as shown in Fig. 12 might be of interest for future study. In this work, we set $\alpha_1=\alpha_4=\delta$, $\alpha_2=\alpha_5=\epsilon$ and $\alpha_3=\alpha_6=1-\epsilon-\delta$, resulting in i.i.d. insertions and deletions. The model was also studied in [7] as a channel with synchronization errors. The difference is that the authors of [7] imposed a maximum insertion length, and set the insertion and deletion probabilities to be equal, to keep the expected length the sequence unchanged after processing the edits. We don't impose these two requirements in our model. The authors in [7] proposed a block code which is a concatenation of a "watermark" code and a LDPC code for this synchronization error channel, and presented the empirical performance of their code.

Another stochastic model, possibly more realistic for human editing behavior, is to allow and embed the randomness of the "cursor" jumping back and forth. This InDel process can also be modeled as a three-state Markov chain. Fig. 13 shows a special case where with uniform cursor jump: at each iteration, the cursor jumps to a position which is uniformly distributed in the current sequence, deletes the symbol in front with probability p_D, or inserts a symbol uniformly drawn from the alphabet $\mathcal{A}$ with probability $p_I=1-p_D$. We believe our approach will derive similar results for this model, because

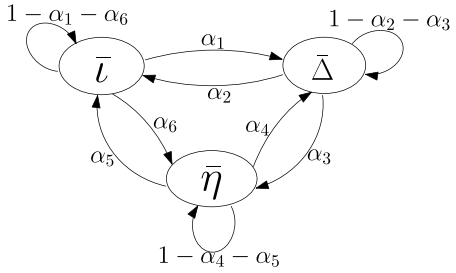

Fig. 12. A general left-to-right markov InDel process.

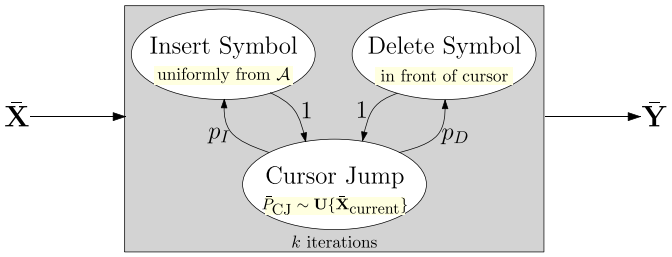

Fig. 13. A stochastic InDel model with random cursor jumps.

the probability of the insertion-deletion interaction is of order $\mathcal{O}(\epsilon\delta)$, which contributes to second order terms. Such a model typically ends up generating "sparse isolated edits". A more sophisticated stochastic model, better presenting realistic edit scenarios, would have a distribution on the cursor jump, and also a distribution on the run-length of insertions and deletions – this is the subject of ongoing investigation.

Since an insertion process can be regarded as the inverse of a deletion process, a random InDel process as in Fig. 14 was studied in [9]. The authors in [9] also considered the edit operation substitution. Here we hide the part corresponding to the substitution process to represent just the InDel process. In Fig. 14, an auxiliary sequence $\bar{\mathbf{Z}} \in \mathcal{A}^n$ is a length-n sequence of symbols drawn i.i.d. uniformly at random from the source alphabet $\mathcal{A}$. Sequences $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ are generated from $\bar{\mathbf{Z}}$ through two i.i.d. deletion processes with deletion probability p_I and p_D respectively. Hence, $\bar{\mathbf{X}}$ is a variable length (Binomial(n, $1-\mathrm{p_I}$)) sequence of i.i.d. symbols from $\mathcal{A}$. The authors in [9] proposed and algorithm which is asymptotically optimal for small insertion and deletion probability. More specifically, their algorithm is $\mathcal{O}(\max(p_I, p_D)^{2-\tau})$ far from optimal $\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$.[9] However, they didn't derive the explicit expression for the term $\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ for the InDel process. For the case with only deletions, the authors of [9] do have an information-theoretic lower bound in their earlier work [13]. One of our main effort is indeed to characterize the explicit expression of the optimal rate.

There are also many different stochastic insertion/deletion model in the literature regarding insertion/deletion channels. A random InDel model where each source bit/symbol is deleted with probability p_D, or with an extra bit/symbol inserted after it with probability p_I, or transmitted/kept (no deletion or insertion after) with probability $1-p_D-p_I$ was studied in both [12] and [35]. In [35], capacity lower bounds for channels modeled as this InDel process are proposed. In [12], an algorithm for two-way file synchronization under non-binary non-uniform source alphabet was proposed. The Gallager model [36], also studied in [37], is an InDel channel where each transmitted bit independently gets deleted with probability p_D or replaced with two random bits with probability p_I.

[9]As opposite to [9], in our paper we use $\bar{\mathbf{X}}$ for the side-information and $\bar{\mathbf{Y}}$ for the sequence to be synchronized.

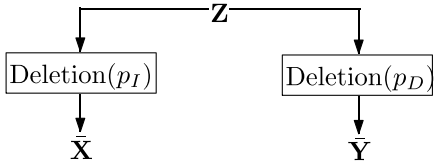

Fig. 14. The stochastic InDel model studied in [9].

APPENDIX A
PROOF OF LEMMA 4

Recall that $\bar{\mathbf{E}} = (\bar{O}^{n+K_\iota}, \bar{C}^{K_\iota})$, where $\bar{O}^{n+K_\iota}$ is an i.i.d. sequence with $P(\bar{O}_1=\bar{\iota})=\epsilon$, $P(\bar{O}_1=\bar{\Delta})=\delta$ and $P(\bar{O}_1=\bar{\eta})=1-\epsilon-\delta$. Hence,

$$\begin{aligned}
H\left(\bar{O}_1\right) &= -\delta\log\delta-\epsilon\log\epsilon-(1-\epsilon-\delta)\log(1-\epsilon-\delta)\\
&= \mathcal{H}(\delta)+\mathcal{H}(\epsilon)+(1-\delta)\log(1-\delta)+\\
&\quad (1-\epsilon)\log(1-\epsilon)-(1-\epsilon-\delta)\log(1-\epsilon-\delta)\\
&\overset{(a)}{=} \mathcal{H}(\delta)+\mathcal{H}(\epsilon)+(1-\delta)(\log e)\left(-\delta-\frac{\delta^2}{2}-\mathcal{O}(\delta^3)\right)\\
&\quad +(1-\epsilon)(\log e)\left(-\epsilon-\frac{\epsilon^2}{2}-\mathcal{O}(\epsilon^3)\right)-(1-\delta-\epsilon)\\
&\quad \cdot(\log e)\left[-(\delta+\epsilon)-\frac{(\delta+\epsilon)^2}{2}-\mathcal{O}((\delta+\epsilon)^3)\right]\\
&= \mathcal{H}(\delta)+\mathcal{H}(\epsilon)-\epsilon\delta\log e+\mathcal{O}(\max(\epsilon,\delta)^3), \qquad (8)
\end{aligned}$$

where step (a) is by Taylor series expansion. Hence,

$$\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{E}})\\
&= \lim_{n\to\infty}\frac{1}{n}\left[H\left(\bar{O}^{n+K_\iota}\right)+H\left(\bar{C}^{K_\iota}|\bar{O}^{n+K_\iota}\right)\right]\\
&\overset{(a)}{=} \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])\,H\left(\bar{O}_1\right)+H\left(\bar{C}^{K_\iota}|\bar{O}^{n+K_\iota}\right)\right]\\
&\overset{(b)}{=} \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])\,H\left(\bar{O}_1\right)+H\left(\bar{C}^{K_\iota}|K_\iota\right)\right]\\
&= \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])H\left(\bar{O}_1\right)+\sum_{k=0}^{\infty}H\left(\bar{C}^{K_\iota}|K_\iota=k\right)\Pr(K_\iota=k)\right]\\
&= \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])H\left(\bar{O}_1\right)+\sum_{k=0}^{\infty}H\left(\bar{C}^{k}\right)\Pr(K_\iota=k)\right]\\
&= \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])H\left(\bar{O}_1\right)+\sum_{k=0}^{\infty}kH(\bar{C}_1)\Pr(K_\iota=k)\right]\\
&= \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_\iota])H\left(\bar{O}_1\right)+H(\bar{C}_1)\sum_{k=0}^{\infty}k\Pr(K_\iota=k)\right]
\end{aligned}$$

$$\stackrel{(c)}{=} \lim_{n\to\infty}\frac{1}{n}\left[(n+E[K_l])H\left(\bar{O}_1\right)+E[K_l]H(\bar{C}_1)\right]$$
$$\stackrel{(d)}{=} \lim_{n\to\infty}\frac{1}{n}\left[\frac{n+\epsilon}{1-\epsilon}H\left(\bar{O}_1\right)+(n+1)\frac{\epsilon}{1-\epsilon}\log|\mathcal{A}|\right]$$
$$= \frac{1}{1-\epsilon}\left(H\left(\bar{O}_1\right)+\epsilon\log|\mathcal{A}|\right)$$
$$\stackrel{(e)}{=} \frac{1}{1-\epsilon}\Big(\mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|-\epsilon\delta\log e+\mathcal{O}(\max(\epsilon,\delta)^3)\Big)$$
$$\stackrel{(f)}{=} \mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|-\epsilon\delta\log\delta-\epsilon^2\log\epsilon$$
$$+(\log e+\log|\mathcal{A}|)\epsilon^2+\mathcal{O}\left(\max(\epsilon,\delta)^3\right)$$
$$\geq \mathcal{H}(\delta)+\mathcal{H}(\epsilon)+\epsilon\log|\mathcal{A}|+2\min(\epsilon,\delta)^{2-\tau}+\mathcal{O}(\max(\epsilon,\delta)^2),$$

for some $\tau\in(0,1)$. Equality (a) is because $n+K_l$ is a determined stopping time for the i.i.d. edit sequence $\bar{O}_1,\bar{O}_2,\ldots$, hence by Theorem 1, $H\left(\bar{O}^{n+K_l}\right)=(n+E[K_l])H\left(\bar{O}_1\right)$. Equality (b) is because given the edit operation sequence $\bar{O}^{n+K_l}$, the insertion content sequence $\bar{C}^{K_l}$ depends only on the number of insertions K_l. From equality (b) to equality (c) is by expanding K_l and noting that $\bar{C}^{K_l}$ is a sequence of i.i.d. variables. Equality (d) is by $E[K_l]=(n+1)\frac{\epsilon}{1-\epsilon}$ and noting that the contents of insertions are uniformly drawn from the alphabet. Equality (e) is by equation (8). Equality (f) is by taking the Taylor series expansion of $\frac{1}{1-\epsilon}$, $\mathcal{H}(\delta)$ and $\mathcal{H}(\epsilon)$.

APPENDIX B
PROOF OF LEMMA 5

The intuition that the uncertainty $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ of the global alignment is small is as follows. When an ambiguous local alignment event occurs, of either type-1 or type-2, denoted by $\Gamma=\Gamma^1\cup\Gamma^2$ (recall Definition 4), one possible edit pattern has an insertion and the other has a deletion. Hence, "locally" the positions of the output $\hat{\mathbf{Y}}$ by applying these two ambiguous edit patterns to $\bar{\mathbf{X}}$ differ by a shift of two symbols. Conversely, the sections of $\bar{\mathbf{X}}$ which leads to the same section of $\hat{\mathbf{Y}}$ through the two ambiguous edit patterns differ by two symbols. If the align module described in Fig. 15 is able to keep aligning $\bar{\mathbf{X}}$ w.r.t. $\hat{\mathbf{Y}}$ via both edit patterns, the local ambiguity is not resolved. That means we can find at least two distinct typicalized edit patterns that convert two "similar" sections of $\bar{\mathbf{X}}$ which differ by two symbols to the same section of $\hat{\mathbf{Y}}$. This means that some symbols (it turns out at least one out of every two neighbouring symbols) in one section of $\bar{\mathbf{X}}$ determine the values of other symbols in $\bar{\mathbf{X}}$ within a short block. This is because of the property of typicalized edits that not too many insertions or deletions (hence no contiguous insertions or deletions) can happen in a short block. Hence, averaging over $\bar{\mathbf{X}}$, the probability that extra information is needed to resolve ambiguous local alignments is small.

Given a specific PreESS $\bar{\mathbf{x}}$, denote the number of $\bar{\mathbf{x}}$-runs by $\rho_{\bar{\mathbf{x}}}$. For $i=1,2,\ldots,\rho_{\bar{\mathbf{x}}}$, we define the following quantity *gap* and a related event from the align module (Fig. 15):

- Gap $G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}$: If after typicalizing $\bar{\mathbf{e}}$ to $\hat{\mathbf{e}}$ and processing $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$, the ith $\bar{\mathbf{x}}$-run encounter an ambiguous local alignment, the gap – denoted by $G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}$ – is the length of the subsequence starting from the first symbol after the ith $\bar{\mathbf{x}}$-run and ending at the symbol before where the next edit in $\hat{\mathbf{e}}$ applies to.
- Event of ambiguity unresolved within gap $\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}$ – after typicalizing $\bar{\mathbf{e}}$ to $\hat{\mathbf{e}}$ and processing $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$, the ith $\bar{\mathbf{x}}$-run encounter an ambiguous local alignment, and within the gap, the ambiguous edit pattern at the ith $\bar{\mathbf{x}}$-run can obtain the same $\hat{\mathbf{y}}$ through some typical edits.

Conditioning on the event that an ambiguous local alignment event occurs to the ith $\bar{\mathbf{x}}$-run, denoted by Γ_i, and the gap $G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g$, the probability $\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i,G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)$ depends only on $\bar{\mathbf{x}}$ and g. We denote the probability $\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i,G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)$ averaged over $\bar{\mathbf{X}}$ by $\Pr_g=\sum_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\Pr(\Gamma_i,G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)$ and bound $\Pr_g$ from above through case analysis. From the definition, $\Pr_g$ is the probability that averaging over $\bar{\mathbf{x}}$, conditioning on the occurrence of an ambiguous local alignment, the probability that ambiguity is still not resolved by continuing the align module after reaching the gap g. Note that $\Pr_g$ is an upper bound on the probability that a path on the alignment tree $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ splits into two branches at a node, averaging over all possible $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$.

We break into four cases based on the type of the ambiguous local alignment and the edit pattern that actually happens. For example, $\Gamma^1(\bar{\imath})$ denotes an ambiguous local alignement of type-1 ($l_{\bar{\mathbf{X}}}=l_{\hat{\mathbf{Y}}}-1$), where the actual edit pattern has an insertion $\bar{\imath}$ (recall Definition 4). We provide detailed analysis on case $\Gamma^1(\bar{\imath})$. The other three cases are similar.

A. Ambiguous Local Alignment Type-1 Γ^1 ($l_{\bar{\mathbf{X}}}=l_{\hat{\mathbf{Y}}}-1$)

W.l.o.g., assume the symbol in the run is 0 and the subsequence of $\bar{\mathbf{X}}$ starting from the run is $0x_1x_2x_3\ldots$. The corresponding $\hat{\mathbf{Y}}$-run to be aligned is 00. There are two possibilities: 1) Case $\Gamma^1(\bar{\imath})$ – this case corresponds to an edit pattern like $0^{\downarrow 0}x_1\cdots\to 00x_1\ldots$, with an insertion of 0. 2) Case $\Gamma^1(\bar{\Delta})$ – this case corresponds to the edit pattern that x_1 is deleted and 0 combines with x_2, which must be 0, resulting in 00. That is, $0\cancel{x_1}0x_3\cdots\to 00x_3\ldots$. If x_2 is not 0, this edit pattern is impossible and the ambiguity is resolved. Averaging over $p(\bar{\mathbf{x}})$, the event $x_2=0$ happens with probability $\frac{1}{|\mathcal{A}|}$. Moreover, this edit pattern results in either $0\cancel{x_1}0x_3\cdots\to 00x_3\ldots$ (if x_3 is not deleted), or $0\cancel{x_1}0\cancel{x_3}x_4\cdots\to 00x_4\ldots$ (if x_3 is also deleted). Hence, the local ambiguous event happens only if either x_3 or x_4 is the same as x_1, which happens with probability

$$1-\left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2=\frac{2|\mathcal{A}|-1}{|\mathcal{A}|^2}. \tag{9}$$

Otherwise, by checking the next symbol in $\hat{\mathbf{y}}$ after 00, one can figure out which case actually happened.

The above initial analysis shows the intuition that only some $\bar{\mathbf{x}}$ with certain structure will cause ambiguity unresolved. In the following, we further analyse the necessary condition for ambiguity unresolved after reaching the gap g.

1) Case $\Gamma^1(\bar{\imath})$: The actual edit in $\hat{\mathbf{e}}$ is a single insertion $\bar{\imath}$, and until the gap g there is no other edit:

$$0^{\downarrow 0}x_10x_3x_4x_5\ldots x_g\to 00x_10x_3x_4x_5\ldots x_g$$

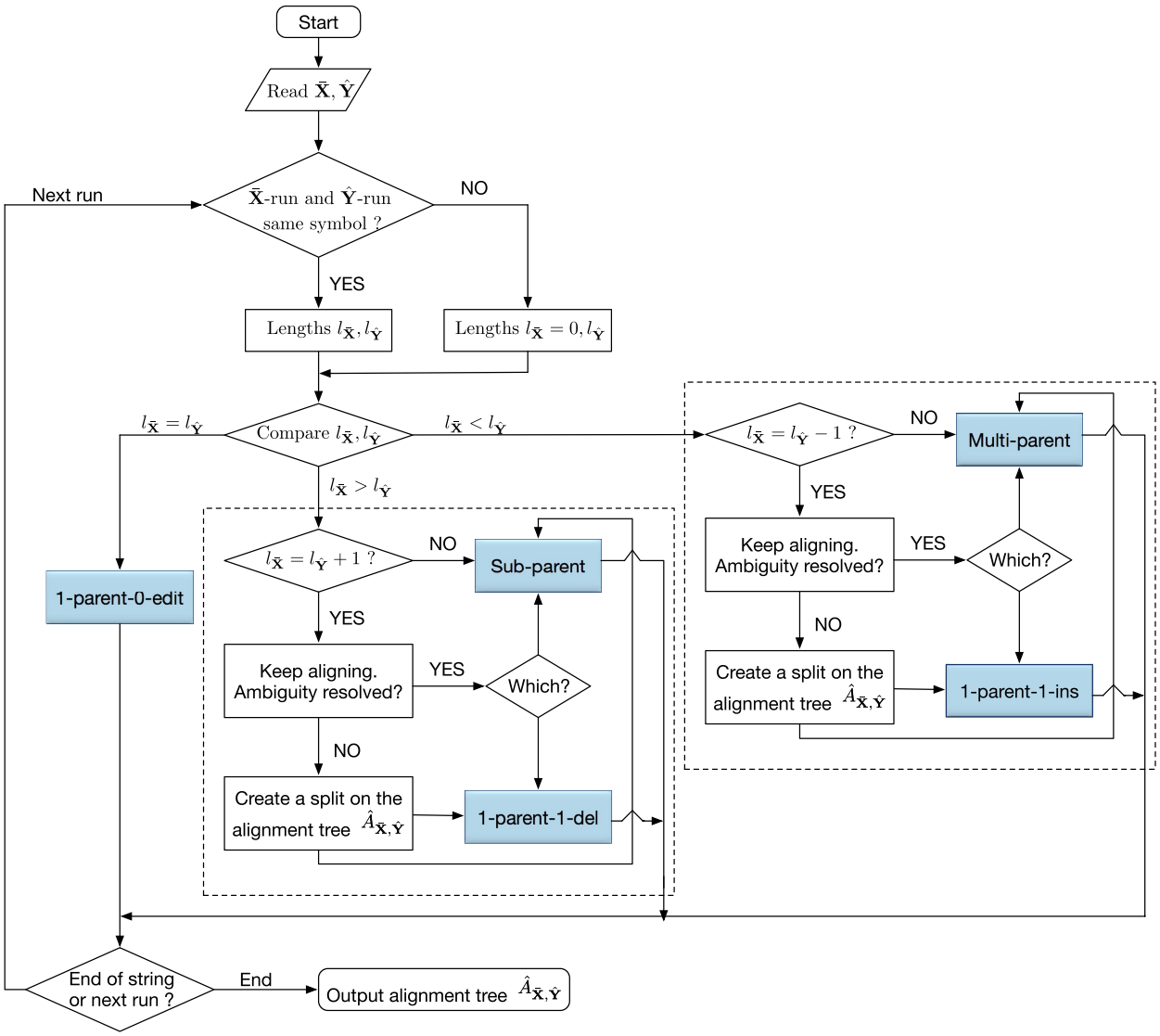

Fig. 15. The flowchart of the *align module* to align $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$: The module takes in $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ as inputs, and outputs all the possible alignments $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ as a binary tree of depth $\rho_{\hat{\mathbf{Y}}}$. Any path of the output tree of length $\rho_{\hat{\mathbf{Y}}}$ is a global alignment of $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ as defined in Definition 5; any partial path starting from the root of the tree with length $l_{P_{\hat{A}}} \leq \rho_{\hat{\mathbf{Y}}}$ is a partial alignment upto depth $l_{P_{\hat{A}}}$ as defined in Definition 6. In the process of aligning $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$, when an ambiguous local alignment occurs, the process keeps both edit patterns and continues aligning further runs with both alignments – this leads to new loops of the algorithm and possible new branches/splits on the tree $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ if the ambiguity is not resolved by aligning further runs.

In this case, the smallest g is 1. Let $g = 2t - 1$ or $2t$ depending on whether g is odd or even, where $t = 1, 2, \ldots$. The ambiguous edit has deletion of x_1 and should also result in the same $\hat{\mathbf{y}}$ through some typical edits, that is:

$$0\cancel{x_1}0x_3\ x_4\ x_5 \ldots x_g \cdots \rightarrow 00x_3\ x_4\ x_5 \ldots x_g \ldots$$
$$\xrightarrow{\text{some typical edits}} 00x_1 0\ x_3\ x_4\ x_5 \ldots x_g$$

The symbol x_1 can equal any symbol from the alphabet but 0, w.l.o.g. assume $x_1 = 1$. If ambiguity is not resolved, there should be some typical edits, which after applied to the sequence $x_3\ x_4\ x_5 \ldots x_g \ldots$, the first g symbols of the resulting sequence should be $10\ x_3\ x_4\ x_5 \ldots x_g$ – shifting two positions rightwards. In the following, we show that averaging over $\Pr(\bar{\mathbf{x}})$, the probability that one can find some typical edits that shift a sequence rightwards by two positions and match up to length g decays with g. Recall that these $\bar{\mathbf{x}}$'s are the ones that may have splits in the tree $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ along the paths with the $\hat{\mathbf{e}}$ we are considering now.

We first argue that shifting rightwards of two positions cannot be accomplished before reaching the gap g. Firstly, typical edits only shift the sequence by one position at a time, because in typicalized edit pattern no contiguous edits can happen. Before the sequence is shifted rightwards by two positions, it must have been shifted rightwards by one position by an insertion. After this insertion, all the symbols after the insertion up to x_g must be the same and no other edits can happen (the symbols form a run). For example $x_3^{\downarrow 0} x_4\ x_5 \ldots x_g \cdots \rightarrow 10x_3\ x_4\ x_5 \ldots x_g$, the insertion of 0 shifts the sequence rightwards by one position. Because x_3 cannot be deleted, x_3 has to equal 1. Hence we have $1^{\downarrow 0} x_4\ x_5 \ldots x_g \cdots \rightarrow 101\ x_4\ x_5 \ldots x_g$. Also, x_4 has to equal 1.

Because for typicalized edit patterns, x_4 can neither be deleted nor can an insertion happen in front of x_4. By continuing the deduction, the symbols $\{x_4, x_5, \dots x_g\}$ should all equal $x_3 = 1$ and there can be no other edits among them because they form a run. Hence, for the ambiguous edit pattern, the sequence which leads to $10x_3\ x_4 \dots x_g$ after some typicalized edits, must include some more symbols after $x_3\ x_4 \dots x_g$.

We prove an upper bound on $\Pr_g$ by induction. Firstly, recall from equation (9) that to have ambiguity unresolved up to matching PosESS $00x_1$, either x_3 or x_4 has to be equal to $x_1 = 1$. Hence for we have the initial condition that when $g = 1$, $\Pr_{g=1} = 1 - \left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2 = \frac{2|\mathcal{A}|-1}{|\mathcal{A}|^2}$. Next, we establish an inductive step by bounding $\Pr_{g+2}$ based on $\Pr_g$. Assume for odd number $g = 2t-1$ where $t = 1, 2, \dots$, the sequence $x_3\ x_4\ x_5 \dots x_g \dots$ can be converted to $10\ x_3\ x_4\ x_5 \dots x_g$ – a shift rightwards by two positions up to the gap g. We look for the condition to make the shifted sequence to be able to match up to the gap $g + 2 = 2t + 1$. As argued above, the position(index) of the sequence cannot shift rightwards by two before the gap, the segment of sequence that results $10\ x_3\ x_4\ x_5 \dots x_g$ via some typical edits ends at index at least $g + 1$. If the index is $g + 1$, that is, $x_3\ x_4\ x_5 \dots x_{g+1}$ converts to $10\ x_3\ x_4\ x_5 \dots x_g$. From the above, to match two more symbols, *i.e.,* to allow two typicalized edit patterns resulting in the same segment of PosESS up to gap $g+2$, the scenario must be $x_{g+3} = x_{g+2} = x_{g+1}$ with probability $\frac{1}{|\mathcal{A}|^2}$. If the index is greater than $g + 1$, for example $g + 2$, *i.e.,* $x_3\ x_4\ x_5 \dots x_{g+2}$ converts to $10\ x_3\ x_4\ x_5 \dots x_g$, then among $x_{g+3}x_{g+4}$, at least one of them should be the same symbol as x_{g+1} or x_{g+2}. For other cases, $x_{g+1}x_{g+2}$ always determine some symbol on the right-hand side. By conditioning on whether x_{g+1} and x_{g+2} are equal, the probability is

$$
\begin{aligned}
&\Pr(\text{ambiguity unresolved till} g+2 | \text{ambiguity unresolved till } g) \\
&\le \Pr(\text{one of } x_{g+3}x_{g+4} \text{ is the same as one of } x_{g+1}x_{g+2}) \\
&= \Pr(x_{g+1}=x_{g+2})\Big(1-\Pr(x_{g+3} \neq x_{g+1})\Pr(x_{g+4} \neq x_{g+1})\Big) \\
&\quad + \Pr(x_{g+1}\neq x_{g+2})\Big(1 - \Pr(x_{g+3} \neq x_{g+1}, x_{g+3} \neq x_{g+2}) \cdot \\
&\quad \Pr(x_{g+4} \neq x_{g+1}, x_{g+4} \neq x_{g+2})\Big) \\
&= \frac{1}{|\mathcal{A}|} \cdot \left(1 - \left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2\right) + \frac{|\mathcal{A}|-1}{|\mathcal{A}|} \cdot \left(1 - \left(\frac{|\mathcal{A}|-2}{|\mathcal{A}|}\right)^2\right) \\
&= \frac{4|\mathcal{A}|^2 - 6|\mathcal{A}| + 3}{|\mathcal{A}|^3} < 1 \qquad (10)
\end{aligned}
$$

Hence, from the initial condition and the inductive step, we have $\Pr_{2t+1} \le \frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3} \cdot \Pr_{2t-1}$. For even numbers $g = 2t$ where $t = 1, 2, \dots$, we can bound the probability $\Pr_g = \Pr_{2t}$ from above by $\Pr_{2t-1}$. Hence, we have $\Pr_g \le \frac{2|\mathcal{A}|-1}{|\mathcal{A}|^2} \cdot \left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t-1$ or $2t$, where $t = 1, 2, \dots$.

2) Case $\Gamma^1(\bar{\Delta})$*:* The actual edit in $\hat{\mathbf{e}}$ is a deletion $\bar{\Delta}$ of x_1, and until the gap g there is no other edit:

$$0\cancel{x_1}0x_3x_4x_5 \dots x_g \to 00x_3x_4x_5 \dots x_g$$

In this case, x_3 can be deleted hence the smallest g is 2. We denote $g = 2t$ or $2t + 1$, where $t = 1, 2, \dots$. The ambiguous edit has a single insertion of 0 in the run of 0's and should also result in the same $\hat{\mathbf{y}}$ through some typical edits:

$$0^{\downarrow 0}x_1 0\ x_3\ x_4\ x_5 \dots x_g \cdots \to 00x_1 0\ x_3\ x_4\ x_5 \dots x_g \dots \xrightarrow{\text{some typical edits}} 00\ x_3\ x_4\ x_5 \dots x_g$$

W.l.o.g., assume $x_1 = 1$. From the above, there should be some typical edits such that, after applying these edits to the sequence $10\ x_3\ x_4\ x_5 \dots x_g \dots$, the first $g-2$ symbols of the resulting sequence should be $x_3\ x_4\ x_5 \dots x_g$, that is, a shift leftwards of two positions.

With similar arguments as in Case $\Gamma^1(\bar{\imath})$, the position(index) of the sequence cannot shift leftwards by two positions to match the index of $\hat{\mathbf{y}}$ before reaching the gap. For the initial condition, $\Pr_2 = 1$ and $\Pr_3 = \frac{1}{|\mathcal{A}|}$. By induction, for even numbers $g = 2t$ where $t = 1, 2, \dots$, $\Pr_{g+2} = \Pr_{2t+2} \le \frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3} \cdot \Pr_{2t}$. For odd numbers $g = 2t + 1$ where $t = 1, 2, \dots$, we can bound the probability $\Pr_g = \Pr_{2t+1}$ from above by $\Pr_{2t}$. Hence we have $\Pr_g \le \left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t$ or $2t + 1$ where $t = 1, 2, \dots$.

B. Ambiguous Local Alignment Γ^2 *($l_{\bar{\mathbf{x}}} = l_{\hat{\mathbf{y}}} + 1$)*

W.l.o.g., assume the symbol in the run is 0 and the subsequence of $\bar{\mathbf{x}}$ starting from the run is $00x_1x_2x_3 \dots$. The corresponding $\hat{\mathbf{y}}$-run to be aligned is 0. There are two possibilities: 1) Case $\Gamma^2(\bar{\Delta})$ – this corresponds to edit patterns like $0\emptyset x_1 \cdots \to 0x_1 \dots$, with a deletion of 0 in the run. 2) Case $\Gamma^2(\bar{\imath})$ – this corresponds to edit patterns with an insertion of a symbol other than 0 in front of the last 0 in the run, breaking the $\bar{\mathbf{x}}$-run into two runs of 0's, with length $(l_{\bar{\mathbf{x}}}-1)$ and length 1 respectively. In this case, $0^{\downarrow \bar{\imath}}0x_1 \cdots \to 0\bar{\imath}0\ x_1 \dots$.

1) Case $\Gamma^2(\bar{\Delta})$*:* The actual edit pattern $\hat{\mathbf{e}}$ has a single deletion $\bar{\Delta}$, and until gap g there is no other edit:

$$0\emptyset x_1x_2x_3 \dots x_g \to 0x_1x_2x_3 \dots x_g$$

In this case, the smallest g is 1. Denote $g = 2t-1$ or $2t$, where $t = 1, 2, \dots$. The ambiguous edit pattern has an insertion of x_1 in front of the last 0 and should also results in the same section in $\hat{\mathbf{y}}$ through some typical edits:

$$0^{\downarrow x_1}0x_1\ x_2\ x_3 \dots x_g \cdots \to 0x_1 0x_1\ x_2\ x_3 \dots x_g \dots \xrightarrow{\text{some typical edits}} 0x_1\ x_2\ x_3 \dots x_g$$

W.l.o.g., assume $x_1 = 1$. From the above, there should be some typical edits, after applying which to the sequence $01\ x_2\ x_3\ x_4 \dots x_g \dots$, the first $g-1$ symbols of the resulting sequence are $x_2\ x_3\ x_4 \dots x_g$ – a shift leftwards of two positions.

This is similar as Case $\Gamma^1(\bar{\Delta})$, *i.e.,* a shift leftwards of two positions. The only difference here is that the length of the sequence needed to match after the shift is $g-1$ instead of $g-2$. In this case, we have $\Pr_g \le \left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t-1$ or $2t$ where $t = 1, 2, \dots$.

2) Case $\Gamma^2(\bar{\imath})$*:* The actual edit pattern $\hat{\mathbf{e}}$ has an insertion of a symbol other than 0 in front of the last 0 in the run, and until gap g there is no other edit:

$$0^{\downarrow\bar{\imath}}0x_1x_2x_3\ldots x_g \to 0\bar{\imath}0x_1x_2x_3\ldots x_g$$

In this case, the smallest g is 1. Denote $g = 2t-1$ or $2t$, where $t = 1, 2, \ldots$. The ambiguous edit pattern has a single deletion of 0 in the run and should also results in the same section of $\hat{\mathbf{y}}$ through some typical edits:

$$0\emptyset x_1\ x_2\ x_3\ldots x_g\cdots \to 0\ x_1\ x_2\ x_3\ldots x_g\ldots \xrightarrow{\text{some typical edits}} 0\bar{\imath}0\ x_1\ x_2\ x_3\ldots x_g$$

The ambiguity exists only if the inserted symbol $\bar{\imath}$ equals x_1. W.l.o.g., assume $\bar{\imath} = x_1 = 1$. From the above, there should be some typical edits, after applying which to the sequence $x_2\ x_3\ldots x_g\ldots$, the first $g+1$ symbols of the resulting sequence should be $01\ x_2\ x_3\ldots x_g$ – a shift rightwards of two positions.

This is similar as Case $\Gamma^1(\bar{\imath})$ – a shift rightwards of two positions. The only difference here is the length of sequence needed to match after the shift is $g+1$ instead of g. In this case, we have $\Pr_g \le \frac{4|\mathcal{A}|-4}{|\mathcal{A}|^2}\cdot\left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t-1$ or $2t$ where $t = 1, 2, \ldots$.

From the above case analysis, for all four cases, we have $\Pr_g \le \left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t-1$ or $g = 2t$ where $t = 1, 2, \ldots$. In the following, we bound $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ from above.

$$\begin{aligned}
&H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})\\
&\overset{(a)}{=} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\hat{\mathbf{y}}}\Pr(\hat{\mathbf{y}}|\bar{\mathbf{x}})H(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})\\
&\overset{(b)}{=} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\hat{\mathbf{y}}}\left(\sum\nolimits_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\to\hat{\mathbf{y}}\}}\Pr(\bar{\mathbf{e}})\right)H(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}})\\
&\overset{(c)}{\le} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\hat{\mathbf{y}}}\left(\sum\nolimits_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\to\hat{\mathbf{y}}\}}\Pr(\bar{\mathbf{e}})\right)\\
&\quad\cdot\sum\nolimits_{P_{\hat{A}}}\frac{\sum_{\{\hat{\mathbf{e}}:P_{\hat{A}}\}}\sum_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\}}\Pr(\bar{\mathbf{e}})}{\sum_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\to\hat{\mathbf{y}}\}}\Pr(\bar{\mathbf{e}})}\cdot N_s(P_{\hat{A}})\\
&\overset{(d)}{=} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\hat{\mathbf{y}}}\sum\nolimits_{P_{\hat{A}}}\left(\sum\nolimits_{\{\hat{\mathbf{e}}:P_{\hat{A}}\}}\sum\nolimits_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\}}\Pr(\bar{\mathbf{e}})\right)\\
&\quad\cdot N_s(P_{\hat{A}})\\
&\overset{(e)}{=} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\hat{\mathbf{y}}}\sum\nolimits_{P_{\hat{A}}}\sum\nolimits_{\{\hat{\mathbf{e}}:P_{\hat{A}}\}}\sum\nolimits_{\{\bar{\mathbf{e}}:(\bar{\mathbf{x}},\bar{\mathbf{e}})\to\hat{\mathbf{e}}\}}\Big(\Pr(\bar{\mathbf{e}})\\
&\quad\cdot N_s\left(P_{\hat{A}}(\bar{\mathbf{x}},\bar{\mathbf{e}})\right)\Big)\\
&\overset{(f)}{=} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\cdot N_s(P_{\hat{A}}(\bar{\mathbf{x}},\bar{\mathbf{e}}))\\
&\overset{(g)}{\le} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\mathbb{1}_{\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}}\\
&= \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}})\\
&= \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum\nolimits_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\sum_{g=1}^{\infty}\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\\
&\quad\cdot\Pr(\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\\
&= \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\sum_{g=1}^{\infty}\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\\
&\quad\cdot\left(\sum\nolimits_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\Pr(\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\right)\\
&\overset{(h)}{\le} \sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\sum_{g=1}^{\infty}\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)(l_i+1)\max(\epsilon,\delta)^2\\
&= \max(\epsilon,\delta)^2\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}(l_i+1)\sum_{g=1}^{\infty}\sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\\
&\le \max(\epsilon,\delta)^2\cdot 2n\sum_{g=1}^{\infty}\sum\nolimits_{\bar{\mathbf{x}}}\Pr(\bar{\mathbf{x}})\Pr(\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}|\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g)\\
&= \max(\epsilon,\delta)^2\cdot 2n\sum_{g=1}^{\infty}\Pr_g. \qquad (11)
\end{aligned}$$

Firstly in steps (a)-(f), we convert the entropy $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ averaging over PreESS $\bar{\mathbf{X}}$ and typicalized PosESS $\hat{\mathbf{Y}}$, to the number of splits on the alignment tree (*i.e.* ambiguous local alignments unresolved, recall Definition 7) averaging over PreESS $\bar{\mathbf{X}}$ and the original edit pattern $\bar{\mathbf{E}}$. In equality (b), the set $\{\bar{\mathbf{e}} : (\bar{\mathbf{x}},\bar{\mathbf{e}}) \to \hat{\mathbf{e}} \to \hat{\mathbf{y}}\}$ denotes the set of edit pattern $\bar{\mathbf{e}}$ such that by typicalizing $\bar{\mathbf{e}}$ to $\hat{\mathbf{e}}$ according to $\bar{\mathbf{x}}$ and processing $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$, we obtain $\hat{\mathbf{y}}$. Inequality (c) follows by bounding the entropy of the alignment tree $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ from above by the expectation of the number of splits $N_s(P_{\hat{A}})$ on the paths. Recall that a path of the alignment tree corresponds to a certain global alignment of $(\bar{\mathbf{x}},\hat{\mathbf{y}})$, hence corresponds to a set of typicalized edit pattern denoted by $\{\hat{\mathbf{e}} : P_{\hat{A}}\}$. The probability of $\hat{\mathbf{e}}$ is the sum of the probabilities of all $\bar{\mathbf{e}}$'s in the set $\{\bar{\mathbf{e}} : (\bar{\mathbf{x}},\bar{\mathbf{e}}) \to \hat{\mathbf{e}}\}$, that is, the set of $\bar{\mathbf{e}}$ resulting in $\hat{\mathbf{e}}$ after typicalization. Equality (e) and (f) follows because recall in Definition 7, by fixing $\bar{\mathbf{x}}$ and $\bar{\mathbf{e}}$, the path on the alignment tree is fixed. Moreover, for all the $\bar{\mathbf{e}}$'s which correspond to the same path, the number of splits $N_s(P_{\hat{A}}(\bar{\mathbf{x}},\bar{\mathbf{e}}))$'s are equal. Only if when $\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}$ occurs, the alignment tree may have a split associated with the ith $\bar{\mathbf{x}}$-run on the path of the alignment associate with $\bar{\mathbf{e}}$, in which case one bit is needed to distinguish the two ambiguous edit patterns. Hence, the total number of bits needed to distinguish the path (hence also alignment) associated with $\bar{\mathbf{e}}$ from other paths is bounded from above by $\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}}\mathbb{1}_{\mathbf{G}_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}}$, as shown in inequality (g). Inequality (h) follows because for fixed $\bar{\mathbf{x}}$, if the gap is reached before the end of $\bar{\mathbf{x}}$, then $\sum_{\bar{\mathbf{e}}}\Pr(\bar{\mathbf{e}})\Pr(\Gamma_i, G_i^{\bar{\mathbf{x}},\bar{\mathbf{e}}}=g) \le (l_i+1)\cdot\max(\epsilon,\delta)\cdot(1-\epsilon-\delta)^g\cdot\max(\epsilon,\delta) \le (l_i+1)\max(\epsilon,\delta)^2$. Otherwise, there is no further edits in $\hat{e}$ until the end of $\bar{\mathbf{x}}$. We argue earlier in the analysis of Case $\Gamma^1(\bar{\imath})$ that the ambiguous edit pattern needs more symbols after the gap to produce the same PosESS through some typical edits. If the gap reaches the end of $\bar{\mathbf{x}}$, there is no more symbol after the gap, hence no ambiguity at all.

We have shown above via case analysis that $\Pr_g \le \left(\frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3}\right)^{t-1}$ for $g = 2t-1$ or $g = 2t$ where $t = 1, 2, \ldots$. Hence from equation (11), we have $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \le \max(\epsilon,\delta)^2\cdot 2n\cdot\sum_{g=1}^{\infty}\Pr_g = \mathcal{O}(\max(\epsilon,\delta)^2)n$.

APPENDIX C
PROOF OF LEMMA 10

Given that the sequence $\tilde{O}^{n+\tilde{\epsilon}n}$ has $\tilde{\epsilon}n$ insertions and $\tilde{\delta}n$ deletions, the distribution of insertion $\imath$, deletion Δ and no-operation η in $\tilde{O}^{n+\tilde{\epsilon}n}$ is

$$p_\imath = \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}}, p_\Delta = \frac{\tilde{\delta}}{1+\tilde{\epsilon}}, p_\eta = \frac{1-\tilde{\delta}}{1+\tilde{\epsilon}}. \tag{12}$$

Hence, the entropy of $\tilde{O}^{n+\tilde{\epsilon}n}$ is

$$\begin{aligned}
&H(\tilde{O}^{n+\tilde{\epsilon}n}) \\
&= (1+\tilde{\epsilon})n \cdot \Big(-\frac{1-\tilde{\delta}}{1+\tilde{\epsilon}} \log \frac{1-\tilde{\delta}}{1+\tilde{\epsilon}} - \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}} \log \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}} \\
&\qquad -\frac{\tilde{\delta}}{1+\tilde{\epsilon}} \log \frac{\tilde{\delta}}{1+\tilde{\epsilon}} \Big) \\
&= n \cdot \Big[\mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + (1-\tilde{\epsilon}) \log(1-\tilde{\epsilon}) + (1+\tilde{\epsilon}) \log(1+\tilde{\epsilon}) \Big] \\
&\overset{(a)}{=} n \Big[\mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + (1-\tilde{\epsilon})(\log e) \big(-\tilde{\epsilon} - \frac{\tilde{\epsilon}^2}{2} - \frac{\tilde{\epsilon}^3}{3} + \mathcal{O}(\tilde{\epsilon}^4) \big) \\
&\quad + (1+\tilde{\epsilon})(\log e) \big(\tilde{\epsilon} - \frac{\tilde{\epsilon}^2}{2} + \frac{\tilde{\epsilon}^3}{3} + \mathcal{O}(\tilde{\epsilon}^4) \big) \Big] \\
&= n \cdot \Big[\mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4) \Big],
\end{aligned}$$

where step (a) is by Taylor series expansion.

Hence, $\lim_{n\to\infty} \frac{1}{n} H(\tilde{O}^{n+\tilde{\epsilon}n}) = \mathcal{H}(\tilde{\delta}) + \mathcal{H}(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)$.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editor for their helpful comments and suggestions.

REFERENCES

[1] J. C. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy, "Potential benefits of delta encoding and data compression for HTTP," in *Proc. ACM SIGCOMM*, 1997, vol. 27. no. 4, pp. 181–194.

[2] R. C. Burns and D. D. Long, "Efficient distributed backup with delta compression," in *Proc. 5th Workshop I/O Parallel Distrib. Syst.*, 1997, pp. 27–36.

[3] T. Suel and N. Memon, "Algorithms for delta compression and remote file synchronization," in *Handbook of Lossless Compression*. New York, NY, USA: Academic, Aug. 2002.

[4] L. Su and O. Milenkovic, "Synchronizing rankings via interactive communication," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2014, pp. 1056–1060.

[5] G. Cormode, M. Paterson, S. C. Şahinalp, and U. Vishkin, "Communication complexity of document exchange," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2000, pp. 197–206.

[6] A. Orlitsky and K. Viswanathan, "Practical protocols for interactive communication," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2001, p. 115.

[7] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[8] R. Venkataramanan, V. N. Swamy, and K. Ramchandran. (Oct. 2013). "Low-complexity interactive algorithms for synchronization from deletions, insertions, and substitutions." [Online]. Available: https://arxiv.org/abs/1310.2026

[9] N. Ma, K. Ramchandran, and D. Tse, "A compression algorithm using mis-aligned side-information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 16–20.

[10] R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2010, pp. 1412–1419.

[11] S. M. S. T. Yazdi and L. Dolecek, "Synchronization from deletions through interactive communication," in *Proc. IEEE 7th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Aug. 2012, pp. 66–70.

[12] N. Bitouzé and L. Dolecek, "Synchronization from insertions and deletions under a non-binary, non-uniform source," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 2930–2934.

[13] N. Ma, K. Ramchandran, and D. Tse, "Efficient file synchronization: A distributed source coding approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2011, pp. 583–587.

[14] S. El Rouayheb, S. Goparaju, H. M. Kiah, and O. Milenkovic. (Sep. 2014). "Synchronizing edits in distributed storage networks." [Online]. Available: https://arxiv.org/abs/1409.1551

[15] Y. Kanoria and A. Montanari, "On the deletion channel with small deletion probability," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2010, pp. 1002–1006.

[16] A. D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[17] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.

[18] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *J. Combinat. Theory A*, vol. 93, no. 2, pp. 310–332, 2001.

[19] V. I. Levenshtein, "Bounds for deletion/insertion correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2002, p. 370.

[20] A. Orlitsky, "Worst-case interactive communication. I. Two messages are almost optimal," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1111–1126, Sep. 1990.

[21] A. Orlitsky, "Worst-case interactive communication. II. Two messages are not optimal," *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 995–1005, Jul. 1991.

[22] A. Orlitsky, "Interactive communication: Balanced distributions, correlated files, and average-case complexity," in *Proc. 32nd Annu. Symp. Found. Comput. Sci.*, Oct. 1991, pp. 228–238.

[23] A. Orlitsky, "Average-case interactive communication," *IEEE Trans. Inf. Theory*, vol. 38, no. 5, pp. 1534–1547, Sep. 1992.

[24] A. Orlitsky, "Interactive communication of balanced distributions and of correlated files," *SIAM J. Discrete Math.*, vol. 6, no. 4, pp. 548–564, 1993.

[25] A. Tridgell, "Efficient algorithms for sorting and synchronization," Ph.D. dissertation, Austral. Nat. Univ., Canberra, Australia, Apr. 2000.

[26] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors," (in Russian), *Automatika Telemekhanika*, vol. 26, no. 2, p. 288292, 1965.

[27] Y. Kanoria and A. Montanari, "Optimal coding for the binary deletion channel with small deletion probability," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6192–6219, Oct. 2013.

[28] S. Ross, *A First Course in Probability*, 8th ed. Upper Saddle River, NJ, USA: Pearson, 2009.

[29] T. M. Cover, "The entropy of a randomly stopped sequence," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1641–1644, Nov. 1991.

[30] F. S. Makri and Z. M. Psillakis, "On success runs of a fixed length in Bernoulli sequences: Exact and asymptotic results," *Comput. Math. Appl.*, vol. 61, no. 4, pp. 761–772, 2011.

[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[32] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168–173, 1974.

[33] E. Ukkonen, "On approximate string matching," in *Foundations of Computation Theory*. Berlin, Germany: Springer, 1983, pp. 487–495.

[34] D. G. Brown. *How I wasted too Long Finding a Concentration Inequality for Sums of Geometric Variables*, accessed on May 8, 2015. [Online]. Available: https://cs.uwaterloo.ca/~browndg/negbin.pdf

[35] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of i.i.d. deletion and duplication channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2693–2714, Aug. 2007.

[36] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," MIT Lincoln Lab Group Rep. 2502, 1961.

[37] M. Rahmati and T. M. Duman, "Bounds on the capacity of random insertion and deletion-additive noise channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5534–5546, Sep. 2013.

Qiwen Wang received her B.Sc. degree in Mathematics and B.Eng. degree in Information Engineering in 2010, and Ph.D. degree in Information Engineering in 2015, all from the Chinese University of Hong Kong, Hong Kong. She is currently a postdoctoral researcher in the Department of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden.

Sidharth Jaggi B.Tech. (00), EE, IIT Bombay, MS/Ph.D. (05) EE, CalTech, Postdoctoral Associate (06) LIDS, MIT, currently Associate Professor, Dept. of Information Engineering, The Chinese University of Hong Kong.

Muriel Médard (S'90–M'95–SM'02–F'08) is the Cecil H. Green Professor of Electrical Engineering and Computer Science at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Her research interests are in the areas of network coding and reliable communications, particularly for optical and wireless networks. She has served as an Editor of many IEEE publications, and she is currently the Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She serves on the Board of Governors of the IEEE *Information Theory Society*, for which she was President in 2012. She has served as a TPC Co-Chair for ISIT, WiOpt, CONEXT, and Netcod and a Co-Chair for ISIT and Netcod. She was the recipient of the 2013 MIT Graduate Student Council EECS Mentor Award, the 2009 Communication Society and Information Theory Society Joint Paper Award, the 2009 William R. Bennett Prize in the Field of Communications Networking, the 2002 IEEE Leon K. Kirchmayer Prize Paper Award, and several conference paper awards. She was also a co-recipient of the MIT 2004 Harold E. Edgerton Faculty Achievement Award. In 2007, she was named a Gilbreth Lecturer by the National Academy of Engineering.

Viveck R. Cadambe (S'06–M'11) is an Assistant Professor in the Department of Electrical Engineering at Pennsylvania State University. Dr. Cadambe received his Ph.D from the University of California, Irvine in 2011. Between 2011 and 2014, he was a postdoctoral researcher jointly with the Research Laboratory of Electronics (RLE), MIT, Cambridge MA, USA and the ECE department at Boston University, Boston, MA, USA. His research interests include information theory, coding theory, and theory of distributed computing, with a focus on applications to wireless communication networks and distributed data storage and computing systems. Dr. Cadambe has served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since December 2014. Dr. Cadambe is a finalist for the 2016 Bell Labs Prize and a recipient of the 2016 NSF CAREER Award, the 2014 IEEE International Symposium on Network Computing and Applications (NCA) Best Paper Award, the 2009 IEEE Information Theory Society Paper Award and the UCI Electrical Engineering and Computer Science Department Best Paper Award for 2008-09. His dissertation received the 2011 CPCC Best Dissertation Award in the UCI Electrical Engineering and Computer Science Department. He was an intern at the Communications, Collaboration and Systems Group at Microsoft Research, Redmond WA during June-September of 2010.

Moshe Schwartz (M'03–SM'10) is an associate professor at the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel. His research interests include algebraic coding, combinatorial structures, and digital sequences. Prof. Schwartz received the B.A. (*summa cum laude*), M.Sc., and Ph.D. degrees from the Technion - Israel Institute of Technology, Haifa, Israel, in 1997, 1998, and 2004 respectively, all from the Computer Science Department. He was a Fulbright post-doctoral researcher in the Department of Electrical and Computer Engineering, University of California San Diego, and a postdoctoral researcher in the Department of Electrical Engineering, California Institute of Technology. While on sabbatical 2012-2014, he was a visiting scientist at the Massachusetts Institute of Technology (MIT). Prof. Schwartz received the 2009 IEEE Communications Society Best Paper Award in Signal Processing and Coding for Data Storage.